

UML 핵심 다이어그램들 간의 일관성 검증을 위한 체계적 기법

허진선

동원대학교 컴퓨터소프트웨어학과
e-mail:annie8@tw.ac.kr

Systematic Techniques for Validating Consistency among Popular UML Diagrams

Jin Sun Hur

Dept. of Computer Software, Tongwon University

요약

본 논문에서는 4개의 UML(Unified Modeling Language) 핵심 다이어그램 간의 일관성 검증을 위한 체계적이고 실용적인 기법을 정의한다. UML 다이어그램 간의 일관성 검증은 프로젝트를 성공으로 이끄는 데 있어 중요한 요소이며 특히 지속가능한 사회 구현을 위해 기술의 융복합이 이루어지는 영역에서는 더욱 그러하다. 따라서 먼저 4개 핵심 다이어그램에 대한 메타모델을 정의한 후, 이들 간의 일관성 검증을 위한 지침을 상세한 수준에서 실용적인 적용에 초점을 맞추어 정의한다. 또한 제시된 검증 기법을 실제 적용한 결과를 보여주고, 검증 기법의 활용 방안에 대해서 제시한다.

1. 서론

국제 표준 UML에서 가장 폭넓게 사용되는 다이어그램은 유스케이스 다이어그램, 클래스 다이어그램, 시퀀스 다이어그램, 활동 다이어그램이다. 설계하는 담당자가 이들 다이어그램을 설계하고 나서 체계적인 검증 과정을 거치지 않으면 결국 설계 품질은 현저히 떨어지게 된다[1]. 따라서 이들 다이어그램 간의 일관성을 검증하는 문제는 중요하며 지속가능한 사회 구현을 위한 기술의 융복합이 이루어지는 영역은 에러 프론트하기에 더욱 엄격한 검증이 요구된다. 설계 모델 간의 일관성 검증 기법에 관한 연구는 이미 많이 이루어졌지만 이들에 다이어그램 사이의 일관성을 다루는 실용적이면서 체계적인 연구는 부족한 상태이다. 이에 본 논문에서는 네 다이어그램을 포함한 핵심 산출물의 요소를 준정형적으로 정의하고 이들 요소간의 일관성 검증을 위한 실용적인 지침 및 활용 방안을 제안한다.

2. 관련연구

Sulaiman의 연구[2]에서는 활동 다이어그램과 클래스 다이어그램 사이의 일관성 검증 규칙을 논리적인 술어를 이용하

여 정의하고 있다. 그러나 이 연구에서 다른 모델은 고려하지 않고 있으며, 논리적인 술어를 이용한 검증에 치우쳐 실용성이 다소 떨어지는 한계점을 가지고 있다.

Torre의 연구[3]에서는 체계적인 매핑 연구에 의해 UML 일관성 규칙을 조사하고 분석하여 통합된 119개의 규칙을 정리하고 있다. 이 연구는 UML 기반 소프트웨어 개발 활동이나 관련 교육 및 미래 연구를 위해 참조하기에 훌륭한 연구 결과이지만, 활동 다이어그램에 대한 고려가 부족하며 유스케이스 명세서가 객체지향 분석 설계의 주요 산출물임에도 이에 대한 명시적인 고려가 배제되어 있다.

Ibrahim의 연구[4]는 유스케이스 다이어그램과 활동 다이어그램 사이의 일관성 규칙을 제안하고 있다. 각 다이어그램을 정형적으로 정의하고 이들 사이의 일관성 규칙을 논리적인 접근 방식을 이용하여 정의하고 있다. 그러나 이 연구는 유스케이스 다이어그램과 활동 다이어그램 사이의 관계에만 한정되어 있고 실용적인 측면에 대한 향상이 필요하다.

3. UML 핵심 다이어그램의 메타모델

3장에서는 대표적인 4개의 UML 다이어그램을 포함한 핵심적인 객체지향 설계 모델에 대해 정의한다. 본 장에서 명세한 구성요소를 기반으로 4장에서 모델간의 일관성 검증 지침을 제안한다.

3.1 유스케이스 모델의 메타모델

유스케이스 모델은 유스케이스 다이어그램과 유스케이스 명세서로 구성된다. 유스케이스 다이어그램을 구성하는 주요 요소를 6가지로 정의하고, 각 구성요소를 위한 용어를 다음과 같이 정의한다.

actor: 액터
uc: 유스케이스
ucRel_generalization: 액터 또는 유스케이스 사이의 일반화 관계
ucRel_association: 액터와 유스케이스 사이의 연관 관계
ucRel_extend: 유스케이스 사이의 확장 관계
ucRel_include: 유스케이스 사이의 포함 관계

[그림 1] 유스케이스 다이어그램의 메타모델

유스케이스 명세서를 구성하는 요소를 9가지로 정의하고, 각 구성요소를 위한 용어를 다음과 같이 정의한다.

ucd_uc: 유스케이스 명세서가 명세하는 대상 유스케이스
ucd_actor: 해당 유스케이스를 사용하는 액터
ucd_desc: 유스케이스에 대한 설명
ucd_pre: 유스케이스가 실행되기 이전에 무조건 참이어야 하는 사전 조건
ucd_post: 유스케이스가 성공적으로 실행된 후에 참이 되는 사후 조건
flow_main: 이벤트흐름 중 정상흐름
flow_alt: 이벤트흐름 중 선택흐름
flow_excep: 이벤트흐름 중 예외흐름
scenario: 유스케이스의 실제적인 한 예

[그림 2] 유스케이스 명세서의 메타모델

3.2 클래스 다이어그램의 메타모델

클래스 다이어그램을 구성하는 주요 요소를 10가지로 정의하고, 각 구성요소를 위한 용어를 다음과 같이 정의한다.

cd_class: 클래스
class_category: model, view, controller와 같은 클래스의 스테레오타입
cd_attr: 클래스의 속성
cd_op: 클래스의 메소드
cdRel_dependency: 클래스간의 의존관계
cdRel_association: 클래스간의 연관관계
cdRel_aggregation: 클래스간의 복합관계
cdRel_composition: 클래스간의 집합관계
cdRel_generalization: 클래스간의 일반화관계
cd_multiplicity: 연관관계의 다중성

[그림 3] 클래스 다이어그램의 메타모델

3.3 시퀀스 다이어그램의 메타모델

시퀀스 다이어그램을 구성하는 주요 요소를 5가지로 정의하고, 각 구성요소를 위한 용어를 다음과 같이 정의한다.

sd_actor: 해당 시퀀스를 initiate하는 액터
sd_object: 시퀀스 다이어그램에 참여하는 객체
sd_msg: 객체 간에 호출하는 메시지
 동기 메시지, 비동기 메시지, 자체 메시지를 포함
sd_returnMsg: 회신 메시지
sd_guard: 가드 조건

[그림 4] 시퀀스 다이어그램의 메타모델

3.4 활동 다이어그램의 메타모델

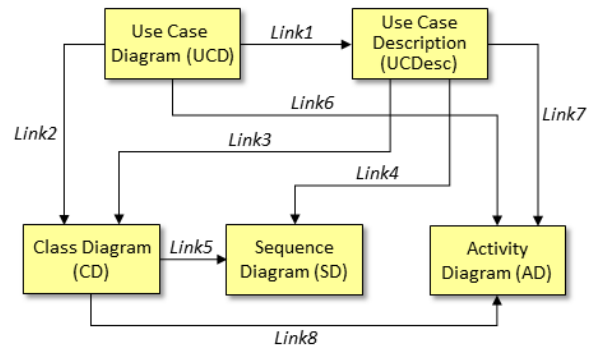
본 섹션에서의 활동 다이어그램의 설계 범위는 대상 시스템 전체에 대한 워크플로우이다. 활동 다이어그램을 구성하는 주요 요소를 10가지로 정의하고, 각 구성요소를 위한 용어를 다음과 같이 정의한다.

ad_action: 더 이상 분해될 수 없는 액션
ad_activity: 몇 개의 액션으로 분해될 수 있는 활동
ad_ctrlFlow: 제어 흐름
ad_initial: 시작 노드
ad_final: 종료 노드
ad_decision: 분기 노드
ad_merge: 병합 노드
ad_fork: 포크 노드
ad_join: 조인 노드
ad_swimlane: 스윘레인

[그림 5] 활동 다이어그램의 메타모델

4. 다이어그램 간 일관성 검증 지침

4장에서 3장에서 정의한 설계 모델들 간의 일관성을 검증하기 위한 실용적인 체크리스트와 적용 과정 및 검증 결과를 제시한다. 유스케이스 모델의 두 가지 모델을 포함한 5가지 설계 모델 간에 일관성 검증이 필요한 링크는 그림 6과 같다. 단, 본 논문에서 고려하는 활동 다이어그램의 설계 범위는 전체 시스템에 대한 워크플로우이기에 SD에서 AD로의 링크는 생략한다. 지면 관계상 두 개 링크에 대한 검증 지침 및 적용 사례를 보여준다.



[그림 6] 다이어그램 간 일관성 검증 링크

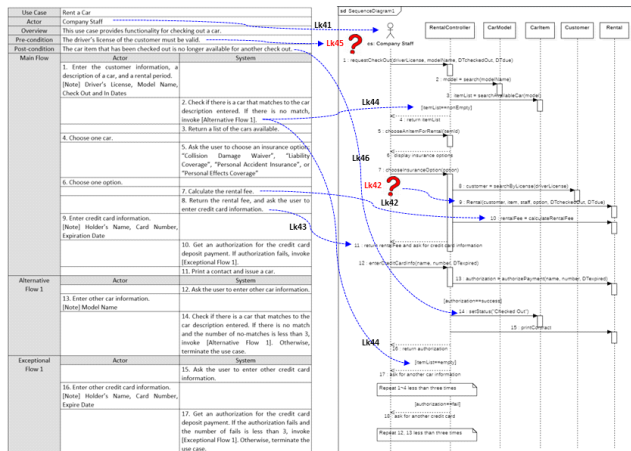
4.1 Link4: UCDesc → SD

본 절에서는 유스케이스 명세서와 시퀀스 다이어그램의 구성 요소들 간의 매핑 관계와 다중성(Multiplicity)을 기반으로 일관성을 검증하기 위한 체크리스트를 정의한다.

[표 1] UCDesc와 SD 간의 일관성 검증 체크리스트

Link ID	UCDesc의 구성요소	SD의 구성요소	다중성	체크리스트
Lk41	ucd_actor	sd_actor	1:1	SD의 sd_actor가 UCDesc에서 ucd_uc를 호출하는 actor와 같은가?
Lk42	flow_main, flow_alt, flow_except	sd_msg	1:*	sd_msg가 flow_main, flow_alt, 또는 flow_except에서 호출되었는가?
Lk43	flow_main, flow_alt, flow_except	sd_returnMsg	1:0..*	sd_returnMsg가 flow_main, flow_alt, flow_except의 시스템 액션에서 호출되었는가?
Lk44	flow_main, flow_alt, flow_except	sd_guard	1:1	sd_guard가 flow_alt 또는 flow_except로 분기되는 flow_main에서 호출되었는가?
Lk45	ucd_pre	sd_guard sd_msg	1:1	ucd_pre는 sd_guard나 sd_msg로 설계되었는가?
Lk46	ucd_post	sd_msg	1:1	ucd_post는 sd_msg로 설계되었는가?

표 1에 제시된 체크리스트를 자동차 대여 시스템(Car Rental System, CRS)의 설계 모델의 일관성을 검증하는데 적용하도록 한다.



[그림 7] CRS의 UCDesc와 SD간의 검증 과정

그림 7은 표 1의 체크리스트를 이용하여 CRS의 ‘자동차 대여’ 유스케이스 명세서와 시퀀스 다이어그램 간의 일관성을 검증하는 과정을 보여주고 있다.

[표 2] CRS의 UCDesc와 SD간의 검증 결과

UCDesc	검증 결과
Register Membership	● Lk41 - Lk46의 모든 체크리스트를 만족함
Register Car Item	● Lk45가 누락됨
Rent a Car	● Lk42의 sd_msg 중에 UCDesc에 명시되지 않은 메시지가 있음 ● Lk45가 누락됨
Return Car	● Lk42의 sd_msg 중에 UCDesc에 명시되지 않은 메시지가 있음 ● Lk45가 누락됨
Retrieve Rental Info	● Lk41 - Lk44의 모든 체크리스트를 만족함 ● Lk45, Lk46은 적용대상에서 제외됨

검증 결과, Lk45의 일관성이 만족되지 못하고 있음을 발견

할 수 있었다. 즉, 고객의 운전면허증이 유효해야 한다는 사전 조건이 있음에도 이에 대한 적절한 설계는 시퀀스 다이어그램에 없는 상황이었다. 이와 같은 방식으로 CRS의 핵심 유스케이스 5개에 대해 검증한 결과는 표 2와 같다.

표 2의 검증 결과를 보면 Register Car Item, Rent a Car, Return Car에서는 사전 조건이 적절한 설계로 이어지지 않은 것을 확인할 수 있었고, Rent a Car, Return Car과 같은 시퀀스 다이어그램에는 유스케이스 명세서에 명시적으로 드러나지 않은 메시지가 있다는 것을 확인할 수 있었다. 따라서 누락된 사전 조건에 대한 설계는 시퀀스 다이어그램에 추가할 수 있었다. 그러나 시퀀스 다이어그램에서 새롭게 추가된 메시지 같은 경우, 일관성이 깨진 것이 아닌 분석 설계의 특성상 구현 단계를 향해 갈수록 불가피하게 생길 수밖에 없는 설계자의 결정사항이기에 이렇게 새롭게 추가된 메시지의 적합성에 대한 검토가 이루어졌다.

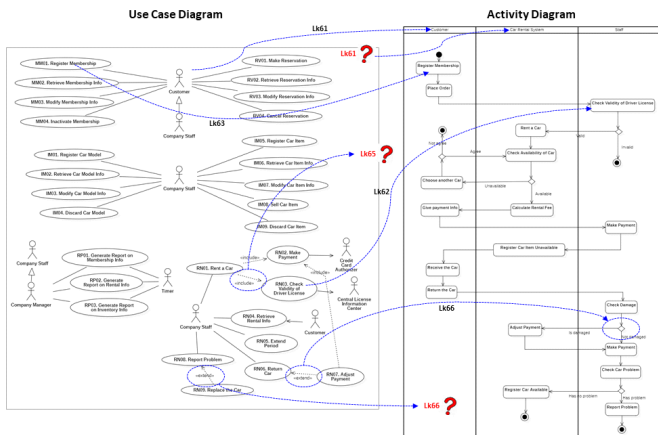
4.2 Link6: UCD → AD

본 절에서는 유스케이스 다이어그램과 활동 다이어그램 사이의 일관성 검증 체크리스트를 정의한다.

[표 3] UCD와 AD 간의 일관성 검증 체크리스트

Link ID	UCDesc의 구성요소	SD의 구성요소	다중성	체크리스트
Lk61	actor	ad_swimlane	1:0..1	ad_swimlane이 actor로부터 호출되었는가?
Lk62	uc	ad_action	1:1..*	ad_action이 어느 한 uc로부터 호출되었는가?
Lk63	uc	ad_activity	1..*: 1..*	ad_activity는 한 개 또는 그 이상의 uc로부터 호출되었는가?
Lk64	ucRel_association	ad_swimlane, ad_action ad_activity	1:1	uc와 actor 사이의 ucRel_association는 관련 ad_swimlane과 ad_action 또는 ad_activity 사이의 포함관계로 나타나고 있는가?
Lk65	ucRel_include	ad_ctrlFlow	1:1	ucRel_include는 관련 ad_ctrlFlow로 나타나고 있는가?
Lk66	ucRel_extend	ad_decision	1:1	ucRel_extend는 ad_decision으로 나타나고 있는가?

표 3에 제시된 체크리스트를 CRS의 유스케이스 다이어그램과 활동 다이어그램 사이의 일관성을 검증하는데 적용해보았다. 그림 8은 검증 과정을 보여주고 있다. 그림 8의 활동 다이어그램은 고객이 회원으로 등록하여 자동차를 대여하고 반납하는 CRS의 대표적인 워크플로우에 대해 설계하였다.



[그림 8] CRS의 UCD와 AD간의 검증 과정

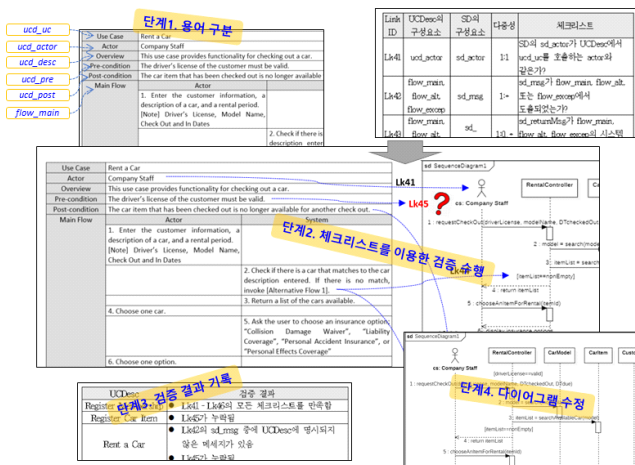
이러한 과정을 거쳐 일관성을 검증한 결과, 다음의 표 4와 같은 문제가 있음을 발견할 수 있었다. 본 검증 결과는 향후 활동 다이어그램을 검토하여 수정하는 작업의 주요 입력물로 활용되었다.

[표 4] CRS의 UCD와 AD간의 검증 결과

LinkID	검증 결과
Lk61	● Car Rental System이라는 swimlane이 애매모호함을 발견
Lk62,Lk63	● 체크리스트 만족함
Lk64	● Lk61의 검증 결과를 반영하여 설계 결과물 수정 시, 업데이트 필요함
Lk65	● RN01.Rent a Car와 RN02.Make Payment 사이의 관계는 제어흐름에 반영이 되었으나, RN01.Rent a Car와 RN03.Check Validity of Driver License 사이의 관계는 제어흐름의 순서상에 일관성이 깨짐
Lk66	● RN06.Return Car와 RN07.Adjust Payment 사이의 관계는 잘 반영이 되었으나, RN08.Report Problem과 RN09.Replace the Car 사이의 관계는 누락되었음

5. 일관성 검증 기법의 활용

본 장에서는 앞에서 정의된 일관성 검증 기법을 활용하기 위한 체계적인 절차를 그림 9와 같이 4 단계로 제시한다.



[그림 9] 일관성 검증 기법 활용 절차

먼저 검증하고자 하는 대상 설계 산출물을 3장에서 정의한 메타모델의 용어별로 구분한다. 그런 후, 4장의 표 1과 표 3과 같이 제시된 검증 체크리스트를 검증 대상 용어들에 적용하여 검증을 수행한다. 4장의 그림 7과 그림 8이 검증하는 과정을 보여준다. 이러한 검증 과정을 거치면서 얻어진 검증 결과를 표 2나 표 4와 같이 정리한다. 그리고 이 검증 결과를 이용하여 설계 결과물을 수정한다. 이 과정을 반복하여 대상 설계 다이어그램 간의 일관성을 맞추도록 한다.

6. 결론

지속가능한 사회 구현을 위해 기술의 융복합이 이루어지는 영역은 더욱 예러가 발생할 확률이 높기에 해당 설계 결과물에 대한 더욱 엄격한 검증이 요구된다. 특히 설계 산출물 간의 일관성 검증 기법에 관한 연구는 프로젝트 성공에 있어 중요한 요소이기에 그간 많은 연구가 진행되었으나 일부 부분적인 설계 산출물 간의 일관성만을 정의하거나 정형적으로 정의하는데 치중한 부분이 있다. 따라서 본 연구에서는 통합적인 관점에서 핵심 UML 다이어그램 간의 일관성을 검증하기 위한 기법을 상세한 수준에서 체계적으로 제안하였다. 특히 높은 실용성과 정확성을 위해 준정형적인 접근 방법과 체크리스트의 형태로 일관성 검증 기법을 제시한다. 제시된 기법을 실제 프로젝트에 적용함으로써 본 연구의 적용성을 검증하였으며, 나아가 본 논문에서 제시된 일관성 검증 기법을 활용하는 방안을 제시한다.

참고문헌

[1] Pierre, B. and Fairley, R. E. eds., "SWEBOK Version 3.0," IEEE, ISBN-10: 0-7695-5166-1, 2014.

[2] Sulaiman, N., Ahmad, S. S. S., and Ahmad, S., "Logical approach: Consistency rules between activity diagram and class diagram," *International Journal on Advanced Science, Engineering and Information Technology*, Vol.9, No.2, pp.552-559, 2019.

[3] Torre, D., Labiche, Y., Genero, M., and Elaasar, M., "A systematic identification of consistency rules for UML diagrams," *Journal of Systems and Software*, Vol.144, 121 - 142, 2018.

[4] Ibrahim, N., Ibrahim, R., Saringat, M. Z., Manso, D., and Herawan, T., "Consistency Rules between UML Use Case and Activity Diagrams Using Logical Approach," *International Journal of Software Engineering and Its Applications*, Vol.5, No.3, pp.119-134, 2011.