creative commons
COMMONS DEED

저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공 학 석 사 학 위 논 문

절차적인 방법을 이용한 속성 도출에 관한 연구

지도교수 여정모

이 논문을 공학석사 학위논문으로 제출함

2014년 11월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

창 위(Chang Wei)

Collection @ pknu

CHANG WEI의 공학석사 학위논문을 인준함

2014년 11월 17일

주 심 이학박사 박홍복 (인)

위 원 이학박사 윤성대 (인)

위 원 공학박사 여정모 (인)

Collection @ pknu

목 차

I. 서론 1
II. 관련연구 3
2.1 선행연구 3
2.1.1 문장기법을 이용하여 대상업무 도출 3
2.1.2 업무중심 엔터티 도출 7
2.2 기존 속성 도출 방법 11
2.2.1 현장경험에 의한 속성 도출 11
2.2.2 소그룹별로 후보군(POOL)에 의한 속성 도출 12
III. 속성 도출 절차 15
3.1 용어 정의 15
3.2 속성 도출절차 18
3.2.1 속성후보 도출(단계1) 18
3.2.2 속성후보 정제(단계2) 19
3.2.3 식별자 도출 및 확정(단계3) 20
3.2.4 속성 확정(단계4) 21
IV. 적용사례 23
4.1 속성후보 도출(단계1) 27
4.2 속성후보 정제(단계2) 32
4.3 식별자 도출 및 확정(단계3) 35
4.4 속성 확정(단계4) 39
V. 제안 절차의 검증 45
VI. 결론 48
VII. 참고문헌 50

Collection @ pknu

표 목 차

표 1. 도출된 대상업무 6
표 2. 엔터티 정의서 9
표 3. 도출된 엔터티 26
표 4. 속성후보 도출할 때 필요한 컬럼명.....29
표 5. 속성후보 도출 결과 31
표 6. 속성후보 정제 결과 34
표 7. 식별자 도출과 확정 결과 38
표 8. 속성 확정 결과 40
표 9. 사례의 속성정의서 44
표 10. 유사도 검사 결과 47

dCollection @ pknu

그 림 목 차

그림 1. 소그룹별 속성후보POOL 14
그림 2. ‘VOC 시스템’의 업무기술서.....25
그림 3. 속성후보들을 수집 경로 27

Collection @ pknu

절차적인 방법을 이용한 속성 도출에 관한 연구

창 위

부경대학교 일반대학원 컴퓨터공학과

요 약

현재의 기업 정보시스템의 골격을 정의한 설계도라고 할 수 있는 데이터 모델은 정밀하게 작성되어야 한다. 데이터 모델의 핵심요소로는 엔터티, 속성, 관계가 있으며, 이 중에서도 속성은 실질적인 정보를 담는 가장 기본적인 단위라 할 수 있으므로 모든 정보의 근원이라 할 수 있을 것이다. 그래서 속성들을 제대로 도출하지 못하면 데이터 모델 전체가 무의미하게 될 수 있다.

기존의 속성도출하는 방법은 설계자의 경험에 많이 의존하고 실질적인 절차가 존재하지 않아 실무경험이 없는 초보자가 도출하기에는 너무나 어려운 것이 현실이다. 이를 해결하는데 도움이 될 수 있도록 본 연구에서는 데이터 모델 설계의 한 과정으로서, 선행연구에서 제시한 업무중심 엔터티 도출 방법을 이용하여 엔터티가 완전히 도출되어 있다고 가정하고 미리 도출되어 있는 엔터티를 바탕으로 속성을 도출하는 절차를 제안한다. 그리고 데이터 모델링 경험이 많이 없는 학부생 및 대학원생을 대상으로 본 연구에서 제안한 절차를 적용하도록 하였다. 기존에 속성을 도출하는 방법이 실질적으로 존재하지 않

Collection @ pknu

기 때문에 학생들이 도출한 속성과 전문 IT 컨설턴트로 멘토가 도출한 모법 답안 간의 유사도검사를 하였다. 최종 유사도 검사를 통하여 전문 IT 컨설턴트인 멘토가 도출한 모법 답안에 상당히 근접하게 속성을 도출할 수 있다는 것을 확인하였다.

따라서 본 연구에서 제안한 절차를 활용한다면 현장에서도 적용 가능하며, 데이터모델링에 실무경험이 없는 초보자나 미숙련자가 적용하여도 속성을 도출할 수 있음을 보였다.

Collection @ pknu

A Study on Elicitation of the Attribute by Procedural Method

CHANG WEI

Dept. of Computer Engineering , The Graduate School,

Pukyong National University

Abstract

Data model which is considered as design drawing to define structure of the current company information system should be precisely composed. Key factors of data model are entity, attribute and relation. Among them, attribute is the most basic unit to contain substantial information, entitled as source of all information. Therefore, if properties are not properly deducted, the entire data model may be useless.

The previous method to deduct properties has been considerably based on experiences of a designer and there has no substantial procedure, and it turns out to be difficult for a beginner without any hand-on-background. In order to help solving this problem, the study suggests a procedure to deduct attribute based on already-deducted entity with presumption of complete deduction of entity by using work-focusing entity deduction methods suggested in the previous studies, as one of processes of data model design. And the procedure suggested by this study was applied to undergraduate and graduate students without abundant data modelling experiences. As there have been no substantial method to deduct properties, properties deducted by students and exemplary model by professional

Collection @ pknu

IT consultant as mentor were tested for similarities between them. Through the final similarity test, it was confirmed that students could deduct properties considerably close to exemplary model by a mentor who is a professional consultant.

Accordingly, if the procedure suggested by this study is applied, a beginner or unskillful programmer without hand-on-background will be able to deduct attribute.

Collection @ pknu

Ⅰ. 서론

현재의 기업 정보시스템의 핵심이 되는 데이터모델에서 속성은 상당히 중요한 요소이다. 데이터모델에서 속성은 엔터티의 구성요소로서 구체적인 정보를 설명하는 항목이고 더 이상 분리될 수 없는 최소 데이터 보관 단위이기 때문에 모델링 작업에서 중요한 부분을 차지하고 있다. 모든 생물의 기능적이며 구조적 기본 단위인 세포와 같이 모델링 작업에서 속성 또한 세포와 유사한 기본적 단위이다. 이렇듯 모델에서 많은 비중을 차지하고 중요한 부분을 담당하는 속성을 정확하지 못한 기준을 가지고 함부로 판단한다면 속성이 포함된 엔터티 뿐만 아니라 데이터 모델 전체가 무의미하게 될 수도 있다. 그러므로 데이터 모델에서 속성을 정확하게 판단하여 도출하는 것은 너무나 중요하다.

기존속성도출방법은 데이터를 분석하여 업무 분석과 데이터 모델링을 동시에 진행하는 경험에 의한 속성 도출방법을 많이 사용하였다[1,6]. 이런 기존방법에서는 구체적으로 정의된 절차 없이 설계자의 경험과 업무지식에 많이 의존하기 때문에 데이터 모델링 초보자가 진행하기 어렵다. 특히 경험이 부족한 초보자가 곧바로 작업하면 판단의 오류를 저지를 수 있을 뿐만 아니라 불필요한 시행착오를 많이 경험할 것이다. 그리고 작업도중 자기가 한 것에 대해 옳고 그름을 판단하지 못한 채 다음으로 진행해 나가는 것은 올바르지 못하다. 또한 잘못된 과정으로 인해 열심히 노력하여 도출한 결과가 모두 쓸모없게 돼버린다. 그러므로 초보자가 데이터 모델링과정에서 속성을 올바르게 도출하는 체계적인 절차가 필요하다.

Collection @ pknu

본 연구에서 제안한 절차는 데이터 모델 설계의 한 과정으로서, 선행연구에서 제시한 업무중심으로서의 엔터티 도출 방법을 이용하여 엔터티가 완전히 도출되어 있다고 가정하고 미리 도출되어 있는 엔터티를 바탕으로 속성을 도출하는 것이다. 그리고 선행연구에 대한 기존의 엔터티 도출 방법은 설계자의 많은 설계 경험과 업무지식을 필요로 한다고 언급했듯이 속성도 마찬가지다[6]. 실무경험이 없는 초보자가 도출하기에는 너무나 어려운 것이 현실이다

따라서 본 연구에서 많은 초보자나 미숙련자들뿐만 아니라 실무적 경험이 풍부한 사람들도 데이터 모델링을 효과적으로 진행할 수 있도록 체계적이고 절차적인 속성 도출 절차를 제안하고자 한다.

Collection @ pknu

II. 관련연구

본 장에서는 선행 연구에서 대상업무 도출방법 및 엔터티 도출 방법을 고찰하고 현장에서 기존에 속성도출 방법 및 소그룹별로 후보군(Pool)에 의한 속성 도출방법에 대하여 알아보고 문제점을 분석한다.

2.1 선행연구

본 절에서 선행 연구에서 제시된 대상업무 도출방법 및 엔터티 도출 방법에 대하여 알아본다.

2.1.1 문장기법을 이용하여 대상업무 도출

먼저 업무중심으로 엔터티 도출을 위해서는 선행연구에서 제안한 문장기법을[3] 이용하여 대상업무가 도출되어야 한다. 문장기법을 이용하여 대상업무 도출 방법은 다음과 같이 5단계로 진행된다.

단계 1 : 업무후보 도출 및 1차 정제 단계

단계1에서는 수집된 요구사항을 문장 형태로 기술한 업무기술서의 각 문장을 읽어가면서 문장 안에 있는 동사나 명사형 동사를 기준으로 업무후보들을 도출한다. 도출과정 중에 나타나는 시스템행위들은 업무행위에 관련된 후보를 제외하고 나머지는 제거한다. 그리고 복합적인 행위를 포함하는 후보는 모두 단일 행위들로 분리하여 도출한다. '업무후보명'

Collection @ pknu

은 ‘행위대상+행위’의 형태로 기술하고, ‘업무후보 설명’은 행위를 명확하게 파악할 수 있도록 업무기술서의 내용을 바탕으로 육하원칙을 적용하여 기술한다. 그리고 데이터를 의미하는 부분도 같은 방법으로 데이터 업무후보를 도출하고, 도출된 모든 후보를 검증한다.

단계 2 : 업무 데이터 표준화 단계

단계2에서는 도출된 데이터에 대하여 업무 데이터 표준화하는 작업을 진행한다. 애플리케이션의 설계와 구축에 관련된 모든 이해당사자들 사이에 의사소통을 원활히 하도록 업무단어와 업무용어에 대하여 데이터 사전을 구성하고 적용한다.

단계 3 : 업무후보 2차 정제 단계

단계3에서는 동일한 의미의 행위 업무후보들을 모아서 그룹핑한다. 각 그룹에서 대표를 선정하고 대표가 아닌 후보는 제거한다. 그 다음에 데이터 업무후보에 대해 또 같은 방법으로 2차 정제를 진행한다.

단계 4 : 대상업무후보 전환 단계

행위 업무후보들을 업무가 발생하는 순서대로 재배치한다. 이 중에 업무의 발생순서와 관련 없거나 대부분의 업무에 공통적인 업무후보들은 마지막부분에 위치하도록 한다. 업무후보가 빠짐없이 도출되었는지를 검증하기 위해서 데이터 업무후보들을 각각 자신을 사용하는 행위 업무후보 아래로 이동시키고 검증한다. 그러고 나서 다시 한 번 그룹핑으로 중

Collection @ pknu

복되거나 필요 없는 업무후보를 제거한다. 남아 있는 행위 '업무후보명'과 '업무후보 설명'을 '대상업무후보명'과 '대상업무후보 설명'으로 전환한다.

단계 5 : 대상업무 확정 단계

단계4의 결과에 대해서 유사한 대상업무후보들을 그룹핑하여 대표 대상업무후보를 선정한다. 대표가 되지 않은 것들을 대표 아래에 들여쓰기 한 후에 대표 대상업무후보를 그룹을 대표할 수 이도록 보완한다. 또한 그룹들도 좀 더 넓은 의미로 그룹핑될 수 있으면 그룹핑하여 계층구조로 구성할 수 있다. 개념과 의미가 명확해진 '대상업무후보명'과 '대상업무후보 설명'을 각각 '대상업무명'과 '대상업무 설명'으로 확정한다. 확정한 결과물을 근거로 사용자와 면담하여 검증하여 보완하도록 한다.

Collection @ pknu

표1은 도출된 대상업무이다.

대상업무명	대상업무 설명
SR(사용자요구)	사용자가 통신도구를 통해 CI에 대하여 도움을 요청하다.
상담	사용자는 정보 시스템의 사용에 대하여 사원과 상담하다.
변경요청	사용자가 CI변경을 요청하다.
장애신고	사용자들이 정보 시스템에 관련된 장애를 신고하다.
기타 요구	
SR 접수	SD담당자가 사용자의 SR을 접수하다
해결방안 검색	SD담당자는 SR이 Known Error에 해당하는지 VOC 시스템에서 신속히 검색하다
해결방안 제시	SD담당자는 SR이 KnownError에 해당하면 처리방안을 제시하다
평가자 지명	두 번 이상 영향평가를 해야 하는 경우, 영향평가자를 임명하다.
SR평가	변경관리담당자가 1차 처리되지 않은 SR에 대해 프로그램 등을 영향평가하다.
처리계획 수립	프로그램담당자가 영향평가를 바탕으로 변경대상 및 일정 등의 처리 계획을 수립한다.
SR처리	사원이 접수된 SR을 처리하다
CI변경	사원이 SR을 처리하는 도중에 CI를 변경하다.
SR처리 변경	사용자가 변경요청한 내용에 대하여 사원이 해당 CI를 변경하다.
업그레이드처리 변경	업그레이드에 의해 CI가 변경되다.
정기 PM처리 변경	정기 PM에 의해 CI가 변경되다.
기타처리 변경	
CI변경이력 관리	사원이 SR처리에 의해 변경되는 CI변경이력을 관리하다.
SR처리완료	프로그램담당자가 처리계획대로 처리하고 처리완료일을 처리계획에 기록하다.
SR완료 안내	SD담당자가 완료SR을 통신도구로ㅆ 사용자에게 안내하다.
처리결과 기록	SD담당자가 처리결과를 사용자에게 확인하여 기록하다.
FAQ 구성	빠른 검색이 가능하도록 FAQ를 구성하고 게재한다.
FAQ 검색	FAQ를 정해진 분류에 따라 빠르게 검색하다
사원 채용	사원은 정규직, 계약직, 임시직으로 구성된다.
담당자 임명	영향평가자, 변경관리담당자, SD담당자, 프로그램관리담당자, 사용자
SD담당자 지정	SD담당자는 계약직 사원이 담당한다.
부서 소속	지역에는 국내, 일본, 미주 등이 있다.

표 1. 도출된 대상업무

Collection @ pknu

2.1.2 업무중심 엔터티 도출

전 단계 이어서 대상업무가 제대로 도출하고 나서 업무중심으로 엔터티 도출해야 한다. 업무중심 엔터티 도출 방법[10]은 다음과 같이 3단계 진행된다.

단계 1 : 엔터티후보 도출

엔터티후보 도출을 위하여 미리 도출된 대상업무를 정리한다. 엔터티 도출 이전 단계인 업무분석 단계에서 확정된 대상업무들을 데이터 업무후보들과 함께 복사하고 '엔터티후보' 란과 '육하원칙 적용 엔터티후보/데이터' 란을 추가한다. 또한 데이터 업무후보의 검토를 위해 해당 대상업무 아래로 이동시킨다.

이후에 행위나 행위결과를 엔터티후보로 도출한다. '대상업무명' 란과 '대상업무 설명' 란에서 행위를 찾아 엔터티후보로 도출하고 '엔터티후보' 란에 기재한다. 행위결과가 불명확한 경우에는 행위만을 기재하고 행위와 행위결과를 함께 도출한 경우에는 '행위/행위결과' 형태로 기재한다.

행위/행위결과에 대하여 육하원칙을 적용하여 도출되는 엔터티후보는 '육하원칙 적용 엔터티후보/데이터' 란에 기재한다. 그리고 엔터티후보를 도출하면서 엔터티후보 간에 관계가 발견되면 표시하여 엔터티후보 간의 관계파악에 도움이 되도록 한다. 다음은 업무분석 단계에서 도출된 데이터업무후보에서 엔터티후보를 도출한다. 먼저 정리된 데이터에서 엔터티로 판단한 엔터티후보를 도출하여 '엔터티후보' 란에 기재한다.

부가적 처리 작업이 있는 경우에는 추가적으로 엔터티후보가 도출될

Collection @ pknu

수 있으므로 ‘<부가처리명>’ 형태로 ‘데이터’ 란에 표기하고 부가적인 처리에 사용된 엔터티후보를 ‘엔터티후보’ 란에 기재한다.

단계 2 : 엔터티후보 정제

단계1에서 엔터티가 될 가능성이 있다면 모두 후보로 도출하기 때문에 불필요한 후보를 정제하는 작업이 필요하다. 단계2에서는 도출된 엔터티 후보에 대해 자격 검증을 수행하고 역할별로 분류하여 동일하거나 유사한 후보들을 통합한다.

단계 3 : 엔터티 확정

엔터티는 엔터티의 개념을 어떻게 정립하느냐에 따라서 여러 엔터티로 분리할 수도 있고, 여러 엔터티를 하나의 엔터티로 통합할 수도 있으므로, 개념정립을 명확하게 해야 한다. 단계3에서는 가장 먼저 엔터티후보에 대하여 개념을 명확하게 정립하여 엔터티후보의 순수성과 동질성 및 독립성을 파악하고, 엔터티후보들을 통합 또는 분리한다. 그러고 나서 이력 엔터티와 서브타입을 확정함으로써 엔터티로 확정해 나간다. 엔터티후보를 엔터티로 확정하기 위해 이전 과정의 결과에서 제거되지 않은 엔터티후보와 서브타입을 새로운 시트로 복사하여 엔터티를 확정할 준비를 한다.

Collection @ pknu

확정한 엔터티에 대하여 엔터티 정의서를 작성한다.

엔터티	분류	정의	데이터처리형태	특이사항
SR	주요	사용자가 통신도구를 통해 도움을 요청한 내용		
상담	서브	사용자가 정보시스템의 사용에 대하여 사원과 상담한 내용	SR구분=1	
변경요청	서브	사용자가 CI변경을 요청한 내용	SR구분=2	
장애신고	서브	사용자들이 정보시스템에 관련된 장애를 신고한 내용	SR구분=3	
기타	서브	사용자가 정보시스템에 대해 요청한 기타 내용	SR구분=4	
CI	핵심	정보시스템 구성요소		
서버	서브	정보시스템 구성요소인 서버	CI구분=1	
N/W	서브	정보시스템 구성요소인 N/W	CI구분=2	
프로그램	서브	정보시스템 구성요소인 프로그램	CI구분=3	
DB Object	서브	정보시스템 구성요소인 DB Object	CI구분=4	
사원	핵심	회사에 근무하는 사람		
정규직	서브	사원의 계약구분 중 하나	계약구분=1	
계약직	서브	사원의 계약구분 중 하나	계약구분=2	
임시직	서브	사원의 계약구분 중 하나	계약구분=3	
직책	핵심	사원의 담당 역할		
SD담당	서브	사원의 담당 역할 중 하나	직책명=1	
변경관리담당	서브	사원의 담당 역할 중 하나	직책명=2	
영향평가	서브	사원의 담당 역할 중 하나	직책명=3	
프로그램관리담당	서브	사원의 담당 역할 중 하나	직책명=4	
프로그램담당	서브	사원의 담당 역할 중 하나	직책명=5	
직책할당	활동	사원에게 직책을 할당하기 위한 역할		
담당	활동	계약직 사원에게 SD를 담당하게 함		
SR처리	주요	사원이 접수된 SR을 처리하는 내용		
SR접수	서브	SD담당자가 사용자의 SR을 접수한 내용	SR처리구분=1	
제시	서브	SD담당자는 SR에 대한 해결방안을 제시한 내용	SR처리구분=2	
SR평가	서브	변경관리담당자가 처리되지 않은 SR에 대해 영향평가한 내용	SR처리구분=3	
처리계획	서브	프로그램관리담당자가 영향평가를 바탕으로 처리계획을 수립한 내용	SR처리구분=4	
처리완료	서브	계획대로 처리하고 처리완료한 내용	SR처리구분=5	
KnownError	활동	정보시스템에 대한 알려진 오류		
해결방안	활동	알려진 오류에 대하여 사원이 해결한 해결방안		
CI변경이력	활동	CI를 변경한 이력정보		
SR발생	서브	사용자요청에 의한 CI변경을 의미	변경근거=1	
정기PM	서브	정기적인 사전예방에 의한 CI변경을 의미	변경근거=2	
업그레이드	서브	업그레이드에 의한 CI변경을 의미	변경근거=3	
기타	서브	기타 사유로 인한 CI변경을 의미	변경근거=4	
부족부분	활동	정보시스템에서 부족하다고 판단된 부분		
개선부분	활동	정보시스템에서 개선의 필요가 있다고 판단된 부분		
SR경향	활동	SR의 경향		
월별SR집계	활동	월별로 집계한 SR통계		
부서별SR집계	활동	부서별로 집계한 SR통계		
FAQ	활동	빈번하게 묻고 답을 한 내용		
키워드할당	활동	FAQ에 키워드를 할당하는 내용		
키워드	핵심	FAQ에 나타나는 중요 단어나 용어		
부서	핵심	회사의 조직		

표 2. 엔터티 정의서

Collection @ pknu

각 단계를 진행하면서 사용자 면담을 통하여 중간 과정을 확인할 수 있으며, 최종적으로 도출된 엔터티에 대하여 사용자의 검증이 필요하다.

Collection @ pknu

2.2 기존 속성 도출 방법

본 절에서는 현실에서 기존의 있는 2가기 데이터 모델링 속성 도출 방법에 대해여 고찰하고 문제점을 분석한다.

2.2.1 현장경험에 의한 속성 도출

현장경험에 의한 속성도출 방법은 수집된 여러 자료들에서 중요 단어들을 중심으로 속성 도출하기 때문에 업무 경험에 따라 도출 경과가 서로 다를 수 있다. 그리고 현장에서는 데이터를 분석하여 업무 분석과 데이터 모델링을 동시에 진행하는 경험에 의한 속성 도출 방법을 많이 사용한다. 현장경험에 의한 속성도출 방법에서는 구체적으로 정의된 절차 없이 수집된 요구사항과 자료들을 보면서 속성후보 도출을 진행하는데, 일반적으로 각 속성후보가 데이터 집합여부, 속성 후보값은 다중 값 여부, 속성 후보 값은 가공된지 등을 검토하여 판단하고 즉시 속성후보로 도출하기 때문에 설계자의 경험과 업무지식에 많이 의존하게 되어야 한다[8].

현장경험 속성도출 방법은 기존 시스템의 설계문서 및 사용자 지침서등에 속성후보들을 도출할 수 있는 유용한 자료, 현업에서 사용하고 있는 장표나 보고자료. 데이터모델링 과정을 하면서 사용자와 면담, Data Flow Diagram의 데이터 저장소와 데이터사전의 정보, 동일한 업무에 대한 전문 서적 또는 자료, 그리고 사내외에 존재하는 관련 시스템이나 유사 시스템의 자료를 통해서 업무를 분석하여 요구사항을 도출하면서 동시에 속성후보를 도출한다. 속성후보가 모두 도출되면 도출된 속성후보들에 대하여 의미를 파악하여 속성의 자격이 있는지 검증하고[4], 속성후보

Collection @ pknu

정제과정을 진행한다.

이런 현장경험에 의한 속성도출 방법의 가장 큰 장점은 빠른 시간내 모법 답안에 근접한 데이터 모델을 설계할 수 있는 것이다. 하지만 이러 방법을 이용하여 효과적 결과를 얻기 위해서 설계자의 많은 경험과 업무 지식에 많이 의존한다. 그리고 데이터 모델링에 경험이 없는 초보자가 이런 방법이 접근하기에는 많은 어려움이 있다. 그러므로 데이터 모델링에 경험이 없는 초보자가 더 많은 시간과 노력을 투자해야 현장경험에 의한 속성도출 방법이 사용하기에는 가능성이 있다.

2.2.2 소그룹별로 후보군(POOL)에 의한 속성 도출

모든 엔터티가 정의되어 있는 상태가 아니라 단지 핵심 엔터티들을 대상으로 모델링을 실시해왔을 뿐이므로 아직은 모든 엔터티가 드러나 있지는 않다. 그렇기 때문에 각 속성 후보들을 적절한 데이터 그룹으로 생성하여 두는 것이 필요하다. 그래서 모든 엔터티들이 드러나 있지 않은 상황에서 속성을 정의하고자 할 때 적용 할 수 있는 방법 중에는 두 가지 방법이 존재한다. 첫 번째는 지금까지 도출된 엔터티에 직접적인 속성만을 대상으로 하는 방법이고, 두 번째는 비록 엔터티는 아직 모두 도출되지 않았지만 속성만큼은 눈에 보이는 모든 것을 도출해 주는 방법이다.

첫 번째 방법에 대한 간단한 예를 들어 보도록 하겠다. 회원이 취득한 자격증 엔터티를 도출하였다면 여기에 속할 것으로 보이는 '자격증 명, 자격증번호, 자격증 유효기간, 자격증 발급기관' 정도만 속성으로 도출될 수 있다. 그러나, 이 방법은 해당 엔터티 입자에서 볼 때는 자신의

Collection @ pknu

속성을 찾았다는 책임을 완수했기 때문에 문제가 되지 않겠지만, 아직 드러나지 않은 설문과 관련된 다른 엔터티들에 들어 갈 속성들은 들어갈 곳이 없어 누락이 된다는 문제가 있다.

두 번째 방법의 경우 엔터티는 모두 도출되지 않았지만 속성만큼은 눈에 보이는 모든 것을 도출해 주는 것이다. 예를 들어 보도록 하겠다면 '자격증과 관련된 대부분의 엔터티가 도출되어 있는 경우, '자격증번호, 자격증명, 자격증등급, 자격증취득일자, 회원명, 회원번호, 자격증유효기관' 등 속성으로 도출될 것이다. 이 방법은 일차로 속성들이 어느 엔터티에 소속되었든 상관없이 앞으로의 속성검증 단계에서나 정규화 단계를 거치면서 여기가 자신이 진정으로 위치해야 할 곳이 아니라고 판단되면 새로운 엔터티를 만들어서 독립하거나, 이미 존재하는 다른 엔터티로 옮겨 가게 되므로 일단 가장 확실해 보이는 엔터티로 소속시키는 것이 필요하다.

Collection @ pknu

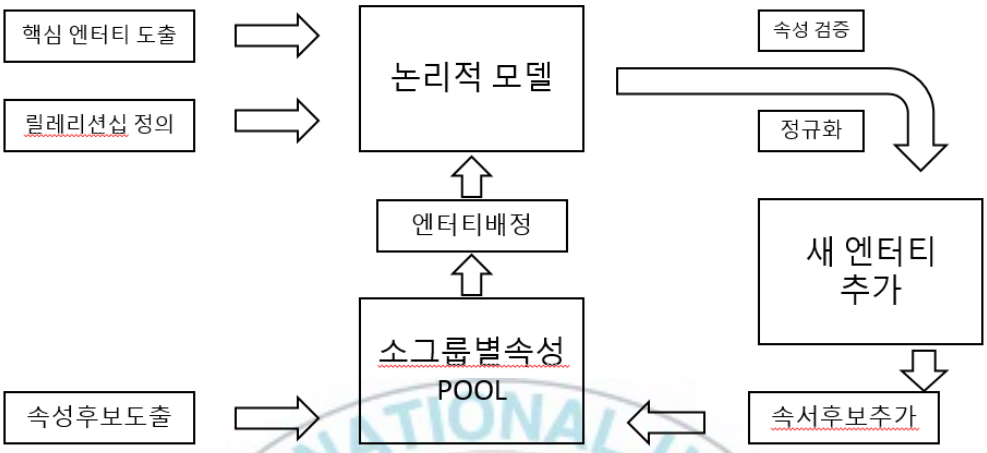

그림 1. 소그룹별 속성후보POOL

그림1에 있는 내용을 자세히 보면 속성을 정의하는 단계에 와서 확보 가능한 모든 속성 후보들을 도출하였고, 엔터티 소그룹별로 분류하여 속성 후보군(POOL)을 만들어 두었다.

이런 소그룹별로 후보군에 의한 속성 도출 방법의 가장 불편한 것은 2 가지 있다. 먼저 해당 엔터티 입장에서 볼 때는 자신의 속성을 찾았다는 책임을 완수했기 때문에 문제가 되지 않겠지만, 아직 드러나지 않은 다른 엔터티들에 들어 갈 속성들은 들어갈 곳이 없어 누락이 된다는 문제가 있다. 뿐만 아니라 새로운 생긴 엔터티의 개념을 명확화한 다음 거기에 있어야 할 것으로 예상되는 속성 후보들을 다시 찾아내어 후보군에 포함시키는 작업을 반복해야 한다는 것이다.

Collection @ pknu

III. 속성 도출 철자

데이터 모델을 정확하고 완전하게 설계하고자 한다면 기업에서 수행하는 엔터티들에 모든 업무가 반영된 정확한 속성들이 도출되어야 한다. 따라서 본 연구에서는 데이터 모델 설계의 핵심부분인 속성 도출 과정에서 실무적 경험이 풍부한 사람들뿐만 아니라 초보자나 미숙련자들도 효과적으로 진행할 수 있도록 체계적이고 절차적인 속성 도출 절차를 제안한다.

3.1 용어 정의

본 절에서는 속성 도출과 데이터 모델링에 관련된 용어에 대하여 설명한다.

속성이라는 의미에는 가공되지 않은 것이라는 의미도 포함되어 있다. 이 말은 곧 원천적인 것을 의미한다. 또한 고유한 성질이란 의미도 가지고 있는데, 이 말은 남의 도움을 받지 않더라도 자기만이 가지고 있는 독자적인 성질이 반드시 있어야 함을 뜻한다. 혼자서도 독자적인 성질을 가지고 있느냐에 대한 문제는 절대적이 아니라 상대적이라는 것 때문에 판단이 쉽지 않다. 다시 말하면 어떤 시스템의, 어떤 엔터티에서, 어떤 목적으로 사용하려고 하느냐에 따라서 달라지기 때문에 사람의 종합적인 판단이 필요하다는 것이다.

데이터모델에서 속성은 엔터티에서 관리되는 구체적인 정보 항목으로 더 이상 분리될 수 없는 최소의 데이터 보관 단위이다. 엔터티에 속하는 모든 인스턴스들이 공통으로 가지는 이러한 특성을 속성(Attribute)이라고 한다. 각 엔터티들은 일련의 속성들에 의해 상세화될 수 있다[8]. 엔터티는 데이터 집합이고, 데이터 집합의 원소는 속성이다. 따라서 속성

Collection @ pknu

(Attribute)은 엔터티를 의미하는 데이터로서, 하나의 엔터티에 속한 속성들은 엔터티의 특성이나 성격을 나타낸다. 즉 속성들로서 엔터티를 표현한다.

속성은 탄생에 따라 또는 성격에 따라 또는 속성 값의 존재 여부에 따라 분류할 수 있다.

우선 속성을 탄생에 따라 분류하면. 전혀 인위성이 없이 탄생한 원시 속성과 조금이라도 인위성을 기지고 탄생된 인조 속성이 있다.

원시 속성(Primitive Attribute)은 인위성이 전혀 없는, 6하원칙에 의해 탄생된 속성으로서, 6하원칙 속성(5W1H)이라고도 한다. 속성에는 6하원칙의 적용 성격에 따라 자신을 의미하거나 행위의 주체 성격을 의미하는 주체형 속성(Who-Role Attribute), 시간을 의미하는 시간형 속성(When-Role Attribute), 장소를 의미하는 장소형 속성(Where-Role Attribute), 목적이나 대상을 의미하는 대상형 속성(What-Role Attribute), 방법이나 참고적인 정보 항목을 의미하는 방법형 속성(How-Role Attribute) 및 원인이나 근거를 의미하는 근거형 속성(Why-Role Attribute) 등으로 분류될 수 있다

인조 속성(Artificial Attribute)은 업무의 필요에 따라 또는 시스템의 관리의 효율을 위하여 인위적으로 생성한 속성이다.

그리고 속성을 성격에 따라 분류할 수도 있다. 순수하게 엔터티 자신에게만 소속되어, 자신을 표현하는 속성을 고유 속성이라 하고, 다른 속성으로 가공한 속성을 추출 속성이라 하며, 자신이 아닌 다른 엔터티를 표현하는 속성을 관계 속성이라 한다.

Collection @ pknu

고유 속성(Inherent Attribute)은 엔터티 자신과 관련하여 자신을 설명하고 표현하는 속성들로써, 주로 자신의 명칭, 내용, 특성 및 성질 등을 나타내는 속성들이 해당된다.

추출 속성(Derived Attribute)은 엔터티 자신이나 다른 엔터티의 속성으로부터 가공된 속성으로서 집계, 유도, 파생, 계산, 중복 등으로 발생된 속성이다.

관계 속성(Relational Attribute)은 자신이나 다른 엔터티와의 관계에 의하여 생성된 속성으로서, 부모가 자식을 탄생시킬 때 상속시켜 준 속성이다.

또 속성 값의 존재의 따라서 필수/ 선택 두 가지 속성이 분류할 수도 있는데, 필수 속성(Mandatory Attribute)은 엔터티의 속성 값이 항상 존재하는 속성이고 선택 속성(Optional Attribute)은 경우에 따라 속성 값이 NULL일 수도 있는 속성이다.

식별자(UID; Unique Identifier)는 엔터티에 소속된 인스턴스들을 식별할 수 있는 최소 속성들로 구성된 집합이다. 상속(Inheritance)은 상대 인스턴스의 식별자 속성값을 가져와서 자신의 속성 값으로 저장한다. 상속해주는 엔터티를 부모 엔터티(Parent Entity), 상속받는 엔터티를 자식 엔터티(Child Entity)라고 한다.

Collection @ pknu

3.2 속성 도출절차

본 연구에서는 선행연구에서 제시한 업무중심으로 엔터티 도출 방법을 이용하여 엔터티가 완전히 도출되어 있다고 가정하고 미리 도출되어 있는 엔터티를 바탕으로 속성을 도출한다.

3.2.1 속성후보 도출(단계1)

속성을 도출하기 위하여 먼저 속성후보로 가능한 것들을 모두 수집이 필요하다. 속성을 결정할 때에도 다양한 후부를 준비하고 이들 중에서 속성의 검증 규칙에 맞는 것을 속성으로 최종 결정하게 된다.

전 단계의 기존 속성후보들을 속성후보로 포함시킨다. 전 단계에서 진행했던 엔터티후보 정제에 있는 속성 데이터로 분류되었던 속성들을 해당 엔터티로 소속시키는데, 주로 업무명세서에 명시된 내용이 대부분일 것이다. 서브타입 엔터티에 해당되는 속성들을 해당 서브타입에 소속시키고, 서브타입들에 공통적인 속성들은 해당 슈퍼타입에 소속시킨다. 또한 속성들을 해당 엔터티에 포함시킬 때, 필요하다면 속성의 명칭을 적절하게 변경하도록 한다.

다음에는 엔터티의 고유한 고유 속성후보들을 도출한다. 고유속성은 전혀 가공되지 않은 속성이기 때문에 어떤 방법으로든 자신을 재현할 수 있는 방법이 없다. 예를 들어 '계약' 엔터티의 '계약일자',' 계약내용', '계약수단' 등은 '계약' 자신을 표현하고 있으므로 고유 속성에 해당된다.

그리고 키 엔터티, 메인 엔터티. 활동 엔터티. 코드 엔터티 순서로 6하

Collection @ pknu

원칙을 적용시켜 해당 속성후보들을 도출한다. 예를 들어 주문 엔터티의 '주문날짜' 속성후보는 시간형 속성후보에 해당되고 결제 엔터티의 '결제수단' 속성후보는 방법형 속성후보에 해당된다. 그런데 너무 무리하게 도출하여 불가능한 속성이나 비관리대상일 확률이 높은 속성후보들은 도출하지 않도록 한다.

3.2.2 속성후보 정제(단계2)

이제까지 도출한 속성후보들은 정제된 속성후보들이 아니기 때문에 검증되기 전 까지는 속성의 자격이 없다. 따라서 정제 과정을 거쳐서 속성으로 확정하여야 한다. 속성을 정제하는 작업은 총 4 단계로 나누어 실시하며, 각 단계마다 매우 객관적인 판단의 기준들이 제시될 것이다.

첫 번째 단계는 속성은 최소 단위까지 분할한다. 일반적으로 가능하면 한 속성은 최고 단위까지 분할하는데 업무상 관리에 따라서 최소 분할 단위를 결합하여 사용하도록 한다. 어떤 업무가 속성의 전체적인 정보를 필요로 한다면 하나의 속성으로 표현할 수도 있고, 동일 업무를 상세하게 관리할 필요가 있는 경우에는 두 개 이상의 속성으로 분할하여 사용할 수도 있다. 예를 들어 '주소'와 같은 경우에는 처음에는 독립적으로 표현해 두었다가, 설계 단계에서 우편번호를 기준으로 하는 '기본주소' 속성과 '상세주소' 속성으로 분할하여 사용하는 것이 효율적이면 분할하여 사용한다.

두 번째 단계는 하나의 값만을 가지는지 검증한다. 속성에서 관리되어야 할 값이 반드시 단 하나만 존재해야 한다는 것이다. 다시 말하면 그

Collection @ pknu

엔터티에 들어가는 개체마다 반드시 하나의 값만 보유하고 있어야 한다는 것이다.

세 번째는 추출 속성인지 검증한다. 속성 정제의 제 3단계는 그 속성이 원천적인 값인지, 다른 속성에 의해 가공되어서 만들어진 값인지를 검증하자는 것이다. 원천적인 값은 다른 것에 의해 만들어진 것이 아닌 태초부터 창조된 것을 말한다. 추출 값은 이들을 가지고 언제라고 쉽게 재현할 수 있는 속성을 말한다.

마지막의 단계는 상세하게 관리하는지 검토한다. 현재까지는 이대로도 충분했지만 수많은 변화가 예상되는 미래를 대비하기 위해 좀 더 근원적인 정보를 관리할 필요성에 대해 검토해 보는 것이다. 이 단계가 데이터 모델링이 현재뿐만 아니라 미래의 시스템 구조를 찾아 나가는 과정이라는 것을 의미한다.

3.2.3 식별자 도출 및 확정(단계3)

속성후보들 중에서 실질 식별자로 사용 가능한 속성후보가 있으면, 속성후보 명칭 앞에 '#' 기호를 붙이고 속성으로 확정함과 동시에 실질 식별자로 확정한다. 실질 식별자로 적절한 속성후보들이 없으면, 실질 식별자로 사용할 수 있는 적절한 인조 식별자를 생성한다. 또한 식별자 속성들이 많아 관리에 불편한 경우 등에서 인조 식별자를 생성하여 사용할 수 있다. 그리고 실질 식별자 대신에 보조 식별자를 상속받기를 원하는 자식 엔터티가 있는지 파악한다. 그러한 경우가 있으면 상속받기를 원하는 식별자를 보조 식별자로 선정한다. 또한 자신의 서브타입 내에서 실체들을 구분할 필요가 있는지 파악하는 것이다. 구분할 필요가 있는

Collection @ pknu

경우에는 서브타입 엔터티의 식별자를 보조 식별자로 선정하도록 한다.

식별자 확정이 모두 끝났으면, 일단 일반 속성후들을 속성들로 확정하고 속성으로 적절한지 판단하고 명칭이 적절한지 판단하여, 속성으로 확정하고 필요하다면 명칭도 적절하게 변경하도록 한다. 그리고 필수 속성인지, 선택 속성인지 구분하도록 한다.

3.2.4 속성 확정(단계4)

이제까지 속성후보들에 대하여 검증이 끝나고 식별자도 대부분 확정된 상태이다. 그런데 아직까지 관계들이 확정되지 않아서 외래 식별자들이 확정된 상태가 아니기 때문에 엔터티의 모든 속성들이 확정된 상태는 아니다. 그래서 확정되지 않는 속성들을 확정하고 외래 식별자들, 본질 식별자들, 실질 식별자들 및 대체 식별자들을 확정하도록 한다.

이런 확정되지 않는 속성은 모두가 부모 엔터티로부터 상속받은 외부 식별자 속성들이 될 것이다. 이러한 속성들에 대하여 부모 엔터티를 먼저 결정해야 한다. 기존에 엔터티 중에서 부모 역할을 할 수 있는 엔터티가 있으면 매핑시켜 주고, 적당한 엔터티가 없는 경우에는 추가로 생성시켜 주어야 한다.

부모 엔터티가 결정된 속성들은 결정된 부모 엔터티로부터 식별자를 상속받아 모두 외부 식별자 속성이 된다. 따라서 외부 식별자 속성으로 사용할 수 있는 적절한 명칭을 부여하여 확정하도록 한다. 그리고 외부 식별자 속성이 필수 입력인 경우, 자식 엔터티들이 전체 참여하는 경우에는 '*' 기호를 붙인다.

Collection @ pknu

마지막에 속성 명칭이 적합한지를 검토한다. 이 과정에서는 사용자와 면담하고 결정하도록 한다. 이런 작업이 모두 검토되고 보완된 상태이며 업무 수행에 적합하다고 판단되면, 속성후보들을 속성들로 확정하도록 한다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

Collection @ pknu

IV. 적용사례

본 절에서는 본 연구에서 제안한 속성 도출 절차를 적용하여 속성을 도출한 사례를 제시한다. 제안 절차의 적용 사례는 'VOC(Voice of Customer) 시스템'이라는 가상적인 정보시스템을 구축한다고 했을 때 기반이 될 데이터 모델을 설계한다고 가정하고 제안 절차를 적용한 사례로, 'VOC 시스템'은 사용자요구에 대한 대응과 체계적인 관리 및 조치를 위한 정보시스템이다. 그림2는 'VOC 시스템'의 업무기술서를 나타낸다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

Collection @ pknu

우리 회사는 국내는 물론이고 일본과 미주 지역에까지 지사를 두고 있는 글로벌마케팅 기업으로, 정보시스템 사용과 관련된 각 지역 사용자들의 다양한 요구와 장애신고 등의 접수창구를 일원화하고 이러한 요구 또는 신고에 대해 신속한 대응과 사용자요구의 체계적인 관리 및 조치를 위한 서비스데스크(SD)를 운영하고자 이를 위한 VOC시스템을 개발하려고 한다.

SD는 12명의 담당자를 계약직으로 채용하여 3교대로 24시간 가동하며, 사용자들이 전화나 e메일, 또는 서비스요청 화면 입력을 통해 도움을 요청하면 SD담당자가 이를 접수하여 이미 해결방안을 갖고 있는 문제인 경우는 곧바로 해결방안을 제시하고, 다음 근무자가 다시 요청자에게 전화하여 해결여부와 서비스만족도를 확인한다. 이러한 모든 과정은 VOC시스템에 기록하여 사용자요구 발생의 경향을 분석하고 정보시스템의 부족한 부분을 찾아내어 사전에 개선함으로써 최적의 비즈니스 지원체계를 구축하고자 한다.

사용자 요구(SR)는 상담, 변경요청, 장애신고, 기타로 구분하며, 변경요청인 경우는 요청내용과 처리희망일을, 장애신고인 경우는 장애가 발생한 시점과 장애의 내용, 장애 구분 등을 알아야 한다. SR이 발생하는 대상은 정보시스템 구성 자원(Configuration Item, CI)으로, 서버, N/W, 프로그램으로 구분하여 SR에 기록되고, 접수된 SR을 처리하는 과정에서는 각각의 CI 중 어떤 것들이 변경되는지를 관리하여 CI의 변경이력까지 관리되도록 해야 한다. 처리과정에서 처리대상이 되는 CI에는 데이터를 저장하는 테이블과 같은 DB Object도 포함된다.

요청자는 우리 회사 사원들이며, 정규직원은 물론이고, 계약직원이나 임시직원들도 불편한 사항이나 장애신고 등을 할 수 있고, 매월 SR 발생과 처리에 대해 요청 부서별 집계를 통해 근본적인 개선이 필요한지 여부를 검토한다.

접수된 SR은 접수부터 처리가 완료되어 종결되는 과정이 모두 일목요연하게 관리되어야 하며, 이 과정에서 SR이 Known Error에 해당하는 경우는 SD담당자가 VOC시스템에서 해당 내용을 신속하게 검색하여 일차 조치를 해 주도록 한다. SR를 처리하는 과정은 SD담당자가 SR를 접수하여 일차적인 조치

dCollection @ pknu

로 해결되지 않으면, 다음 단계로 변경관리 담당자에 의해 SR처리를 위한 영향평가가 이루어지고, 한 번의 영향평가로 부족한 경우는 평가자를 지목하여 몇 차례 더 영향평가를 할 수 있다. 영향평가가 완료되면 이를 바탕으로 해당 프로그램의 관리 담당자가 변경대상 및 일정 등의 처리계획을 수립기록하고, 처리 후 각 계획내용의 처리완료일을 다시 기록한다. 처리가 완료된 SR은 SD담당자가 요청자에게 전화나 메일로 안내하고, 서비스만족도를 점수로 리턴받아 기록한다. SD관리자나 변경관리자는 SR의 처리과정을 일목요연하게 확인하고 예정된 일정대로 처리가 이루어지도록 관리할 수 있어야 한다.

SR처리에 의해 변경되는 CI는 언제 어떤 변경이 있었는지를 CI변경이력으로 관리해야 하며, CI의 변경이 일어나는 경우는 SR 외에도 정기PM이나 업그레이드, 기타 여러 원인이 있을 수 있다. CI 변경이 SR에 의해 일어나는 경우는 해당 SR ID가 변경 근거로써 함께 관리되어야 한다.

접수 및 처리된 SR내용은 정기적으로 재검토를 하여 사원들이 SR을 발생시키기 전에 미리 확인하여 스스로 해결이 가능하도록 FAQ로 정리하여 게재하며, FAQ는 미리 정해진 분류에 따라 빠른 검색이 가능하도록 구성한다.

그림 2. 'VOC 시스템'의 업무기술서

그림3와 같이 문장으로 기술된 업무기술서를 기준으로 선행연구의 업무중심으로 엔터티 도출 방법을 이용하여 엔터티가 완전히 도출되어 있다고 가정하고 미리 도출되어 있는 엔터티를 바탕으로 속성을 도출한다. 표3은 미리 도출된 엔터티이다.

Collection @ pknu

엔터티	분류	정의	데이터처리형태	특이사항
SR	주요	사용자가 통신도구를 통해 도움을 요청한 내용		
상담	서브	사용자가 정보시스템의 사용에 대하여 사원과 상담한 내용	SR구분=1	
변경요청	서브	사용자가 CI변경을 요청한 내용	SR구분=2	
장애신고	서브	사용자들이 정보시스템에 관련된 장애를 신고한 내용	SR구분=3	
기타	서브	사용자가 정보시스템에 대해 요청한 기타 내용	SR구분=4	
CI	핵심	정보시스템 구성요소		
서버	서브	정보시스템 구성요소인 서버	CI구분=1	
N/W	서브	정보시스템 구성요소인 N/W	CI구분=2	
프로그램	서브	정보시스템 구성요소인 프로그램	CI구분=3	
DB Object	서브	정보시스템 구성요소인 DB Object	CI구분=4	
사원	핵심	회사에 근무하는 사람		
정규직	서브	사원의 계약구분 중 하나	계약구분=1	
계약직	서브	사원의 계약구분 중 하나	계약구분=2	
임시직	서브	사원의 계약구분 중 하나	계약구분=3	
직책	핵심	사원의 담당 역할		
SD담당	서브	사원의 담당 역할 중 하나	직책명=1	
변경관리담당	서브	사원의 담당 역할 중 하나	직책명=2	
영향평가	서브	사원의 담당 역할 중 하나	직책명=3	
프로그램관리담당	서브	사원의 담당 역할 중 하나	직책명=4	
프로그램담당	서브	사원의 담당 역할 중 하나	직책명=5	
직책할당	활동	사원에게 직책을 할당하기 위한 역할		
담당	활동	계약직 사원에게 SD를 담당하게 함		
SR처리	주요	사원이 접수된 SR을 처리하는 내용		
SR접수	서브	SD담당자가 사용자의 SR을 접수한 내용	SR처리구분=1	
제시	서브	SD담당자는 SR에 대한 해결방안을 제시한 내용	SR처리구분=2	
SR평가	서브	변경관리담당자가 처리되지 않은 SR에 대해 영향평가한 내용	SR처리구분=3	
처리계획	서브	프로그램관리담당자가 영향평가를 바탕으로 처리계획을 수립한 내용	SR처리구분=4	
처리완료	서브	계획대로 처리하고 처리완료한 내용	SR처리구분=5	
KnownError	활동	정보시스템에 대한 알려진 오류		
해결방안	활동	알려진 오류에 대하여 사원이 해결한 해결방안		
CI변경이력	활동	CI를 변경한 이력정보		
SR발생	서브	사용자요청에 의한 CI변경을 의미	변경근거=1	
정기PM	서브	정기적인 사전예방에 의한 CI변경을 의미	변경근거=2	
업그레이드	서브	업그레이드에 의한 CI변경을 의미	변경근거=3	
기타	서브	기타 사유로 인한 CI변경을 의미	변경근거=4	
부족부분	활동	정보시스템에서 부족하다고 판단된 부분		
개선부분	활동	정보시스템에서 개선의 필요가 있다고 판단된 부분		
SR경향	활동	SR의 경향		
월별SR집계	활동	월별로 집계한 SR통계		
부서별SR집계	활동	부서별로 집계한 SR통계		
FAQ	활동	빈번하게 묻고 답을 한 내용		
키워드할당	활동	FAQ에 키워드를 할당하는 내용		
키워드	핵심	FAQ에 나타나는 중요 단어나 용어		
부서	핵심	회사의 조직		

표 3. 도출된 엔터티

Collection @ pknu

4.1 속성후보 도출(단계1)

속성을 도출하기 위하여 먼저 속성후보로 가능한 것들을 모두 수집하여야 한다. 그림3은 속성후보들을 수집할 수 있는 각종 경로를 나타낸 것이다.

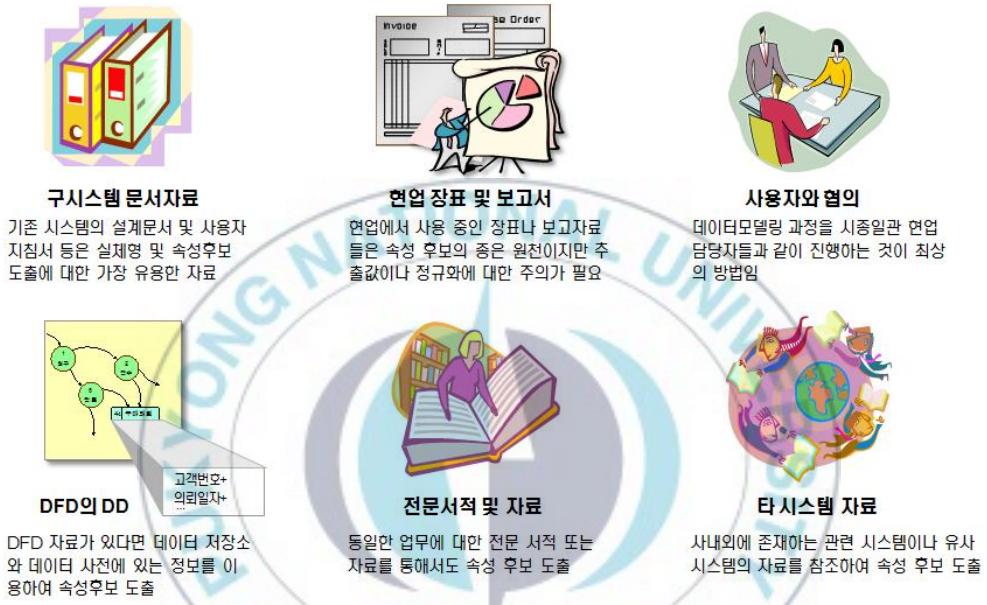

그림 3. 속성후보들을 수집 경로

1)구 시스템의 문서 자료

가동 중인 시스템의 데이터 구조 및 프로세스 명세들이 나타나 있는 설계 자료에서부터 사용자를 위한 지침서에 이르기까지 다양한 문서가 있을 것이다. 이 자료는 엔터티 후부 및 속성 후보를 도출하는 데 가장 유용하게 사용할 수 있다.

2)현업 장표/보고서

dCollection @ pknu

현업에서 사용하고 있는 각종 장표나 보고 자료들을 수집해 조사해 보는 것이다. 현업 업무 중에는 업무를 효과적으로 처리하기 위하여 많은 장표와 각종 보고 자료를 만들고 있다. 물론 이들을 그대로 속성 후보로 선정할 수 있는 것은 결코 아니다. 그 중의 상당 부분은 다른 속성에 의해 만들어질 수 있는 가공된 결과(추출 속성, Derived Value)인 경우가 많기 때문이다. 더구나 이것들은 대부분 정규화가 되어 있지 않기 때문에 정확히 파악해야만 진정한 의미의 속성 후보를 찾아낼 수 있다.

3)사용자와 협의

데이터 모델링 과정에서부터 시종일관 현업 담당자들과 같이 진행하는 것이 데이터를 모델링하는 최상의 방법이다.

4)DFD의 DD(Data Dictionary)

업무 파악 및 시스템 분석을 위한 기능 설계로 자료 흐름도가 존재한다면 여기에 있는 데이터 저장소(Data Store)와 데이터 사전(Data Dictionary)에 있는 정보를 이용하여 속성의 후보를 도출할 수 있다. 자료 흐름도를 작성할 때 생성되는 자료 저장소는 비록 테이블처럼 표현되지만 사실은 이러한 내용의 데이터가 저장되어야 한다는 데이터의 추상화된 집합이라 할 수 있으며, 그 속에 들어가야 할 구체적인 속성이 데이터 사전에 기술된다.

5) 전문 서적 및 자료

동일한 업무에 대한 전문 서적 또는 자료를 통해서도 속성의 후보를 도출할 수 있다.

6)다른 시스템 자료

Collection @ pknu

개발할 시스템의 주변을 살펴보면 사내외에 이와 관련된 시스템이나 유사한 시스템을 찾을 수 있을 것이다.

속성의 후보를 선택할 때는 최대한 충분히 수집해야 한다. 그것은 검토한 결과 자신의 속성이 아닌 것으로 판명될지라도 검토할 기회를 제공해 주는 단서라는 중요한 역할을 하기 때문이다.

수집된 여러 자료들로부터 속성후보들을 도출하고 업무적으로 유사한 성격이나 의미별로 분류하여 소그룹 풀에 저장하고, 엔터티의 성격에 맞는 특징적인 정보 항목으로 도출한다.

그리고 속성후보 도출을 위하여 미리 도출된 엔터티를 정리한다. 엔터티 도출과 확정 단계부분을 그대로 복사하고 서브타입을 가진 엔터티에 서브타입의 가장 상단에 빈 행을 추가한다. 엔터티후보를 도출하고 정제할 때, 속성 데이터로 분류되었던 속성들을 해당 엔터티의 '기존 속성후보' 란에 포함시킨다. 또한 '기존 속성후보', '6하원칙 속성후보', '인조 속성후보', '추출 속성후보' 란 등을 추가한다. 표4는 속성후보 도출할 때 필요한 컬럼명.

엔터티	서브타입	기존 속성후보	6하원칙 속성후보=원시 속성후보						인조 속성후보	추출 속성후보
			주체형	장소형	시간형	대상형	근거형	방법형		

표 4. 속성후보 도출할 때 필요한 컬럼명

여기서 6하원칙을 사용하기 때문에 어떠한 대상에 대해서도 그 내용을 정확히 설명할 수 있는 것이다. 또 관리하고자 하는 엔터티의 개체에 대해서 관리항목이 무엇인가를 구체적으로 결정하는 단계는 엔터티의 진정한 가치를 좌우한다. 그리고 '기존 속성후보'들의 성격을 파악하여 해

Collection @ pknu

당 속성후보 란으로 이동한다. 즉 '기존 속성후보'가 6하원칙 속성인지, 인조 속성인지, 추출 속성인지 파악하여 해당 란으로 이동한다. 이동된 '기존 속성후보'들은 모두 제거표시로 붉은 색 처리한다.

해당 엔터티에 대해 6하원칙을 적용하여 도출 가능한 모든 속성후보들을 도출하도록 한다. 업무에 필수적이고 명확한 속성후보들만을 도출하도록 한다. 키 엔터티에서 6하원칙 속성후보들을 도출한다는 것은 키 엔터티에서 업무에 필요하고 관리대상인 속성후보들을 도출한다는 것이다. 코드 엔터티의 6하원칙 속성후보도 키 엔터티와 마찬가지 방법으로 도출한다. 하지만 코드 인스턴스는 코드 자체를 의미하므로 '~코드' 속성은 인조 속성의 일종이지만 인조 속성이 아니라 주체형 속성에 해당한다. '~코드' 속성은 코드 인스턴스 자신을 의미하기 때문이다. 메인 엔터티나 활동 엔터티에서 6하원칙 속성후보들을 도출한다는 것은 인스턴스에 대하여 6하원칙을 적용하여 해당되는 속성후보들을 도출한다는 것이다. 교차 엔터티나 관계 엔터티에서 6하원칙 속성후보들을 도출할 때는 먼저 자신을 탄생시킨 주체에 해당하는 엔터티를 속성후보로 도출하고, 관리할 필요가 있는 속성후보들을 도출하면 된다. 집계나 분석 역할의 엔터티는 집계 용도가 목적이므로 집계의 주체가 되는 속성이 주체형 속성이 되고, 집계 대상이 되는 속성이 대상형 속성이 되므로 주의하여야 한다.

각 엔터티에서 인위적으로 생성할 인조 속성이 필요한지 검토하여, 필요한 인조 속성후보들을 추가하도록 한다. 인조 속성들은 일반적으로 식별자 도출에서 많이 생성되어 인조 식별자로 사용된다. 각 엔터티에 대하여 추출 속성후보들을 도출하도록 한다. 다른 엔터티의 속성을 상속받

Collection @ pknu

은 속성은 관계속성이 되고, 다른 엔터티(들)의 속성(들)을 복사하거나 가공하여 사용하는 속성은 추출속성이 된다. 지금까지 수형한 결과를 표 5에 나타내었다.

엔터티	서브타입	기존 속성후보	6하원칙 속성후보= 원시 속성후보						인조 속성후보	추출 속성후보
			주체형	장소형	시간형	대상형	근거형	방법형		
부서		지역	부서명	지역				주업무		
사원,사원구분			사원명,주민등록번호	근무서				사원구분		
	정규직									
	계약직									
	임시직									
직책,직책구분			직책명					직책구분,업무내용		
	SD담당									
	변경관리담당									
	영향평가									
	프로그램관리담당									
	프로그램담당									
직책할당			사원		시작일자,종료일자	직책				
담당			계약직사원		시작일자,종료일자	SD담당				
CI변경이력,변경근거		변경일자,변경사유			변경일시	대상CI	변경근거	변경내용		
	SR발생	SR_ID								SR_ID
	정기PM									
	업그레이드									
	기타									
CI,CI구분								CI구분		
	서버		서버명							
	N/W		N/W명							
	프로그램		프로그램명							
	DB객체		DB객체명							
Known Error					등록일자	대상CI		오류내용		
해결방안					등록일자	알려진오류		해결내용		
FAQ			게시자		등록일자	대상CI		질문, 답변		
FAQ키워드			FAQ			워드				
SR,SR구분		통신도구,SR_ID	요청자		요청일자	대상CI		SR구분,요청방법	SR_ID	SR종료일자
	상담				상담희망일자			상담내용		
	변경요청	요청내용,처리희망일			처리희망일자			요청내용		
	장애신고	장애발생시점,장애내용,장애구분			장애발생일시			장애내용,장애구분		
	기타요구							내용		
SR처리,SR처리구분								SR처리구분		
	SR접수		SD담당자		접수일자	대상SR				
	제시		SD담당자		제시일자	대상SR,해결방안	SR접수			
	SR평가		평가자		평가일자	대상SR	SR접수	평가내용		
	처리계획	변경대상,일정,처리완료일	수립자		수립일자,처리예정일자	대상SR,대상CI	SR접수	처리내용		
	처리완료	처리여부,만족도	SD담당자		등록일자,처리완료일자	대상SR,대상CI	SR접수	해결여부,서비스만족		

표 5. 속성후보 도출 결과

Collection @ pknu

4.2 속성후보 정제(단계2)

단계1에서 속성이 될 가능성이 있다면 모두 속성후보로 도출하기 때문에 불필요한 후보가 생길 수 있고 이를 정제하는 작업이 필요하다. 단계2에서는 기존 속성후보 란을 삭제하고, 속성후보들을 정제할 준비한다. 서브타입들이 공통적으로 가지고 있는 속성후보를 슈퍼타입으로 이동시킨 다음, 이동된 속성후보들은 제거 표시로 붉은색 처리한다.

각 속성후보에 대하여 속성으로서의 자격이 있는지 검토하여야 한다. 속성후보가 엔터티의 성격이 있는지, 속성후보의 값이 변하는지 또는 여러 개의 값을 가지는지, 속성후보에 대하여 분리와 결합이 가능한지 등을 검토하여 속성후보들을 정제하도록 한다.

먼저, 엔터티 성격의 속성후보들을 파악하여 속성후보가 엔터티 성격이라고 판단되면, 해당 엔터티의 존재여부를 확인하고, 해당 엔터티가 존재하지 않는다면, 엔터티를 추가 도출한다. 그리고 속성후보 명칭 앞에 해당 엔터티를 점과 함께 표기하고, 해당 속성후보가 완전하게 정제되지 않았음을 의미하는 녹색으로 처리한다.

다음으로, 속성후보의 값이 변하거나 한 시점에 여러 개의 속성 값을 가지고 있을 때, 자신을 제거하고, 자신을 포함하는 자식 엔터티에게 자신의 식별자를 상속해줌으로써 이를 해결한다.

그 다음으로, 각 속성후보들을 검토하면서, 속성후보가 업무적으로 효율적이라면 엔터티들을 통합 한다.

마지막으로, 하나의 속성후보가 두 개 이상의 속성으로 구성되어 있는

Collection @ pknu

경우는 분리하도록 한다. 그리고 너무 상세하게 분리된 속성후보들이 존재한다면, 통합 가능한 속성후보들을 필요하다면 통합한다. 각 서브타입에서의 명칭들은 실제 값으로 저장되는 서브타입 속성은 아니지만, 해당 서브타입 인스턴스의 명칭이 되므로 가상 속성(Pseudo Attribute)이 된다. 또한 제거되어야 하는 속성후보가 있으면 붉은 색 처리하여 제거하도록 한다.

지금까지 정제한 결과를 표6에 나타내었다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

Collection @ pknu

엔터티	서브타입	6하원칙 속성후보= 원시 속성후보						임조 속성후보	추출 속성후보
		주체형	장소형	시간형	대상형	근거형	방법형		
부서		부서명	지역				주업무		
사원,사원구분		사원명,주민등록번호	부서,근무부서				사원구분		
	정규직								
	계약직								
	임시직								
직책,직책구분		직책명					직책구분,업무내용		
	SD담당								
	변경관리담당								
	영향평가								
	프로그램관리담당								
	프로그램담당								
직책할당		사원,사원		시작일자,종료일자	직책,직책				
담당		사원,계약직사원		시작일자,종료일자	직책,SD담당				
CI변경이력,변경근거				변경일시	CI,대상CI	변경근거	변경내용		
	SR발행								SR,SR_ID
	정기PM								
	업그레이드								
	기타								
CI,CI구분							CI구분		
	서버	서버명							
	N/W	N/W명							
	프로그램	프로그램명							
	DB객체	DB객체명							
Known Error				등록일자	CI,대상CI		오류내용		
해결방안				등록일자	KnownError,알려진오류		해결내용		
FAQ		게시자		등록일자	CI,대상CI		질문, 답변		
FAQ키워드		FAQ			워드				
SR,SR구분		사원,요청자		요청일자	CI대상CI		SR구분,요청방법	SR_ID	SR종료일자
	상담			상담희망일자			상담내용		
	변경요청			처리희망일자			요청내용		
	장애신고			장애발생일시			장애내용,장애구분		
	기타요구						내용		
SR처리,SR처리구분		사원,처리자			SR,대상SR	(SR처리,SR처리구분(SR접수)),SR접수	SR처리구분		
	SR접수	사원,SD담당자(p)		접수일자(p)	대상SR				
	제시	사원,SD담당자(p)		제시일자(p)	대상SR,해결방안	SR접수			
	SR평가	사원,평가자(p)		평가일자(p)	대상SR	SR접수	평가내용		
	처리계획	사원,수립자(p)		수립일자(p),처리예정일자	대상SR,대상CI	SR접수	처리내용		
	처리완료	사원,SD담당자(p)		등록일자(p),처리완료일자	대상SR,대상CI	SR접수	해결여부,서비스만족		

표 6. 속성후보 정제 결과

dCollection @ pknu

4.3 식별자 도출 및 확정(단계3)

단계2에서 도출된 속성후보들에 대하여 정제 과정이 모두 끝났으면, 정제된 각 속성후보에 대하여 필수하거나 선택하거나 결정하도록 한다. 필수 속성이라고 판단되는 속성후보는 명칭 앞에 '*' 기호를 표기하고, 선택 입력의 속성후보는 어떠한 표기도 하지 않는다. 그리고 각 엔터티에 대하여 본질 식별자, 실질 식별자 및 대체 식별자를 도출하여 확정한다.

속성후보의 값이 NULL이라고 가정하였을 때, 업무 처리나 데이터 처리에 전혀 문제가 없는 경우, 그리고 값을 얻거나 구할 수 없는 경우가 한 번이라도 발생할 수 있다면, 해당 속성후보는 선택속성이 된다. 하지만 속성값이 있어야만 업무가 처리되거나 데이터 처리가 가능하다면, 해당 속성은 필수 속성이 된다. 이와 같은 판단은 업무적인 측면에서 결정되어야 하고, 모호한 경우가 있다면, 실무자나 사용자와 면담하여 속성값을 확인하고 필수/선택을 결정한다.

슈퍼타입 속성의 필수/선택 여부는 모든 서브타입들에 적용되어야 하므로 소속되는 모든 서브타입들에서 값을 가져야 하는지를 검사하여 결정하도록 한다. 서브타입 속성의 필수/선택 여부는 자신의 서브타입 내에서만 통용되므로 자신의 내부에서만 필수/선택 여부를 판단하면 된다. 따라서 필수/선택이 결정된 속성후보에 대하여, 속성의 자격이 충분하면 속성으로 확정하는데 녹색 처리된 부분은 제외한다.

대부분의 속성들이 확정되었으면, 엔터티들의 인스턴스들을 식별할 수 있는 각종 식별자를 도출하고 확정하도록 한다. 식별자는 본질식별자, 실질식별자, 대체식별자등 여러 가지 형태의 식별자 있다. 본질식별자는

Collection @ pknu

인스턴스가 탄생하는데 필요한 속성들로써 인스턴스들을 식별하는 역할을 한다. 속성명 앞에 '<#>' 기호를 표기한다. 본질 식별자 속성들을 올바르게 도출해야 실질 식별자 속성들을 올바르게 도출할 수 있으므로, 식별자들 중에서 가장 먼저 도출되어야 한다. 상위 엔터티들이 하위 엔터티들에게 영향을 미치기 때문에 키 엔터티, 코드 엔터티, 메인 엔터티, 키 엔터티에 관련되니 활동 엔터티, 메인 엔터티에 관련된 활동 엔터티 순으로 본질 식별자 속성들을 도출하고 확정하는 것이 효율적이다. 도출 방법을 예를 들면, '부서', '사원', 등의 키 엔터티들은 탄생번호를 본질 식별자로 사용하는 것이 적합하기 때문에 '부서번호', '사원번호' 등의 탄생번호라는 인조 식별자를 생성하여 각 엔터티의 본질 식별자로 사용한다.

실질 식별자는 인스턴스들을 실제로 식별하고 관리하기 위한 목적으로 사용한다. 속성명 앞에 '#' 기호를 붙여 표기한다. 구성방법으로는 세 가지가 있는데, 본질 식별자 속성의 전부 또는 일부로 실질 식별자를 구성하는 방법, 인조 식별자를 내세워 실질 식별자로 구성하는 방법, 본질 식별자 속성 일부와 인조 식별자를 사용하여 실질 식별자를 구성하는 방법이다. 실질식별자도 본질 식별자와 마찬가지로 키 엔터티, 코드 엔터티, 키 엔터티 관련 엔터티, 메인 엔터티 관련 엔터티 순으로 도출하고 확정하는 것이 효율적이다. 도출 방법을 예로 들자면, '직책' 엔터티는 본질 식별자인 '직책명' 속성만으로도 실질 식별자의 자격이 충분하지만, 직책 인스턴스들은 직책들을 분류하는 성격이 강하기 때문에 관리의 편리를 위하여 '직책코드'라는 인조 식별자를 추가 생성하여 실질 식별자로 사용하기로 한다.

Collection @ pknu

대체 식별자는 실질 식별자 대신 대체 식별자를 상속받기를 원하는 자식 엔터티가 있는 경우와 서브타입 인스턴스들을 식별할 필요가 있을 경우, 그리고 인스턴스들의 유일성을 확보해야 할 경우에 주로 사용한다. 특히 본질 식별자가 실질 식별자가 되지 못한 경우에 본질 식별자 속성들의 유일성을 확보하기 위해 본질 식별자를 대체 식별자로 구성할 수 있다. 대체 식별자는 속성명 앞에 '(#)' 기호를 붙여 표기한다. 대체 식별자는 실질 식별자 역할을 하지만, 엄연히 실질 식별자와는 구별된다. 실질 식별자는 구성되는 모든 속성들이 필수 입력이어야 하지만, 대체 식별자의 속성들은 선택 입력이 될 수 있으며, 실질 식별자는 기본키 제약(PKC; Primary Key Constraint)을 가지지만, 대체 식별자는 유일키 제약(UKC; Unique Key Constraint)을 가진다.

표7은 사례의 단계3 결과를 나타낼 것이다.

Collection @ pknu

엔터티	서브타입	6하원칙 속성후보= 원시 속성후보						인조 속성후보	추출 속성 후보
		주체형	장소형	시간형	대상형	근거형	방법형		
부서		*부서명	지역				주업무	#<#>*부서번호	
사원,사원구분		*사원명,(#)주민등록번호	*부서 근무부서				*사원구분	#<#>*사원번호	
	정규직								
	계약직								
	임시직								
직책,직책구분		<#>*직책명					직책구분,업무내용	#*직책코드	
	SD담당								
	변경관리담당								
	영향평가								
	프로그램관리담당								
	프로그램담당								
직책할당		#<#>*사원,사원		#<#>*시작일자,#<#>*종요일자	#<#>*직책,직책				
담당		#<#>*사원,계약직사원		#<#>*시작일자,#<#>*종요일자	#<#>*직책,SD담당				
CI변경이력,변경근거				<#>*변경일시	#<#>*CI,대상CI	*변경근거	<#>*변경내용	#*변경번호	
	SR발생								*SR,SR_ID
	정기PM								
	업그레이드								
	기타								
CI,CI구분		*CI명					*CI구분	#<#>*CI_ID	
	서버	*서버명(p)							
	N/W	*N/W명(p)							
	프로그램	*프로그램명(p)							
	DB객체	*DB객체명(p)							
Known Error				등록일자	#<#>*CI,대상CI		<#>*오류내용	#*오류번호	
해결방안				등록일자	#<#>*KnownError,알려진오류		<#>*해결내용	#*해결방안번호	
FAQ		<#>*사원,게시자		등록일자	#<#>*CI,대상CI		<#>*질문,<#>*답변	#*FAQ번호	
FAQ키워드		#<#>*FAQ			#<#>*워드				
SR,SR구분		<#>*사원,요청자		<#>*요청일자	<#>*CI,대상CI		*SR구분,요청방법	#*SR_ID	SR종료일자
	상담			*상담희망일자			*상담내용		
	변경요청			*처리희망일자			*요청내용		
	장애신고			*장애발생일시			*장애내용,장애구분		
	기타요구						*내용		
SR처리,SR처리구분		<#>사원,처리자		<#>처리일자	#<#>*SR,대상SR	SR접수	*SR처리구분	#*SR처리번호	
	SR접수	사원,SD담당자(p)		*접수일자(p)					
	제시	사원,SD담당자(p)		제시일자(p)	*해결방안				
	SR평가	*사원,평가자(p)		*평가일자(p)			*평가내용		
	처리계획	*사원,수립자(p)		*수립일자(p)					
	처리완료	사원,SD담당자(p)		등록일자(p)					

표 7. 식별자 도출과 확정 결과

Collection @ pknu

4.4 속성 확정(단계4)

전 3단계로 인해 엔터티에 대한 속성들을 대부분 확정하였지만, 확정되지 않은 속성후보들이 있었다. 이러한 속성후부들은 부모 엔터티로부터 대부분 상속받은 속성들이다. 확정되지 않은 속성후부들을 확정하고, 외래 식별자들, 본질 식별자, 실질 식별자들 및 대체 식별자들을 확정하도록 한다.

부모 엔터티로부터 상속받은 속성후보들에 대하여 먼저 검토한다. 이러한 각각의 속성후보에 대하여 기존 엔터티 중에서 부모 역할을 할 수 있는 엔터티가 있으면 찾아 부모 엔터티로 하고, 적당한 엔터티가 없는 경우에는 추가로 생성하여 부모 엔터티로 확정한다. 그리고 확정된 부모 엔터티에 대하여 자식에서의 부모 엔터티의 역할을 자식관점 관계명으로 결정하고, 부모 엔터티의 명칭과 관계명을 '-' 기호로 연결하여 속성명 뒤에 또는 속성명 다음 줄에 관호로 묶어 녹색 처리한다.

부모 엔터티로부터 식별자를 상속받은 속성후부들을 모두 외래 식별자 속성으로 확정되어야 한다. 따라서 이들에 대하여 외래 식별자 속성으로 사용할 수 있는 직절한 명칭을 부여하여 흑색 처리하고 외래 식별자 속성으로 확정한다.

외래 식별자까지 확정된 상태이므로 속성까지 포함하여 나머지 식별자인 본질 식별자, 실질 식별자 및 대체 식별자 등을 모두 확정할 수 있다. 이미 이전의 속성 도출과 확정 과정에서 이들 식별자들이 정제만 되지 않았을 뿐 모두 도출된 상태이다. 속성후부들이 정제되어 외래 식별자로 모두 확정이 되면 자동적으로 모든 식별자 속성을 확정된다. 표8은 속성 확정 결과를 나타낼 것이다.

Collection @ pknu

엘터티	서브타입	6하원칙 속성후보= 원시 속성후보						인조 속성후보	추출 속성후보
		주체형	장소형	시간형	대상형	근거형	방법형		
부서		*부서명	지역				주업무	#(#)*부서번호	
사원,사원구분		*사원명,(#)주민등록번호	*근무부서번호(부서-근무부서로서)				*사원구분	#(#)*사원번호	
	정규직								
	계약직								
	임시직								
직책,직책구분		(#)*직책명					직책구분,업무내용	#*직책코드	
	SD담당								
	변경관리담당								
	영향평가								
	프로그램관리담당								
	프로그램담당								
직책할당		#(#)*사원,사원(사원-직책할당사원으로서)		#(#)*시작일자,#(#)*종료일자	#(#)*직책,직책(직책-할당직책으로서)				
CI변경이력,변경근거				(#)*변경일시	#(#)*대상CI_ID(CI-변경대상CI로서)	*변경근거	(#)*변경내용	#*변경번호	
	SR발생								*SR,SR_ID(SR-CI변경원인으로서)
	정기PM								
	업그레이드								
	기타								
CI,CI구분		*CI명					*CI구분	#(#)*CI_ID	
	서버	*서버명(p)							
	N/W	*N/W명(p)							
	프로그램	*프로그램명(p)							
	DB객체	*DB객체명(p)							
KnownError				등록일자	#(#)*대상CI_ID(CI_오류대상CI로서)		(#)*오류내용	#*오류번호	
해결방안				등록일자	#(#)*대상CI_ID,#(#)*오류번호(KnownError,알려진오류로서)		(#)*해결내용	#*해결방안번호	
FAQ		(#)*사원,게시자		등록일자	#(#)*대상CI_ID(CI-FAQ대상CI로서)		(#)*질문,(#)*답변	#*FAQ번호	
FAQ키워드		#(#)*FAQ			#(#)*워드				
SR,SR구분		(#)*요청자,사번(사원,요청자로서)		(#)*요청일자	(#)*대상CI_ID(CI-변경대상CI로서)		*SR구분,요청방법	#*SR_ID	SR종료일자
	상담			*상담희망일자			*상담내용		
	변경요청			*처리희망일자			*요청내용		
	장애신고			*장애발생일시			*장애내용,장애구분		
	기타요구						*내용		
SR처리,SR처리구분		(#)처리자,사번(사원,처리자로서)		(#)처리일자	#(#)*대상SR_ID(SR-처리대상SR로서)		*SR처리구분	#*SR처리번호	
	SR접수	SD담당자 번호(p)(사원-SR접수자로서)		*접수일자(p)					
	제시	SD담당자 번호(p)(사원-해결방안제시자로서)		제시일자(p)	*해결방안				
	SR평가	*평가자 사번(p)(사원-SR평가자로서)		*평가일자(p)			*평가내용		
	처리계획	*수립자 번호(p)(사원-처리계획수립자로서)		*수립일자(p)					
	처리완료	SD담당자 번호(p)(사원-처리완료등록자로서)		등록일자(p)					

표 8. 속성 확정 결과

Collection @ pknu

속성들이 확정되었으면, ‘엔터티’, ‘속성’, ‘실/보’, ‘본질’, ‘필/선’, ‘외부’, ‘정의’ 란들을 가진 시트를 준비하여 속성정의서를 작성하도록 한다.

먼저 각 엔터티에 속성들을 포함시킨다. 해당 속성의 정의를 ‘정의’ 란에 기재한다. 속성의 개념과 의미를 명확하게 파악할 수 있도록 속성을 정의하여야 하며, 명칭만으로 충분하지 못한 부분도 함께 기재한다. 그리고 실질 식별자 속성은 ‘실/보’ 란에 ‘#’ 기호를 표기하고, 대체 식별자 속성은 ‘실/보’ 란에 ‘(#)’ 기호를 표기하도록 한다. 또한 본질 식별자 속성은 ‘본질’ 란에 ‘<#>’ 기호를 표기한다. 필수 속성은 ‘필/선’ 란에 ‘*’ 기호를 표기하고, 선택 속성은 ‘필/선’ 란을 공란으로 둔다. 외래 식별자는 ‘외부’ 란에 ‘FK’ 를 기재한다.

표9는 사례의 속성정의서이다.

Collection @ pknu

엔터티	속성	설/보	본질	필/선	외부	정의
개선부분	개선내용			*		CI를 개선한 내용
	개선번호	#		*		CI 개선 건수의 번호
	개선자사번		<#>		FK	CI를 개선한 사원의 사번
	개선일자		<#>			CI를 개선한 일자
	부족부분번호		<#>	*	FK	개선 대상인 CI의 부족부분 번호
담당	시작일자		<#>	*		계약직 사원이 SD 담당을 시작한 일자
	종료일자					계약직 사원이 SD 담당을 종료한 일자
	계약직사번	#	<#>	*	FK	계약직 사원의 사번
	SD담당	#	<#>	*		SD담당 직책
부서	부서명			*		부서의 이름
	부서번호	#	<#>	*		부서의 고유번호
부서별SR집계	발생건수			*		부서에서 발생한 SR을 집계한 건수
	부서명	#	<#>	*		부서의 이름
	처리건수			*		부서에서 발생한 SR을 처리한 건수
부족부분	부족번호	#		*		CI의 부족한 건의 고유번호
	부족내용		<#>	*		CI의 부족한 내용
	대상CI_ID		<#>	*	FK	부족하다고 판단된 CI의 ID
사원	계약구분			*		사원의 계약을 구분
	사원명		<#>	*		사원의 이름
	사원번호	#		*		사원의 고유번호
	주민등록번호	(#)	<#>	*		사원의 주민등록번호
	근무부서번호			*		사원이 근무하는 부서의 번호
월별SR집계	년월	#	<#>	*		년월
	발생건수			*		년월 단위로 SR이 발생한 건수
	처리건수			*		년월 단위로 발생한 SR을 처리한 건수
직책	업무내용					직책의 업무내용
	직책명	#	<#>	*		직책의 이름
직책할당	시작일자		<#>	*		직책의 담당을 시작한 일자
	종료일자					직책의 담당을 종료한 일자
	사번	#	<#>	*	FK	직책이 할당된 사원의 사번
	직책명	#	<#>	*	FK	사원에 할당된 직책명
키워드	워드	#	<#>	*		검색하기 위한 단어나 용어
키워드할당	FAQ_ID	#	<#>	*	FK	키워드가 할당된 FAQ의 ID
	키워드	#	<#>	*	FK	FAQ에 할당된 키워드
해결방안	해결내용			*		알려진 오류를 해결한 내용
	해결방안번호	#		*		해결방안의 고유번호
	담당자사번		<#>	*	FK	알려진 오류를 해결한 사원의 사번
	오류번호		<#>	*	FK	해결한 오류의 번호
CI	서버_서버명			*		서버 이름
	프로그램_프로그램명			*		프로그램의 이름
	CI_ID	#	<#>	*		CI의 고유번호
	CI구분			*		CI의 구분
	DB_Object_DB객체명			*		DB객체의 이름
	N/W_N/W명			*		N/W의 이름

dCollection @ pknu

CI변경이력	변경근거		<#>	*		CI를 변경한 근거
	변경내용			*		CI를 변경한 내용
	변경번호	#		*		CI를 변경한 건의 고유번호
	변경일자					CI를 변경한 일자
	변경자사번		<#>	*	FK	CI를 변경한 사원의 사번
	대상CI_ID		<#>	*	FK	변경한 CI의 ID
	SR발생 ,SR_ID			*	FK	CI변경의 원인이 되는 SR의 ID
FAQ	게시일자					FAQ를 게시한 일자
	답변					질문에 대하여 답변한 내용
	질문		<#>	*		질문한 내용
	FAQ_ID	#		*		FAQ의 고유번호
	게시자사번		<#>	*	FK	FAQ 답변을 게시한 사원의 사번
	대상CI_ID		<#>		FK	FAQ에 해당하는 CI의 ID
KnBKwnErrBKr	오류내용		<#>	*		오류의 내용
	오류번호	#		*		오류의 고유번호
	대상CI_ID		<#>	*	FK	알려진 오류의 대상인 CI의 ID
SR	기타 내용					요청한 기타 내용
	상담 ,상담희망일자					상담을 희망한 일자
	변경요청 ,요청내용			*		요청한 내용
	요청방법		<#>	*		요청한 방법
	요청일자		<#>	*		요청한 일자
	장애신고 ,장애구분			*		장애의 구분
	장애신고 ,장애내용			*		장애의 내용
	장애신고 ,장애발생시점					장애가 발생한 시점
	변경요청 ,처리희망일자					요청한 내용의 처리를 희망한 일자
	SR_ID	#		*		요청한 내용의 고유번호
	SR구분			*		요청한 내용의 구분
	요청자사번		<#>	*	FK	SR을 요청한 사원의 사번
	대상CI_ID		<#>	*	FK	SR의 대상인 CI의 ID
SR경향	경향		<#>	*		요청한 내용의 경향
	경향번호	#		*		요청한 내용을 분석한 건의 고유번호
	분석일자					요청한 내용을 분석한 일자
	분석자사번		<#>	*	FK	SR의 경향을 분석한 사원의 사번
	대상CI_ID		<#>	*	FK	SR경향의 대상인 CI의 ID
SR처리	처리완료 ,서비스만족도					요청처리를 서비스한 만족도
	SR접수 ,접수일자					요청처리를 접수한 일자
	제시 ,제시일자					요청처리의 해결방안을 제시한 일자
	처리계획 ,처리내용			*		요청을 처리한 내용
	처리번호	#		*		요청처리의 고유번호
	처리계획 ,처리예정일자					요청의 처리예정 일자
	처리계획 ,처리완료일자					요청처리를 완료한 일자
	SR평가 ,평가내용			*		요청을 평가한 내용
	SR평가 ,평가순번	(#)		*		하나의 요청에 대하여 평가한 번호
	SR평가 ,평가일자					요청을 평가한 일자
	처리완료 ,해결여부			*		요청처리를 해결한 여부
	SR처리구분		<#>	*		요청처리의 구분
	대상SR_ID		<#>	*	FK	처리하는 SR의 ID

dCollection @ pknu

	SB접수_SD담당자사번			*	FK	SB을 접수한 사원의 사번
	제시_SD담당자사번			*	FK	SB의 해결방안을 제시한 사원의 사번
	제시_해결방안번호			*	FK	SB의 해결방안 번호
	SB평가_평가자사번			*	FK	SB을 평가한 사원의 사번
	SB평가_대상CI_ID			*	FK	SB평가의 대상인 CI의 ID
	SB평가_SB접수번호			*	FK	SB을 접수한 번호
	처리계획_처리자사번			*	FK	SB을 처리한 사원의 사번
	처리계획_변경대상CI_ID			*	FK	SB을 처리할 때 변경하는 CI의 ID
	처리완료_SD담당자사번			*	FK	SB 완료를 처리한 사원의 사번

표 9. 사례의 속성정의서

각 단계를 진행하면서 사용자 면담을 통하여 중간 과정을 확인할 수 있으며, 최종적으로 도출된 속성에 대하여 사용자의 검증이 필요하다.

Collection @ pknu

V. 제안 절차의 검증

본 연구에서 제안한 절차의 검증을 위한 현업에서 전문 IT 컨설턴트로 활동 중인 데이터 모델링 전문가들이 멘토가 도출한 속성을 모범 답안으로 하여 학생들이 도출한 속성과의 유사도검사로 하였다. 왜냐하면 현실세계의 업무에 대한 데이터 모델링이나 방법이나 결과는 정답이 있을 수 없기 때문이다. 그러므로 이를 통해 제안 절차를 이용하여 모범 답안에 근접하게 속성을 도출할 수 있다면 제안한 절차가 타당성을 있다는 사실을 알게 된다.

본 연구에서 제안한 절차의 검증을 위하여 학부생 및 대학원생을 대상으로 본 연구에서 제안한 절차를 적용하도록 하였다. 그리고 참여하는 학생들을 총 25개 팀으로 구성해가지고, 각 팀별로 가상의 업무기술서를 바탕으로 본 연구에서 제안한 절차를 적용하도록 하였다. 각 팀별로 전문 IT 컨설턴트로 활동중인 데이터 모델링 전문가들이 멘토의 역할로 참여하여 절차적인 진행하도록 하다. 마지막으로 각 팀에서 학생들이 도출한 속성과 멘토가 모범 답안으로 도출한 속성 간의 유사도를 검사하였다. 모범 답안의 각 속성후보가 학생들이 본 제안한 절차로 도출한 결과에서 몇 퍼센트 존재하는지를 검사한 것이다.

같은 의미를 가진 속성들에 대해서 멘토와 학생들이 사용하는 용어가 다를 수 있기 때문에 이를 일치시키기 위한 표준화 작업을 선행한다. 그 다음 멘토 및 학생들이 도출한 모든 속성들을 원시속성, 인조속성, 실질식별자, 본질식별자, 필수속성, 선택속성으로 분류한 후 각 부분별로 EXCEL안의 한 열에 넣고 비교를 통해 분석한다. 이때 EXCEL중에서

Collection @ pknu

COUNTIF함수를 이용하여 양쪽의 데이터를 비교한다. 먼저 멘토가 도출한 속성들을 A열에 넣고, 학생들이 도출한 속성들을 B열에 놔둔 후 COUNTIF(A열전체범위, B1)공식을 이용하여 학생들이 도출한 B1속성들을 멘토가 도출한 속성들에 소속되어 있는지 판단된다. 반복적으로 COUNTIF 함수를 사용하여 학생들이 도출한 속성들이 멘토가 도출한 속성들과 얼마나 일치하는지에 대한 유사도를 구할 수 있다. 이와 같은 방법을 통해 6가지 속성부분의 백분율을 구한 뒤 마지막에 전체 속성 유사도까지 구할 수 있다.

검사 결과, 원시 속성 간의 평균유사도가 87%, 인조 속성 간의 평균유사도가 88%, 실질식별자 간의 평균유사도가 88%, 본질식별자 간의 평균유사도가 81%, 필수속성 간의 평균유사도가 77%, 선택속성 간의 유사도가 78%로 나타났고, 전체 속성 간의 평균유사도는 83%의 유사도를 보였다.

표10은 학생들이 도출한 속성과 모범 답안 간의 유사도검사 결과를 보여 준다.

Collection @ pknu

Team Number	원시속성 Similarity	인조속성 Similarity	실질식별자 Similarity	본질식별자 Similarity	필수속성 Similarity	선택속성 Similarity	전체속성 Similarity
1	95%	92%	91%	80%	88%	84%	88%
2	88%	90%	96%	87%	86%	85%	89%
3	91%	89%	89%	81%	75%	68%	82%
4	93%	88%	86%	85%	76%	78%	84%
5	88%	84%	78%	81%	85%	78%	82%
6	87%	88%	88%	75%	75%	65%	80%
7	94%	90%	95%	90%	65%	75%	85%
8	86%	90%	96%	87%	88%	78%	88%
9	85%	88%	89%	80%	76%	72%	82%
10	80%	85%	87%	65%	65%	76%	76%
11	82%	86%	75%	95%	78%	85%	84%
12	85%	84%	80%	80%	65%	66%	77%
13	80%	75%	76%	77%	55%	87%	75%
14	92%	95%	91%	85%	85%	87%	89%
15	76%	80%	72%	74%	65%	73%	73%
16	85%	87%	88%	85%	75%	83%	84%
17	88%	90%	95%	77%	88%	86%	87%
18	84%	91%	95%	85%	82%	86%	87%
19	90%	92%	96%	90%	74%	68%	85%
20	86%	92%	98%	80%	78%	76%	85%
21	82%	86%	88%	75%	79%	74%	81%
22	94%	96%	90%	85%	76%	88%	88%
23	80%	85%	70%	75%	76%	73%	77%
24	82%	79%	85%	76%	86%	82%	82%
25	91%	92%	95%	85%	88%	81%	89%
Average	87%	88%	88%	80%	78%	80%	83%

표 10. 유사도 검사 결과

Collection @ pknu

Ⅵ. 결론

본 연구에서는 미리 도출된 대상업무 및 엔터티를 배경으로, 데이터모델링 초보자나 미숙련자들을 중심으로 용이하게 속성을 도출하는 절차를 제안하였다.

본 연구에서 제안한 절차는 구체적이며 적합한 절차를 제공하여 속성들을 도출한다는 것이다. 가상의 'VOC시스템'이라는 가상의 업무용 정보시스템의 대상업무로부터 속성을 도출하는 과정을 보였으며, 학생들을 대상으로 팀을 구성하고 본 연구에서 제안한 절차로 가상 업무중심으로 속성을 도출하게 하였다. 그리고 나서 학생들이 도출한 속성과 전문 IT 컨설턴트로 멘토가 도출한 모범 답안 간의 유사도검사를 하였다. 결과가 모범 답안에 전체 속성을 대상으로 83%의 유사도를 나타냈으며, 이 유사도를 통하여 본 연구의 제안한 절차를 이용하여 모법 답안에 상당히 근접하게 속성을 도출할 수 있다는 것을 확인하였다.

따라서 본 연구에서 제안한 절차를 활용한다면 데이터모델링에 실무경험이 없는 초보가나 미숙련자가 적용하여도 속성을 도출할 수 있음을 보였다. 물론 본 연구의 제안한 절차를 이용하여 속성을 도출하는 소요기간은 다소 긴 것으로 나타났다. 그런데 실무경험이 없는 초보자나 미숙련자가 데이터모델링 설계하지 못하는 것이 보다 이렇게 많은 시간을 투자하는 것은 아니라고 판단된다. 뿐만 아니라 제안한 절차를 잘 파악하고 연습하여 익숙해진다면 그렇게 많은 소요시간이 반드시 단축할 수 있을 것이다.

본 연구에서 제안한 절차는 데이터모델링 초보자나 미숙련자들을 중심

Collection @ pknu

으로 제안하지만 실무에서도 큰 도움을 가져올 수 있을 것으로 기대한다.

또한 데이터 모델 설계는 여러 과정으로 거쳐서 진행하는 작업인데, 제안 절차에서 도출 된 결과를 이용하여 데이터 모델 설계의 이후 과정인 관계 도출 과정을 진행할 수 있을 것으로 기대한다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

Collection @ pknu

VII.참고문헌

[1] Hwasik Lee, “Data Architecture Solution I”, En-core, 2003

[2] Hwasik Lee, Hyunsan Kim, Haesul Yang, “Expression Method and Application Plan of the Entity for a Data Modeling”, Korea Information Processing Society, vol. 12, no 2, 2005

[3] Jeongmo Yeo, Sunyi Park, “A Study on Elicitation Method of Target Businesses for Application Development”, Korea Institute of Information and Communication Engineering, vol 15, no 12, pp.2599-2608, 2012

[4] Jan Speelpenning, “Data Modeling and Relational Database Design”, Oracle, 2001

[5] Steve Hoberman, “Data Modeling Master Class”, Technics Pubn Llc, 2011

[6] Michael C. Reingruber, “The Data Modeling Handbook”, Intervision, 2002

[7] Jonghyun Kim, “Information Engineering and Data Modeling”, Korean Institute of Information Scientists and Engineers, vol 12, no 11, 1994

[8] Korea Database Agency, “Data Architecture Professional Guide”, Korea Database Agency, 2010

[9] Graeme Simsion, “Data Modeling Essentials”, MorganKaufmannPub, 2005

Collection @ pknu

[10] Doyu Kim, Jeongmo Yeo, “A Study on Elicitation Procedures of the Entity for Data Model”, Korea Information Processing Society, Vol. 2, No.7, 2013

[11] Ki Chang Kim “Relational Data Modeling Note”, Openmade consulting, 2014

[12] Carlis, John, and Joesph Maguire, ‘Mastering Data Modeling”, Addoson-Wesley, 2000

Collection @ pknu