

온톨로지 기반의 소프트웨어 설계에러검출방법

서진원¹, 김영태¹, 공헌택¹, 임재현¹, 김치수^{1*}
¹공주대학교 컴퓨터공학부

A Method based on Ontology for detecting errors in the Software Design

Jin-Won Seo¹, Young-Tae Kim¹, Heon-Tag Kong¹, Jae-Hyun Lim¹
and Chi-Su Kim^{1*}

¹Dept. of Computer Engineering, Kongju National University

요 약 본 논문은 소프트웨어 설계 시 향상된 오류 검출방법을 통해서 소프트웨어 설계의 질을 향상시켜 그에 따른 소프트웨어 제품의 질을 향상시키는데 목적을 두고 있다. 본 논문에서 오류검출의 범위는 일관성결여 오류로 제한하여 일관성 오류에 관한 명세에 초점을 맞춘다. UML 표현의 문제점인 의미 일관성 표현의 한계를 극복하기 위해 ODES 모델을 제안하였으며 검증방법으로 일관성 검사 방법을 제안한다. UML 설계에서 확인된 의미적으로 중요한 특징이 ODES 모델로 구현되며, UML 모델을 ODES 모델로 변환과정에서의 일관성검사방법을 제시한다. ODES 모델로의 변환과정은 ODES 모델의 인스턴스를 생성하기 위한 알고리즘에서 복수의 사상테이블을 이용하는 소프트웨어 설계의 어휘분석과 의미분석을 포함한다.

Abstract The objective of this thesis is to improve the quality of a software product based on the enhancement of a software design quality using a better error detecting method. Also, this thesis is based on a software design method called as MOA(Methodology for Object to Agents) which uses an ontology based ODES(A Method based on Ontology for Detecting Errors in the Software Design) model as a common information model. At this thesis, a new format of error detecting method was defined. The method is implemented during a transformation process from UML model to ODES model using a ODES model, a Inter-View Inconsistency Detection technique and a combination of ontologic property of consistency framework and related rules. Transformation process to ODES model includes lexicon analysis and meaning analysis of a software design using of multiple mapping table at algorithm for the generation of ODES model instance.

Key Words : Consistency, Software design, Error detection

1. 서론

소프트웨어 개발 프로젝트의 분석결과에 따르면 가능한 한 개발초기에 오류를 발견하는 것이 오류를 수정하는데 필요한 시간, 노력, 비용 등을 절약할 수 있다. 오류를 발견하지 못하고 제품을 고객에게 인도할 경우에는 컴퓨터를 재시작하는 단순한 번거로움으로부터 사람의 생명에 영향을 미치는 심각한 사고, 고객 유실, 판매 감

소, 보수비용 증가 등의 문제를 발생시킬 수 있다[1]. 일반적으로 설계오류에는 불완전성, 일관성 결여, 중복성 등을 들 수 있다.

본 논문에서는 소프트웨어 설계에서 가장 보편적이며 가장 파악하기 어려운 오류인 일관성 결여를 검출하는 방법을 제시한다.

소프트웨어 개발에 있어서 일반적인 일관성 결여의 예로는 1개 이상의 이름으로 단일 사실, 행위 또는 제한조

본 연구는 한국과학재단 특장기초연구 (R01-2006-000- 10555-0)지원으로 수행되었음.

*교신저자 : 김치수(cskim@kongju.ac.kr)

접수일 09년 03월 18일

수정일 (1차 09년 04월 14일, 2차 09년 10월 07일)

게재확정일 09년 10월 14일

건을 참조하는 것(aliasing), 행위, 사실 또는 제한조건에 대한 표현 사이의 모순과 부정확한 표현 등이 있다[2].

일관성의 결여로 인해 중대한 설계 문제의 해석 오류 또는 다중 해석이 발생할 수도 있다. 또한 소프트웨어 개발 모델에서의 일관성 결여는 신뢰성 및 안전성 등의 시스템 지원 특성에서의 어려움, 스케줄 지연, 비용 증가, 유지관리의 어려움 등 다양한 문제점을 발생시킨다[3].

본 논문에서는 구문 일관성 결여 검출능력만 제공하는 기존 설계도구들의 문제점을 개선하여 의미 일관성 결여를 검출할 수 있는 설계 방법을 제안하고자 한다. 즉, ODES 모델의 컨스트럭트(construct)를 기반으로 하여 반복적인 뷰간 일관성 에러 검출 및 모듈화 된 프로젝트에서의 설계 에러 검출이 가능한 자동화 일관성 프레임워크를 제안하고자 한다.

2. 관련연구

2.1 UML

본 논문은 UML 2.0을 사용하여 명시된 UML 설계에 초점을 맞추고 있는데 표준 UML 2.0은 UML 아키텍처 중 특히 UML 메타-모델 계층과 모델 계층을 다룬다.

UML 2.0은 활동, 클래스, 통신, 컴포넌트, 혼합 구조, 배치, 대화, 객체, 패키지, 시퀀스, 스테이트머신, 타이밍, 유스케이스 다이어그램을 포함한다.

본 논문에서는 UML 다이어그램 중 분석과 설계에 한정된 다이어그램만을 다루기로 하며 클래스 다이어그램, 객체 다이어그램, 시퀀스 다이어그램, 유스케이스 다이어그램, 스테이트머신 다이어그램만을 포함하는 UML 집합 사이의 일관성으로 제한한다[4,5].

2.2 온톨로지

온톨로지는 공유된 개념화의 형식적이며 명확한 명세로 정의한다. 온톨로지가 컴퓨터로 이해되고 관리되기 위해서는 형식화되어야 하고, 온톨로지에 대한 여러 가지 형식적 정의가 있다[6].

본 논문에서는 "Ontology Development 101"을 사용한다[7]. 기본적인 단계는 ①온톨로지의 도메인과 범위 확인 ②존재하는 온톨로지의 재사용 평가 ③온톨로지서 사용되는 중요한 용어 확인 ④클래스와 그들의 계층 관계 확인 ⑤클래스 프로퍼티 확인 ⑥클래스 특성의 정의 ⑦클래스 인스턴스 생성이다. ODES(Ontology for Software Specification and Design) 모델의 개발은 "Ontology Development 101"의 1단계에서 6단계까지를

따르고 있다.

2.3 일관성

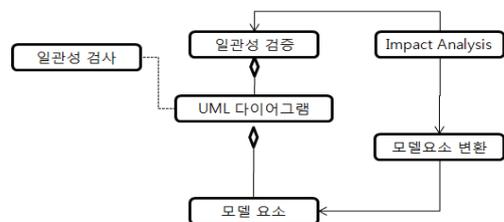
UML 설계에서 일관성 결여를 검출하는 접근법에는 PVS(Prototype Verification System), CTFL(Concurrent Transaction Frame Logic), NuSMV 등이 있다. 이러한 도구는 구문 일관성 결여와 같은 간단한 오류를 검출하는데 유용하지만 고급의 검증 결과를 지원하지 못하거나 KAOS를 지원하는 형식적 요건 공학 도구의 장애 인식 및 충돌관리 기법을 지원하지 못한다.

UML내의 정확한 시맨틱의 부족으로 인해 유발되는 문제와 부적절성을 논하는 연구가 지난 수년 동안 많이 수행되었다[5]. 수많은 이론, 연구 프로젝트 및 도구가 개발되어 UML내의 정확한 시맨틱 부족을 다루었다. 이런 대부분 접근의 주요 쟁점은 UML의 시맨틱을 형식화하는 것이다. 형식적 검증과 더불어 소프트웨어 명세의 특성은 일반적으로 수학적으로 검증된다. UML을 형식화하려는 시도는 UML의 특성으로 인해 수많은 난관에 부딪치게 되는데 이러한 특성에는 “이질적인 준 형식적 표기”, 다중적인 관점, 상투적인 문구 및 UML은 “어떤 특정한 개발절차를 규정하지 못한다.”는 사실 등이 있다[5].

최근의 여러 연구에서 UML 설계에서의 일관성 결여를 검출하는 규칙을 정의하지만 대부분은 소수의 규칙만 정의하는데 예외적으로 100개의 규칙을 정의하는 경우가 있었다[8].

2.3 기존 사례 연구

L. Briand 등은 순수한 UML 모델로 생성된 소프트웨어의 에러검출을 위한 iACMTool 연구에서 그림 1과 같이 "Impact Analysis Tool"을 정의하고 변경된 요소들에 의하여 생성된 변경모델을 적용하여 적용 전·후의 일관성 검사를 시행하여 에러를 검출하는 방법을 제시하였다[9]. 그러나 UML 모델내의 모델요소가 동일한 내용의 문제임에도 불구하고 "Impact analysis tool"을 구동하였을 경우 별도의 변수로 인지하여 Program의 1차 소스와 비교되는 2차 소스에서 일관성을 유지하지 못하는 단점이 있다.



[그림 1] Impact Analysis Tool개념도

M. Encarnación Beato 등의 UML Automatic Verification Tool (TABU:Tool for the Active Behaviour of UML) 연구에서는 소프트웨어 개발의 에러검출을 위하여 UML Tool에 의해 생성된 XMI에 TABU를 적용하여 자동으로 SMV(Symbolic Model Verifier)를 생성한다. 이 과정에서 CASE간 일관성 검사를 수행하며, S/W의 일관성 에러를 1회에 한하여 검출한다. 그러나 완전히 자동화된 검출 체계를 갖지는 못하는 단점이 있다[10].

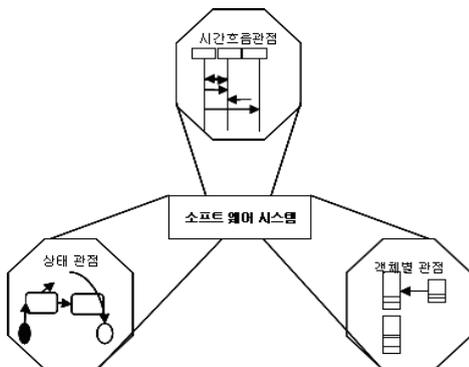
3. 소프트웨어 설계의 다중관점

3.1 문제점 및 해결방법

UML은 형식의 일관성은 잘 유지되어진다. 그러나 케이스별 독립적인 명세로 인해 의미적인 일관성 결여 문제는 남아 있다. 즉, 모든 UML내의 정확한 시맨틱의 부족으로 인해 유발되는 문제와 부적절성에 관한 연구가 많이 수행되고 있다[5].

비형식적인 소프트웨어 모델링 기법에서 가장 보편적으로 사용되는 UML은 소프트웨어 시스템 모델링의 표준으로 명세의 높은 자율성을 보장하도록 설계된 다양한 모델링 표기법의 확장 표기법들의 집합이다. 그러나 이러한 표기법은 소프트웨어 설계자가 그림 2와 같이 시스템의 부분적으로 중첩된 시각을 명시할 수 있게 한다. 따라서 이러한 유용성은 때때로 소프트웨어 설계에 일관성의 결여를 발생시킨다.

본 논문에서는 이러한 일관성의 문제를 해결하기 위해 온톨로지를 이용한 통합된 모델을 제안한다. 제안하는 온톨로지 기반 모델은 이질적인 모델을 통합하는 방법으로 모델 및 애플리케이션에 독립적인 방법을 제공한다. 여기서 온톨로지는 실제적으로 통합된 모델에 필요한 개념적인 독립성을 제공한다.

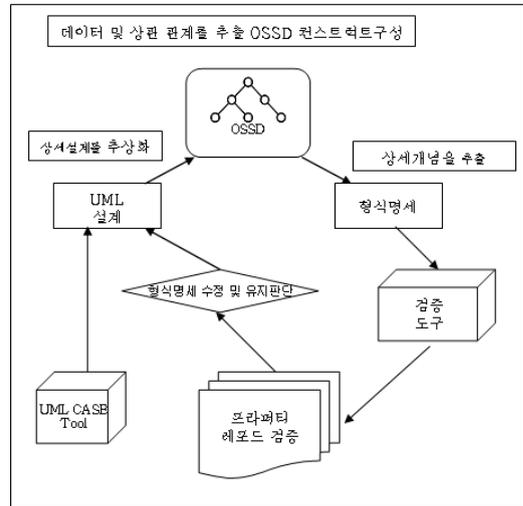


[그림 2] 소프트웨어 설계의 다중관점

3.2 ODES 모델

ODES(Ontology for Software Specification and Design)는 비형식적 소프트웨어 설계를 형식적 에이전트 지향적 요건명세로 변환하는 과정에서 일반 모델로서 사용되기 위하여 개발된 것이다. 따라서 ODES는 소프트웨어 설계의 여러 관점을 통합하고 객체 지향적 개념과 에이전트 지향적 개념을 통합한다.

그림 3에서 첫 단계는 UML을 이용하여 명시된 소프트웨어 디자인을 ODES 모델 인스턴스로 변환시킨다. 두 번째 단계는 연관된 형식적 명세로 변환되는데 이는 명세에 대한 일관성 속성을 추론하는 형식적인 검증 도구를 이용하기 위한 입력 자료로 사용된다. 세 번째는 생성된 명세를 처리하여 초기 UML 디자인 내의 일관성 결여 리스트를 생성한다.



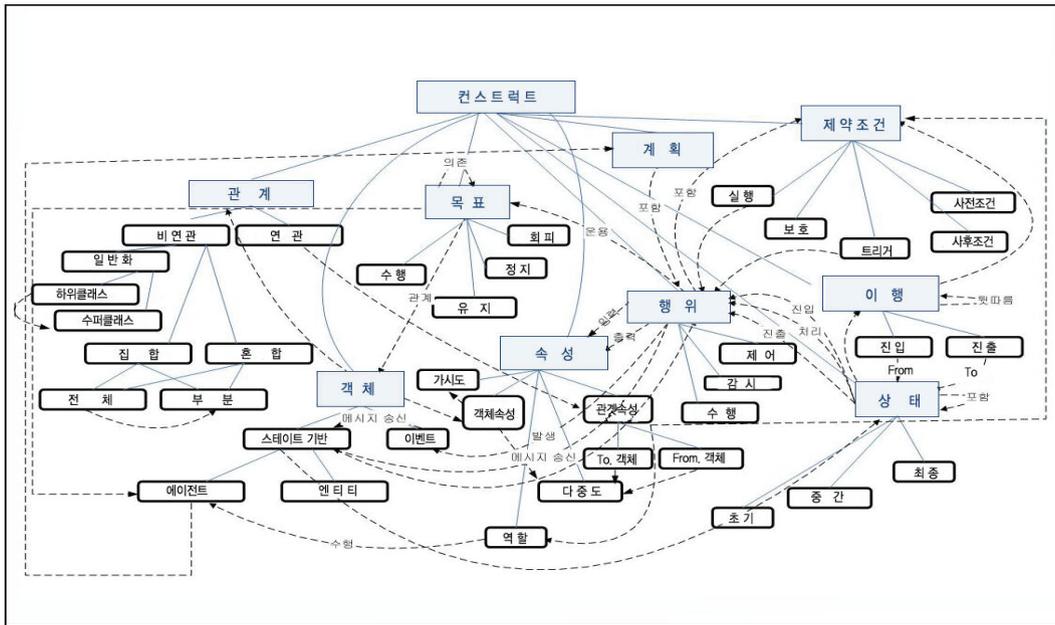
[그림 3] 통합설계방법의 전체 구성도

즉, ODES는 온톨로지 기반 모델로서 UML설계에서 구조, 데이터 및 상관관계를 추출하여 9개의 ODES 컨스트럭트로 구성된 인스턴스로의 변환 및 ODES 컨스트럭트 구조간의 행위관계를 묘사하는 프로퍼티의 표현에 온톨로지 개념을 적용한다.

그림 4에 나타난 바와 같이, ODES 모델은 자동화된 운용을 위하여 의도된 소프트웨어 개발 개념의 계층적 분해구조이다.

4. UML(다중) 관점에서의 일관성 검사

본 논문에서 오류검출의 범위는 일관성결여 오류로 제



[그림 4] ODES 계층구조

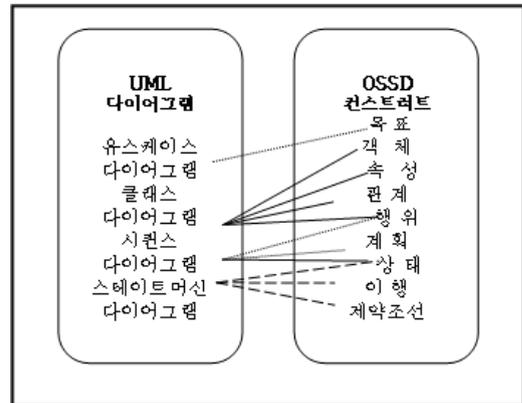
한하며 UML 다이어그램의 표현으로부터 ODES로 변환하고 일관성 검사에 의한 에러검출과정을 표현한다.

4.1 UML에서 ODES 모델로의 변환

UML 클래스 다이어그램은 ODES 모델의 객체, 속성, 관계, 행위 컨스트럭트 등과 연관된다. 시퀀스 다이어그램은 행위의 ODES 개념을 상세화하며 행위와 연관된 제약조건을 확인한다. 시퀀스 다이어그램내의 각 메시지는 UML 시그널 또는 운영호출에 해당하는지에 따라 행위를 생성한다.

또한 스테이트머신 다이어그램은 제약조건외의 ODES 개념을 상세화하며 ODES 모델내의 상태와 이행을 확인한다. 유스케이스 다이어그램은 ODES 모델내의 객체 및 행위와 연관된 목표를 확인한다. 마지막으로 분석된 정보는 UML 설계에 대한 ODES 모델의 인스턴스로 결합된다. 그림 5는 UML 다이어그램과 ODES 모델 컨스트럭트 간의 개념적 관계를 보여준다.

UML의 ODES 변환은 그림 6에서와 같이 품사, 행위, 관계, 객체, 상태, 이행, 제약조건, 분류, 목표 등으로 ODES 표현을 위한 분석시행을 한다.

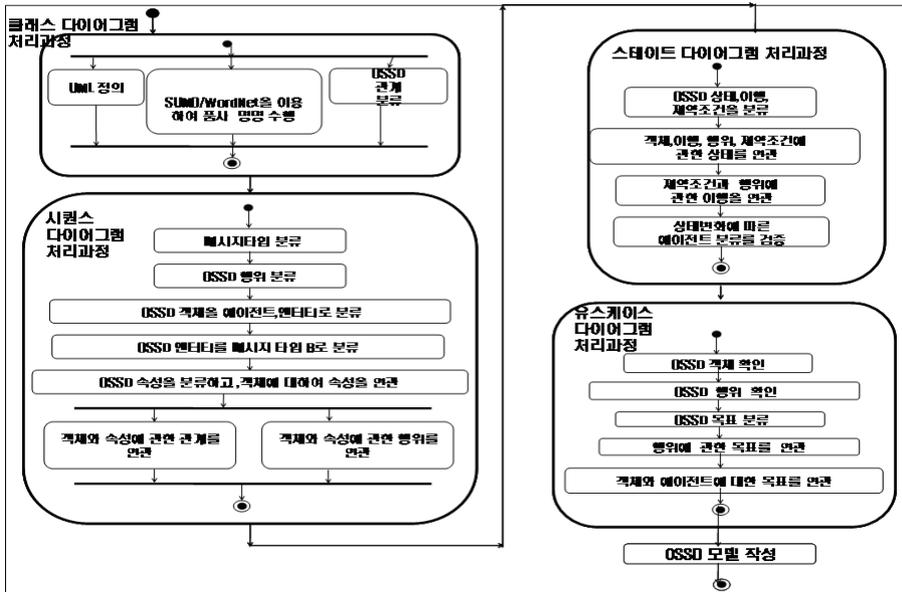


[그림 5] UML에서 ODES로의 사상에 대한 관점

4.2 일관성 검사

UML 다이어그램의 행위를 위한 부간 비일관성 검사 기법은 일관성 프레임워크에 기반하고 있다. 이 기법의 주요 목적은 설계의 부분적으로 중첩된 관점을 통하여 모델 구성요소의 정의 안에서의 일관성 결여를 확인하는 것이다.

다음은 ODES_부간일관성규칙 IC(Inter-View InConsistency)의 3개를 형식적으로 정의한다. 각 규칙의 표기와 의미는 다음과 같다.



[그림 6] UML의 ODES변환

IC규칙1:

$$\forall o1 \text{ [컨스트럭트:객체}(o1) \rightarrow (\text{클래스 다이어그램}(o1) \wedge (\exists o2[\text{시퀀스 다이어그램}(o2) \vee \text{스테이트머신 다이어그램}(o2)]) \wedge o1=o2))]$$

의미 : ODES 객체는 반드시 UML 클래스 다이어그램에 정의되어야 하며 최소하나의 UML 시퀀스 다이어그램 및 하나의 UML 스테이트머신 다이어그램에서 참조되어야 한다.

IC규칙2:

$$\forall r1 \text{ [컨스트럭트:관계}(r1) \rightarrow (\text{클래스 다이어그램}(r1) \wedge (\exists r2[\text{시퀀스 다이어그램}(r2) \wedge r1 = r2]))]$$

의미 : ODES 관계는 반드시 UML 클래스 다이어그램에 정의되어야 하며 최소 하나의 UML 시퀀스 다이어그램에 참조되어야 한다.

IC규칙3:

$$\forall b1 \text{ [컨스트럭트:행위}(b1) \rightarrow (\text{클래스 다이어그램}(b1) \wedge (\exists b2[(\text{시퀀스 다이어그램}(b2) \wedge \text{스테이트머신 다이어그램}(b2)) \wedge b1=b2]))]$$

의미 : ODES 행위는 반드시 UML 클래스 다이어그램에 정의되어야 하며 최소 하나의 UML 시퀀스 다이어그램 및 하나의 UML 스테이트머신 다이어그램에 참조되어야 한다.

본 논문은 위의 표현에 나타난 것과 같은 규칙의 집합과 표 1에 보인 것과 같은 뷰간 비일관성 탐색테이블(Inter-View Inconsistency Detection Table)의 조합을 통해서 일관성 결여를 검출한다. 뷰간 비일관성 탐색테이블에 모인 정보를 관점간의 일관성 규칙과 조합하여 일관성 결여를 확인한다. 표 1에 뷰간 비일관성 탐색테이블 입력 사항의 예를 제공하는데 참조의 편의를 위하여 ODES구성요소에 숫자 접미사가 더해졌다.

[표 1] 뷰간 비일관성 탐색테이블

UML	ODES	클래스 다이어그램	시퀀스 다이어그램	스테이트머신 다이어그램	유스케이스 다이어그램
클래스이름	에이전트3	Y	Y	Y	Y
연관이름	연관6	Y	N	N	N
오퍼레이션이름	행위5	N	Y	Y	Y

[표 2] 기존 방법과 ODES의 비교

	iACMTool	UML Design Tools	TABU	ODES
UML CASE 내 오류검출	○	○	○	○
UML 뷰간 일관성 에러 검출	×	×	○	○
모듈화 된 프로젝트에서의 설계 시 에러검출	○	×	○	○
자동 에러 검출	○	×	△	○

표 1은 에이전트3과 관련하여 에이전트3의 UML 동치에 대한 참조가 클래스, 시퀀스, 스테이트머신, 유스케이스 다이어그램에 존재함을 나타내는 {Y, Y, Y, Y}의 조합이 있음을 보여준다. 그에 따라 에이전트3은 IC규칙1을 따르게 된다.

그러나 연관6과 관련하여 연관6의 UML 동치가 클래스 다이어그램에만 존재함을 나타내는 {Y, N, N, N}의 조합이 있다. 이에 따라 연관6은 IC규칙2를 위배하게 되는데 이는 연관6이 시퀀스 다이어그램에서 참조되지 않기 때문이다.

마지막으로 행위5와 관련하여 행위5의 UML 동치가 클래스다이어그램에 존재하지 않으며 시퀀스, 스테이트머신, 유스케이스 다이어그램에 존재함을 나타내는 {N, Y, Y, Y}의 조합이 있음을 보여준다. 따라서 행위5는 IC규칙3을 위배하게 된다.

4.3 일관성 규칙 추가

UML 다이어그램의 유형에 따라 뷰간 비일관성 탐색 테이블에 추가적인 규칙을 더할 수 있다. 예를 들어 UML 클래스 다이어그램과 연관되는 UML 유스케이스 다이어그램에 각 액터를 요구하는 일관성 규칙이 더해질 수 있다. UML 클래스 다이어그램의 각 클래스가 객체로 ODES에 구현된다는 지식이 주어졌을 때, 이러한 규칙을 확인하는 것은 뷰간 비일관성 탐색테이블에서 UML 유스케이스 다이어그램 내의 각 액터와 어떤 객체사이의 1:1 대응을 보여주기 위해 다음과 같은 일관성 규칙들을 추가하여 검사를 함으로써 오류검출을 한다.

- 각 UML 클래스는 관련된 스테이트머신을 가져야 한다.
- 클래스다이어그램 내 각 UML 클래스는 스테이트머신 다이어그램 내에서 적어도 하나의 객체 생명선과 연관되어야 한다.
- 시퀀스 다이어그램 내의 UML 객체는 적어도 하나의 클래스 다이어그램에 존재해야 한다.

어떤 클래스가 UML 유스케이스와 클래스다이어그램에서 확인되며, 시퀀스 다이어그램과 스테이트머신 다이어그램에서는 확인되지 않는다면 추가규칙에 위배는 일관성 에러가 검출되며 표 1과 같이 나타낼 수 있다.

본 논문에서 제안한 ODES 에러 검출 방법을 적용하면 표 2에 보인 것과 같이 UML 뷰간 일관성 에러 검출과, 모듈화 된 프로젝트에서의 설계 시 에러검출을 자동화 프로세스에 의해 처리하므로 소프트웨어 설계 품질을 향상시킬 수 있다.

5. 결론 및 향후 연구

본 논문에서 오류검출의 범위는 일관성 결여 오류로 제한하여 일관성 오류에 관한 명세에 초점을 맞춘다. 일관성 검사방법은 UML 모델에서 ODES 모델로의 변화과정의 결과로 뷰간 일관성 테이블의 갱신을 하며, 이 테이블을 이용한 다중 관점에서의 일관성 검사를 포함한다.

본 논문에서 제안한 ODES 에러 검출 방법은 소프트웨어 설계의 여러 관점을 통합하고, UML 뷰간 일관성 검출방법을 제공함으로써 소프트웨어 설계 품질을 향상시킨다.

향후 연구 방향으로는 ODES 모델을 ODES 모델의 OWL 구현을 하는 온톨로지 모델링과 지식기반획득도구를 이용하여 구축하는 연구, ODES 모델의 일관성 규칙을 시멘틱 규칙 언어를 위한 W3C의 제안인 Semantic Web Rule Language(SWRL)을 이용하여 명시하는 연구 등이 필요할 것이다.

참고문헌

[1] S. Easterbook, "Model Management and Inconsistency in Software Design", Proc. NSF Workshop on the Science of Design, 2003.
 [2] G. Engels, R. Heckel, and J. Kuster, "Rule-based Specification of Behavioral Consistency based on the

UML Meta-Model", Proc. UML 2001.

- [3] Object Management Group, Unified Modeling Language(UML) 2.0, 2003.
- [4] 채홍석, 객체지향 CBD 개발 Bible, 한빛 미디어, 2003.
- [5] 김영태, 임재현, 김치수 "효율적인 온톨로지 개발을 위한 UML의 변경", 한국산학기술학회 제9권 2호 pp.415-421, 2009.
- [6] 서진원, 공헌택, 임재현, 김치수 "온톨로지 분석 기반의 UML클래스 모델을 이용한 데이터 통합", 한국산학기술학회 제9권 2호 pp.422-430, 2009.
- [7] J. Gennari, M. Musen, R. Fergerson, W. Grollso, M. Crubezy, H. Eriksson, N. Noy, and S. Tu, "The Evolution of Protege: An Environment for knowledge - Based Systems Development", International journal of Human-Computer Studies, vol. 58, no. 1, pp. 89-123, 2003.
- [8] E. Neuhold, C. Niederee and M. Fuchs, "Semantic Web Application Models", ER 2003 - 22nd International Conference on Conceptual Modeling, 2003.
- [9] L. Briand, Y. Labiche, and L. O'Sullivan, "Impact Analysis and Change Management of UML Models", Technical Report SCE-03-01, Carleton University, 2003; also, Proceedings of IEEE International Conference on Software Maintenance (ICSM), pp. 256-265, 2003.
- [10] M. Encarnación Beato, Manuel Barrio Solórzano, Carlos E. Cuesta, "UML Automatic Verification Tool (TABU)", Electronic Notes in Theoretical Computer Science Volume 127, Issue 4, 2005.

서진원(Jin-Won Seo)

[정회원]



- 1994년 2월 : 한밭대학교 전자계산학과 졸업(학사)
- 2004년 2월 : 공주대학교 멀티미디어공학과 졸업(석사)
- 2008년 2월 ~ 현재 : 공주대학교 컴퓨터공학과 박사과정

<관심분야>

데이터통합, UML, 온톨로지

김영태(Young-Tae Kim)

[정회원]



- 2000년 2월 : 공주대학교 전자계산학과 졸업(학사)
- 2002년 2월 : 공주대학교 전자계산학과 졸업(석사)
- 2004년 2월 : 공주대학교 컴퓨터공학과 박사과정 수료

<관심분야>

데이터통합, UML, 온톨로지

공헌택(Heon-Tag Kong)

[정회원]



- 1984년 5월 : Northeast Missouri State Univ. 전산학과(학사)
- 1987년 12월 : Utah State Univ. 전산학과(석사)
- 1998년 2월 : 단국대학교 전산통계학과(박사)
- 1988년 1월 ~ 1990년 3월 : 한국국방연구원 전산체계연구부 근무
- 1990년 4월 ~ 현재 : 공주대학교 컴퓨터공학부 교수

<관심분야>

병렬 알고리즘, 병렬처리 컴퓨터, 데이터베이스

임재현(Jae-Hyun Lim)

[정회원]



- 1986년 2월 : 중앙대학교 전자계산학과 졸업(학사)
- 1988년 8월 : 중앙대학교 대학원 전자계산학과 졸업(석사)
- 1998년 8월 : 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
- 1998년 9월 ~ 현재 : 공주대학교 컴퓨터공학부 부교수

<관심분야>

상황인식, RFID/USN, 온톨로지, 인터넷 기술

김 치 수(Chi-Su Kim)

[정회원]



- 1984년 2월 : 중앙대학교 전자계산학과 졸업(학사)
- 1986년 8월 : 중앙대학교 대학원 전자계산학과 졸업(석사)
- 1990년 8월 : 중앙대학교 대학원 전자계산학과 졸업(박사)
- 1990년 9월 ~ 1992년 8월 : 공주교육대학교 전임강사
- 1992년 9월 ~ 현재 : 공주대학교 컴퓨터공학부 교수

<관심분야>

소프트웨어 개발 방법론, 온톨로지