

# 정보구조모델링 분석의 일관성 검증 규칙

고병선<sup>0</sup> 한미정 이서정\* 박재년  
숙명여자대학교 컴퓨터학과, 동덕여자대학교 교양교직학부\*  
{ kobs, mjhan, jnpark }@cs.sookmyung.ac.kr, sjlee@dongduk.ac.kr

## Techniques for Checking the Consistency in Analysis phase of Information Structure Modeling

Byung-Sun Ko<sup>0</sup> Mi-jung Han Seo-Jeong Lee\* Jae-Nyun Park  
Dept. of Computer Science, Sookmyung Women's University  
\*Dept. of General Education, Dongduk Women's University

### 요 약

시스템 개발 단계 중 분석 단계는 이후의 설계, 구현 및 유지 보수 단계를 지원하기 위한 가장 중요한 단계로, 이후 단계에 미치는 영향이 크다. 그러므로, 분석단계의 산출물이 정확한지, 산출물간의 일관성을 검증하는 작업이 필요하다. 이에, 본 논문에서는 비즈니스 업무 개발에 초점을 두어, 사용자의 참여가 이루어지는 단순한 표기법을 사용하는 정보구조 모델링의 분석 단계의 여러 구조도들간의 일관성 검증 방법을 제시한다. 이로써 분석 단계 중에 범하기 쉬운 오류들을 최소화하여, 정확한 분석 모델을 통한 사용자 만족도가 높은 시스템을 만들 수 있다.

### 1. 서 론

사용자의 만족도가 높은 소프트웨어를 개발하기 위해서는 시스템 개발 초기에 사용자의 요구사항을 정확히 분석하여 분석 모델을 만들어야 한다. 분석 단계의 주요 작업은 시스템 내의 정보, 사용자의 요구, 그 처리 단계 및 현재 사용하고 있는 시스템의 상태와 관리 사항 등을 분석해 내는 것이다. 시스템 개발 단계 중 분석 단계는 이후의 설계, 구현 및 유지 보수 단계를 지원하기 위한 가장 중요한 단계로, 이후 단계에 미치는 영향이 크다[4,5]. 그러므로, 분석단계의 산출물이 정확한지, 산출물간의 일관성을 검증하는 작업은 필요하다[1,2].

이에, 본 논문에서는 비즈니스 업무 개발에 초점을 두어, 사용자의 참여가 이루어지는 단순한 표기법을 사용하는 정보구조 모델링의 분석 단계의 여러 구조도들간의 일관성 검증 방법을 제시한다. 이로써 분석 단계 중에 범하기 쉬운 오류들을 최소화하여, 정확한 분석 모델을 통한 사용자 만족도가 높은 시스템을 만들 수 있다.

### 2. 정보구조모델링

#### 2.1. 기본 개념

비즈니스 업무는 사용자와 시스템으로 구성되며, 사용자는 시스템에 서비스를 요구하고, 시스템은 시스템 내의 정보를

이용해 서비스나 제품을 통하여 원하는 응답을 제공하는 기능을 한다.

사건과 그에 대한 응답을 처리하기 위해 필요한 데이터를 중심으로 개체를 파악하여, 사건과 개체간의 관계를 나타내는 것이 정보구조모델링의 주요 작업이다. 정보구조모델링은 개발자 중심이 아닌 사용자 중심의 단순한 표기법을 사용하는, 사건 중심 모델링 방법(Event-driven Modeling Approach)으로, 분석 단계에서 나타나는 산출물은 배경도, 이벤트 구조도, 이벤트 명세서, 정보 구조도, 개체 구조도, 행위 구조도 이다[3].

#### 2.2. 분석 단계

##### (1) 배경도 (Context Diagram, CD)

시스템에 직접 접근하여 요구(request)를 할 수 있는 사용자를 관련자(actor)로 하여, 시스템에 대한 요구 및 응답을 나타내어 시스템의 범위를 파악한다.

##### (2) 이벤트 구조도 (Event Diagram, ED)

관련자가 시스템에 요구하는 것을 이벤트(event)라 하고, 이를 서술식으로 WOD(Wanier-Orr Diagram) 형식으로 나타내어, 관련자와 시스템 사이의 기능 및 관계를 나타낸다.

##### (3) 이벤트 명세서 (Event Specification, ES)

이벤트 명세서는 이벤트 구조도에 나타난 각 이벤트를 처리

하기 위한 세부 행위(behavior)를 상세하게 기술하는 것이다. 행위는 이벤트를 처리하기 위한 선행조건이나, 이벤트를 수행한 후의 조치, 또는 단순히 이벤트의 일부분이 될 수 있으며, 상세히 순서적 흐름을 따라 기술한다.

**(4) 정보 구조도 (Information Structure Diagram, ISD)**

관련자의 요구인 이벤트를 해결하기 위해 필요한 데이터를 중심으로 개체를 파악하여, 이벤트와 대응되는 형태로 정보 구조도를 그린다. 여기에는 관련자, 이벤트, 이벤트를 해결하기 위해 시스템이 갖추어야 할 모든 정보가 포함되므로, 정보 구조도를 통해 사용자는 시스템에 어떤 요구를 할 수 있고, 그 요구를 만족시키기 위해 어떤 자료들이 준비되어 있어야 하는가를 알 수 있다.

즉, 정보구조도는 이벤트 구조도에 나타난 이벤트를 수행하기 위해 필요한 개체(entity)와 개체 뷰(view)를 나타내며, 이는 엔티티 객체와 인터페이스 객체로 도출될 수 있다.

**(5) 개체 구조도 (Entity Structure Diagram, ESD)**

같은 개체를 참조하더라도 이벤트가 다르면 그 뷰가 다를 수 있다. 개체 구조도는 정보 구조도의 이벤트와 개체들간의 관계를 분석하여, 사용자의 요구를 처리하기 위해 시스템이 저장하고 있어야 하는 엔티티 객체와 컨트롤 객체가 실제 데이터베이스 스키마로 변환되기 위한 개체의 논리적 내부 구조를 보여준다.

**(6) 행위 구조도 (Behavior Structure Diagram, BSD)**

행위 구조도는 관련자의 요구인 이벤트에 포함되는 세부 행위를 파악하여 명사형으로 도출하여, 정보 구조도와 같은 형태로 그린다. 이벤트 구조도, 정보 구조도, 행위 구조도에 나타나는 이벤트는 추상적인 형태이고, 이에 포함되는 세부 행위들은 이벤트 명세서에 기술되거나, 행위 구조도의 사각형 내부에 나타낸다.

분석의 최종 결과로 클래스 다이어그램을 그릴 수 있다. 이때, 클래스의 속성(attribute) 부분은 정보 구조도를 통해, 책임(responsibility) 부분은 행위 구조도를 통해 도출 가능하다. 정보 구조도와 행위 구조도는 클래스의 정적 정보인 속성과 동적 정보인 책임 추출을 위해 서로 협력적 관계에 있으며, 정보구조도모델링에서 중요한 모델 중 하나이다.

**2.3. 도출 개체 (Derived Entity)**

정보구조도모델링 분석의 결과로 도출되는 개체에는 세가지 종류가 있다. 첫째 개체는 시스템이 사용자의 요구를 받아들이고 응답을 하는데 필요한 정보를 가진 개체로, 이후 설계

및 구현 단계를 거쳐 사용자와 시스템 사이의 매개체 역할을 하는 화면이 된다. 둘째 개체는 시스템이 장기간 또는 영구히 보존할 정보로서 데이터베이스에 저장되는 정보이다. 이에는, 사용자, 시스템이 사용자에게 제공하기 위한 재료, 시설, 기자재, 비즈니스 규칙, 운영 정보 등이 포함된다. 셋째 개체는 인터페이스 객체의 요구로 여러 엔티티 객체에 비즈니스 규칙을 적용하여 조합, 처리, 응답하기 위한 비즈니스 로직, 트랜잭션, 그에 대한 결과물인 제품 등이 포함된다.

정보 구조도로부터 도출되는 개체(entity)를 표현, 정보, 행위의 세가지 관점에서 분류하면, 위의 세가지 개체는 순서대로 인터페이스 객체, 엔티티 객체, 컨트롤 객체로 대응된다[6,7].

**3. 일관성 검증 규칙**

정보구조도모델링 분석 단계의 산출물인 각 구조도 간에 일관성을 검증하기 위한 규칙들을 제시한다. 구조도와 그 구성요소를 구별하기 위해 구조도 이름에는 괄호를 사용하고, 대응관계는 -기호 '-' 으로 나타낸다.

**Rule\_1. (CD) Actor : (ISD) Object**

배경도의 관련자는 정보 구조도에 개체로 나타나야 한다.

**Rule\_2. (CD) Actor : (ESD) Object**

배경도의 관련자는 개체 구조도에 개체로 나타나야 한다.

**Rule\_3. (CD) Actor : (BSD) Object**

배경도의 관련자는 행위 구조도에 개체로 나타나야 한다.

**Rule\_4. (CD) Request : (ED) Event**

배경도의 관련자의 요구는 이벤트 구조도에 이벤트로 나타나야 한다.

**Rule\_5. (ED) Event : (ISD) Event**

이벤트 구조도의 이벤트는 정보 구조도에 이벤트로 나타나야 한다.

**Rule\_6. (ED) Event : (BSD) Event**

이벤트 구조도의 이벤트는 행위 구조도에 이벤트로 나타나야 한다.

**Rule\_7. (ED) Event : (ES) Behaviors = 1:n**

이벤트 구조도의 이벤트는 이벤트 명세서의 여러 개의 세부행위로 나타나야 한다.

- 배경도는 관련자가 시스템에 일으키는 기능적인 요구를 추상적으로 표현한다. 이런 추상적 이벤트를 구성하는 여러 이벤트들은 이벤트 구조도에, 하나의 이벤트에 대한 선/후 조건 및 상세한 순서적 흐름은 이벤트 명세서에 나타낸다.

Rule\_8. (CD) Actor : (ED) = 1:1

배경도의 관련자 수만큼 이벤트 구조도가 그려져야 한다.

Rule\_9. (CD) Actor : (ISD) = 1:1

배경도의 관련자 수만큼 정보 구조도가 그려져야 한다.

Rule\_10. (CD) Actor : (BSD) = 1:1

배경도의 관련자의 수만큼 행위 구조도가 그려져야 한다.

Rule\_11. (ESD) Attribute\_Name of Object : (ISD)

Attribute\_Name of Object

개체 구조도의 엔티티 객체의 속성명은 정보 구조도내의 개체의 속성명으로 존재해야 한다.

Rule\_12. (ESD) Attribute\_Name of Object : (ES) Behaviors

개체 구조도의 컨트롤 객체의 속성명은 정보 구조도내의 개체의 속성명으로부터 도출되거나, 이벤트 명세서에 나타나야 한다.

4. 사례 연구

(정보구조모델링 분석 산출물 및 일관성 검증 규칙 적용)

다음은 정보구조 모델링을 적용하여 수강신청 시스템을 분석한 산출물의 예이다. 수강신청 시스템에서 교수는 개설 과목, 강의 계획서를 입력하고 강의를 할 수 있고, 학생은 학교 내부 규정에 의해 최대 수강 학점을 제한 받아 수강 신청을 하여 강의를 들을 수 있으며, 직원은 강의실 배정, 강의 폐강, 수강부, 출석부 관리 등의 업무를 처리한다. 그림 1 은 수강신청 시스템의 배경도이다. 관련자 학생에 대한 이벤트 구조도

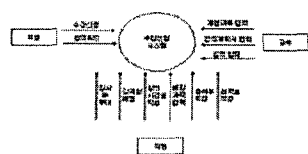


그림 1. 배경도

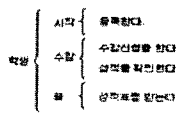


그림 2. 이벤트 구조도

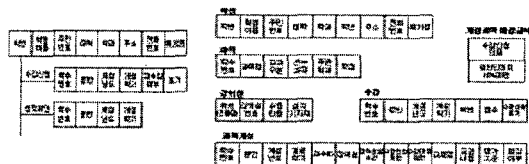


그림 3. 정보 구조도

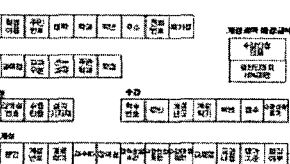


그림 4. 개체 구조도 (엔티티 객체들)

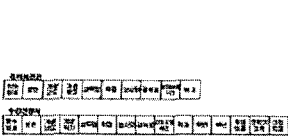


그림 5. 개체 구조도 (컨트롤 객체들)

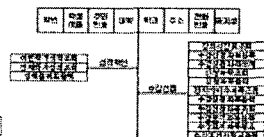


그림 6. 행위 구조도

는 그림 2 이고, 정보 구조도는 그림 3 이고, 행위 구조도는 그림 6 이며, 시스템의 엔티티 객체와 컨트롤 객체의 개체 구조도는 그림 4 와 5 이다.

5. 결론 및 향후 연구과제

본 논문에서는 단순한 표기법을 사용하는 비즈니스 업무 개발을 위한 정보구조모델링의 여러 분석 모델들 간의 일관성 검증을 위한 규칙을 제시했다. 이로써 분석 단계 중에 범하기 쉬운 오류들을 최소화하여, 정확한 분석 모델을 통해 신뢰있는 시스템을 개발할 수 있다.

앞으로, 좀더 다양한 규칙이 제시되어 정보구조모델링을 이용한 분석 모델을 기능적, 정적, 동적 측면에서 일치성 뿐만 아니라 정확성까지 검증하기 위한 연구가 있어야 하고, 좀더 정확한 의미를 위해 검증 규칙들을 정형 명세 기법으로 명세하는 것이 필요하다.

6. 참고 문헌

[1] 김도형, 정기원, "객체지향 분석단계에서 오류와 일관성 검증 방법", 정보과학회논문지, Vol.26, No.3, pp.380-392, 1999.  
 [2] 박진호, 김수동, 류성렬, "UML 다이어그램들 간의 일관성 검증 방법", 정보과학회 춘계학술발표대회, pp.524-526, 1998.  
 [3] 이서정, 사용자 중심의 일관된 시각을 지원하는 객체지향 시스템 분석 및 설계, 숙명여자대학교 박사학위 논문, Feb., 1998.  
 [4] I. Sommerville, Software Engineering, 5th Ed., Addison-Wesley, New York, 1995.  
 [5] Roger S. Pressman, Software Engineering, 4th Ed., McGraw-Hill, 1997.  
 [6] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addison-Wesley, 1999.  
 [7] Ivar Jacobson, The Object Advantage : Business Process Reengineering with Object Technology, Addison-Wesley, 1994.