

IEEE-1471에 기반한 엔터프라이즈 아키텍처의 관점 자동 생성

솔루션링크 김형호 · 민상윤

1. 서 론

엔터프라이즈 아키텍처(Enterprise Architecture, 이하 EA)[9,11]는 정보 시스템의 효율적인 통제를 위한 기반을 제공하는 기법으로, 많은 조직들의 핵심 업무들이 정보 시스템에 기반하여 구축됨에 따라서 공공 부문을 포함하여 널리 적용되고 있다. 일반적으로 EA에서는 다양한 이해 당사자들의 관점을 효율적으로 지원하기 위해서 구성되는 모델들을 각 원칙에 따라서 비즈니스 아키텍처(business architecture), 어플리케이션 아키텍처(application architecture), 정보 아키텍처(information architecture), 기술 아키텍처(data architecture) 등으로 분할하여 기술되는 경우 [8]가 많다. 이러한 경우에 각 아키텍처 간에 정보들이 일관되게 유지되어야 함으로 이를 하나의 모델로 관리하는 것이 바람직할 것이다.

본 연구에서는 EA를 효과적으로 지원하기 위해서 IEEE-1471[7]에서 제안된 개념 모델에 따라서 단일한 아키텍처 기술에서 이해 당사자에 따른 다양한 관점(view-point)에 해당하는 뷰들을 자동으로 생성하는 기법을 제안하고자 한다. IEEE-1471은 요즘 널리 적용되고 있는 소프트웨어 아키텍처[1,2]를 효과적으로 적용하기 위해서 제안된 IEEE의 표준이다. 이 기법의 기반을 이루는 기술은 크게 세 가지이다. 첫째, 관점의 제공을 위해서 다이어그램의 추상화 기법[3]을 적용하였다. 추상화의 필요성을 설명하기 위해서 다음의 예를 들어 보자: EA 메타 모델에서는 비즈니스 프로세스는 활동들로 구성되며, 각 활동들은 수행 중에 다루어지는 정보들과 연관된다. 이러한 메타 모델에서 비즈니스 프로세스 설계자의 입장에서 비즈니스 프로세스 중심의 관점을 생성하면서 간결성을 위해서 비즈니스 프로세스와 이와 관련된 정보만을 포함하고자 한다면, 메타 모델을 기반으로 비즈니스 프로세스와 정보 간의 관계를 적절히 유추해야 할 것이다. 메타 모델에서는 이들 간의 직접적인 관계가 없기 때문이다

두 번째로 정보 은닉(information hiding) 원칙에 따른 구성 요소의 클러스터링을 지원함으로써, 관점을 나타내는 다이어그램의 간결성의 향상을 지원한다. 예를 들어 하나의 비즈니스 프로세스가 동일한 조건의 다수의 정보와 연관되는 경우, 이러한 정보들을 하나의 클러스터로 표현함으로써, 다이어그램 내의 요소의 수를 감소함으로써 보다 이해하기 쉽도록 다이어그램을 구성할 수 있을 것이다. 특히, EA 모델들은 많은 요소들로 구성됨으로 하나의 다이어그램 상에서 표시되는 요소들의 수를 줄이는 것은 바람직하다. 이를 위해서, 본 연구에서는 수학적인 컨셉 분석(concept analysis)에 기반한 기법[6,10]을 적용하였다.

마지막으로 개념적 모델로부터 관점을 생성하기 위한 자동 레이아웃(automatic layout) 기술을 사용한다. 메타 모델에서 추출된 정보들을 효과적으로 제공하기 위해서는 사용자가 손쉽게 이해할 수 있도록 잘 배치된 요소들로 구성된 다이어그램이 바람직하기 때문이다. 특히, 본 연구에서는 오픈소스 도구로서 널리 사용되어온 Dot[4,5]를 적용하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 본 연구에서 사용하는 EA의 메타 모델 및 모델의 예제를 설명하고 IEEE-1471에 따른 이를 토대로 이해 당사자에 대한 다양한 관점의 필요성을 설명한다. 3장에서는 추상화를 사용하여 모델의 요소 간에 관계를 유추하는 기법에 대해서 설명한 후, 4장에서는 클러스터링을 통한 모델의 단순화 과정을 설명한다. 그리고 마지막으로 5장에서 결론을 제시할 것이다.

2. 예 제

본 절에서는 EA의 예제로 사용하는 EA 메타 모델 및 이 모델에 따르는 예제 모델을 설명한다. 이로서 IEEE-1471에서 설명하는 개념적인 모델의 필요성을 설명하고, 본 연구에서 해결하고자 하는 문제를 간략하게 설명한다. 또한, 예제 모델을 다이어그램으로 구성

3. 다이어그램 추상화

본 장에서는 다이어그램 추상화의 문제를 예로 들어 설명하고 이를 해결하기 위한 기법을 제안한다. 앞서 그림 2에서 기술된 아키텍처 기술에서 이해 당사자인 비즈니스 프로세스 설계자가 비즈니스 프로세스와 정보만 포함된 관점을 생성하기를 원할 경우, 메타 모델에서는 이들 간의 직접적인 관계가 없기 때문에 적절한 관계를 유추해야 한다. 그렇지 않다면, 제시되는 EA 요소들 간에 관계가 설명되지 않기 때문에 관점을 나타내기에 적절치 않을 것이다.

이 문제를 해결하기 위해서 본 연구에서는 Alexander의 기법을 적용하고자 한다: Alexander는 [3]에서 클래스 다이어그램을 보다 높은 수준 다이어그램으로 추상화하기 위해서 저 수준의 클래스들과 이 들간의 관계를 보다 높은 수준의 관계로 재해석하는 규칙들을 제안하였다. [3]에서 설명하듯이 이러한 규칙들은 간단한 구조를 갖고 있어서 개별적인 적용을 할 경우에 그 다지 효과가 없지만, 복합적으로 적용될 경우 다이어그램을 효과적으로 추상화 한다.

본 연구에서도 아키텍처 기술을 구성하는 요소들과 이 요소간의 관계들을 토대로 새로운 관계를 유추하는 간단한 규칙들을 정의하고, 이를 적용함으로써 이 문제를 해결하고자 한다. 다음은 비즈니스 프로세스와 정보간의 관계를 유추하기 위해서 본 연구에서 정의한 규칙들이다. 이러한 규칙은 그래프 재기술(graph rewriting) 스타일로 작성되었다.

표 1 비즈니스 프로세스와 정보 간의 관계 유추 규칙

[규칙 1] 활동(a) and 비즈니스 프로세스(b) and 시작(b, a) → 구성(b, a)
[규칙 2] 활동(a1) and 활동(a2) and 다음(a1, a2) and 비즈니스 프로세스(b) and 구성(b, a1) → 구성(b, a2)
[규칙 3] 활동(a) and 정보(d) and ?관계(a, d) and 비즈니스 프로세스(b) and 구성(b, a) → ?관계(b, d)

규칙들의 간략한 의미는 다음과 같다(표 1): 규칙 1은 비즈니스 프로세스와 활동 간에 시작 관계가 있을 경우, 해당 비즈니스 프로세스와 활동 간의 구성 관계를 생성한다. 규칙 2는 활동 간의 선행 관계에서 구성 관계를 유추하기 위한 규칙으로 활동 간의 다음 관계가 있고, 선행 활동이 비즈니스 프로세스와 구성 관계가 있을 경우 다음 관계에 있는 활동과 비즈니스 프로세스 간의 구성 관계를 생성한다. 규칙 3은 생성된 구성 관계를 토대로 비즈니스 프로세스와 정보 간의 관계를 유추하기 위한 것으로 활동과 정보 간의 모든 관계(?관계)에 대해서 해당 활동과 구성 관계가 있는 비즈니스 프로세스에 대해서 동일한 관계를 생성한다. 다음의 그림 3은 이러한 유추 규칙을 적용하여 생성된 관점이다.

생성된 다이어그램을 보면, 비즈니스 프로세스와 정보 간에 다수의 관계가 정의되고 있는 것을 볼 수 있다. 이는 비즈니스 프로세스를 구성하는 여러 활동들이 정보들을 다루는 방식이 서로 다르기 때문이다. 이러한 다수의 관계는 다음과 같은 규칙(표 2)을 통해서 비즈니스 프로세스와 정보 간에 두개의 관계가 정의되었을 경우, 이를 새로운 관계로 변환함으로써 추상화할 수 있다. 예를 들어 읽기 관계와 쓰기 관계가 있을 경우, 이를 쓰기 관계로 변환한다. 이 규칙에서 보듯이 읽기 보다는 쓰기가 강한 관계이며, 쓰기 보다는 생성 및 삭제 관계가 강한 관계로 규정하고 있다. 또한 생성과 삭제 간에는 순서를 부여하지 않고, 새로운 관계 생성/삭제를 정의함으로써 추상화 하고 있다. 그림 4는 관계 추상화 규칙을 적용하여 생성된 다이어그램을 보여준다.

표 2 비즈니스 프로세스와 정보 간의 관계 추상화 규칙

	생성	읽기	쓰기	삭제	생성/삭제
생성	생성	생성	생성	생성/삭제	생성/삭제
읽기	생성	읽기	쓰기	삭제	생성/삭제
쓰기	생성	쓰기	쓰기	삭제	생성/삭제
삭제	생성/삭제	삭제	삭제	삭제	생성/삭제
생성/삭제	생성/삭제	생성/삭제	생성/삭제	생성/삭제	생성/삭제



그림 3 비즈니스 프로세스 설계자 관점을 지원하는 추상화된 다이어그램(1차)

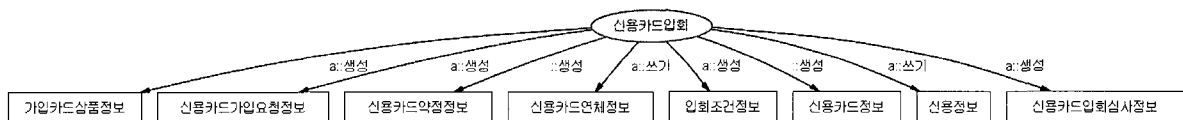


그림 4 비즈니스 프로세스 설계자 관점을 지원하는 추상화된 다이어그램(2차)

4. 클러스터링 기반 추상화

본 장에서는 클러스터링 기법을 이용하여 관점을 구성하는 요소들을 정보 은닉의 원칙에 따라서 추상화하는 기법을 제안한다. 앞서 3장에서 개발된 추상화 기법은 추상화 대상을 새로운 관계로 변환하는 반면에 클러스터 기법은 추상화 대상들을 새로운 요소로 변환한다. 이를 설명하기 위해서 앞서 생성된 그림 4를 보자. 이 다이어그램은 하나의 비즈니스 프로세스에 총 여덟 개의 정보가 관계된 것으로 생성되었다. 이러한 수는 실제보다 간략화된 것으로 실제 EA 모델에서는 보다 많은 정보들이 해당 비즈니스 프로세스에 관계됨으로써 생성된 관점의 간결성을 저해할 것이다.

이러한 문제를 해결하기 위해서 수집된 정보들과 같이 수가 많은 요소들을 특정한 기준에 의해서 클러스터링하는 것이 바람직하다. 본 연구에서는 이를 위해서 모달-스타일(modal-style) 연산자를 사용하여 구성되는 컨셉(concept) 기반의 클러스터링을 사용할 것이다(표 3). 이 기법들은 모델(6)을 구성하는 관계 정보를 사용하여, 여러 요소들 간의 공통점을 분석하여 이를 나타내는 여러 컨셉을 제안하는 데이터 분석 기법들에서 파생되었다. 우선, 이러한 컨셉 기반 클러스터링의 주요 연산자들을 설명한 후 이들을 적용하여 추상화된 모델을 설명할 것이다.

모델을 구성하는 요소들의 집합 U와 요소들이 가지는 속성들의 집합 V, 그리고, 이들 간의 관계를 나타내는 집합 $R \subseteq U \times V$ 을 가지고 다음과 같은 모달 스타일 연산자들을 정의할 수 있다.

이러한 연산자들을 토대로 컨셉 간의 래티스(lattice)를 구성하는 다양한 연구들이 있다. 이러한 연구들 중에서 본 연구에서는 정보 은닉을 효율적으로 반영하는 객체지향 컨셉(object-oriented concept)(10)을 적용

하여 클러스터링을 수행할 것이다. 객체지향 컨셉은 임의의 요소 집합 X에 대해서 $\langle R^{-1} \rangle [R](X)$, $[R](X)$ 로 정의되며 모달 연산자의 특성에 따라서 $[R](X)$ 에 해당하는 속성을 갖는 모든 요소는 $\langle R^{-1} \rangle [R](X)$ 에 포함된다. 따라서, $\langle R^{-1} \rangle [R](X)$ 으로 요소의 집합을 정의할 경우, 이를 제외한 요소들은 속성 $[R](X)$ 를 갖지 않음으로 이 속성들을 은닉할 수 있다.

표 3 모달 연산자

Sufficiency 연산자:	$[[R]](X) = \{b \in V : (\forall a \in U)[a \in X \rightarrow aRb]\}$
Possibility 연산자:	$\langle R \rangle (X) = \{b \in V : (\exists a \in X)aRb\}$
Necessity 연산자:	$[R](X) = \{b \in V : (\forall a \in U)[aRb \rightarrow a \in X]\}$
Dual Sufficiency 연산자:	$\langle\langle R \rangle\rangle(X) = \{b \in V : (\exists a \in X)[a \notin X \wedge \neg aRb]\}$

표 4 어플리케이션과 정보 관계에서 식별된 객체지향 컨셉

객체지향 컨셉
{(입회조건정보, 가입카드품질정보), (신용카드심사시스템)}
{(신용카드연체정보), (카드사시스템)}
{(신용카드입회심사정보, 신용카드약정정보, 신용카드정보, 신용카드 가입요청정보), (신용카드발급시스템)}
{(신용정보), (전국은행연합회시스템)}

예를 들어 앞서 생성된 비즈니스 프로세스 설계자의 관점에서 정보의 수를 줄이기 위해서 어플리케이션을 활용하여 객체지향 컨셉을 구성하기로 했다라고 가정하자. 다음의 그림 5는 어플리케이션이 추가된 관점을 보여준다.

그림 5를 보면, 어플리케이션과 정보 간의 관계가

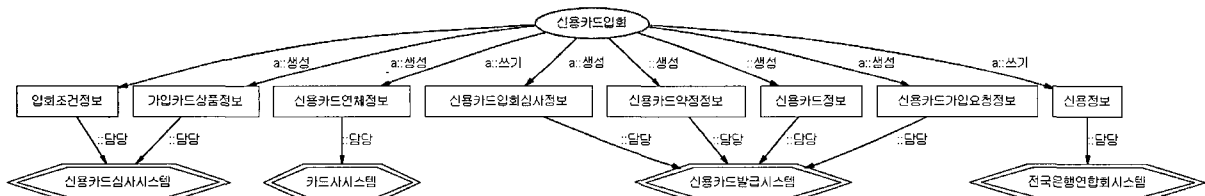


그림 5 어플리케이션이 추가된 비즈니스 프로세스 설계자 관점

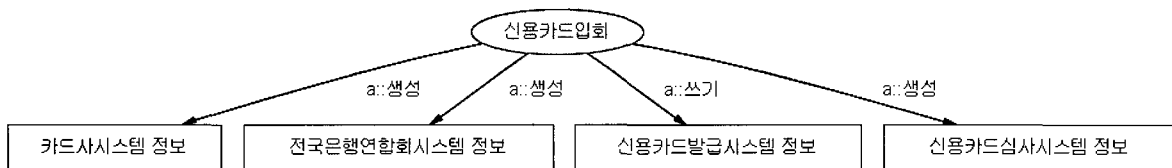


그림 6 객체지향 컨셉으로 클러스터링 된 비즈니스 프로세스 설계자 관점

이상적으로 배분되어 있어 정보를 요소로 어플리케이션을 속성으로 정의함으로써 적절한 객체지향 컨셉을 구성할 수 있음을 알 수 있다. 다음의 표 4는 이러한 설정 상에서 구성되는 가장 작은 객체지향 컨셉들을 열거한다. 식별된 컨셉들로 개별적인 정보들을 교체함으로써 그림 6과 같은 관점을 생성할 수 있다.

이처럼, 클러스터링을 통해서 관점에서 표현되어야 할 요소의 수를 줄임으로써, 생성되는 관점의 간결성을 향상할 수 있다.

5. 결 론

본 연구에서는 EA 모델이 가질 수 있는 규모의 문제를 해결하기 위해서 IEEE-1471에 기반한 이해 당사자의 관점들을 자동으로 생성하는 기법을 제안하였다. 아키텍처 기술을 토대로 다양한 관점을 자동으로 생성하기 위해서 크게 다음과 같은 기법들을 적용하였다. 첫번째 포함되지 않은 요소들로부터 유용한 관계 정보를 유추하기 위한 다이어그램 추상화 기법을 개발하였고, 두번째로 관점에 나타내는 요소의 수를 줄이기 위해서 객체지향 컨셉에 기반한 클러스터링을 지원하였다. 마지막으로 유추된 정보들이 이해당사자에게 적절한 형태로 제시될 수 있도록 자동 레이아웃 도구를 활용하여 다이어그램으로 생성하였다.

본 연구가 효과적으로 사용될 수 있기 위해서는 제안된 기법을 적절히 제공하는 환경의 구축과 함께, 생성된 관점에서 수정이 발생하였을 경우 이를 전반적인 아키텍처 기술에 효과적으로 반영하는 기법이 연구되어야 할 것이다.

참고문헌

- [1] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Addison-Wesley, 1997.
- [2] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Standford, *Documenting Software Architecture: Views and Beyond*, Addison-Wesley, 2003.
- [3] Alexander Egyed, "Automated Abstraction of Class Diagrams," *ACM TOSEM*, Vol.11, No.4, Oct., 2002
- [4] E. R. Gansner, E. Koutsofios, S. C. North, and K. P. Vo. "A Technique for Drawing Directed Graphs," *IEEE-TSE* Vol.19, No. Mar., 1993.

- [5] E.R. Gansner and S.C. North. An open graph visualization system and its applications to software engineering. *Software-Practice and Experience*. Vol.30, No.11, 2000.
- [6] G. Gediga and I. Duntsch, "Modal-Style Operators in Qualitative Data Analysis," *Technical Report # CS-02-15*, May, 2002.
- [7] IEEE-SA Standard Board, *IEEE Std-1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, Sep., 2000.
- [8] M. Rosen, "Enterprise Architecture Rollout and Training," *Enterprise Architecture Advisory Service Executive Report*, Vol.8. No.7, Cutter Consortium, 2005.
- [9] S. Spewak, *Enterprise Architecture Planning*, Boston:QED Publishing Group, 1993.
- [10] Y.Y. Yao, "Concept Lattices in Rough Set Theory," in *Fuzzy Information*, 2004. Processing NAFIPS '04, June 2004.
- [11] J. A. Zachman, "A Framework for Information Systems Architecture," *IBM System Journal* Vol.26, No.3, 1987.

김 형 호



1996 서강대학교 전산학(학사)
Summa Cum Laude
1998 한국과학기술원 전산학 석사
(소프트웨어공학)
2000 한국과학기술원 박사과정 수료
(소프트웨어공학)
2000~현재 (주)솔루션링크, 수석 컨설턴트
E-mail: hhkim@sol-link.com

민 상 윤



1993 San Francisco State University, 전산학(학사)
Cum Laude
1997 한국과학기술원 전산학 석사
(소프트웨어공학)
2002 한국과학기술원 전산학 박사
(소프트웨어공학)
2000~현재 (주)솔루션링크 대표이사,
수석 컨설턴트
E-mail : sang@sol-link.com
