

프레임을 이용한 다이어그램 검증 모델의 설계

○

김진수* 김재웅** 황선명*** 김치수****

*건양대학교 정보전자통신공학부, **공주문화대학 컴퓨터정보과

대전대학교 컴퓨터공학과, *공주대학교 전자계산학과

jinskim@kytis.konyang.ac.kr

A Design of Diagram Verifying Model using the Frames

Jin-Soo Kim* Jae-Woong Kim** Sun-Myung Hwang***
Chi-Soo Kim****

*Dept. of Information & Electronics Eng., Konyang University

**Dept. of Computer Information, Kongju Culture College

***Dept. of Computer Engineering, Taejon University

****Dept. of Computer Science, Kongju University

요약

본 논문에서는 최근 객체지향 설계에 많이 사용되고 있는 객체지향 다이어그램들의 일관성과 완전성을 해결하기 위하여 [1]에서 제공된 일관성과 완전성 규칙들을 기반으로 다이어그램들을 검증할 수 있는 검증 모델을 프레임에 이용하여 설계하였다. 본 논문에서는 프레임을 이용한 설계 과정을 보이기 위하여 다이어그램의 일관성과 완전성을 검증할 수 있는 검증 모델에 대한 각 다이어그램을 대상으로 정보를 유도하였고 유도된 정보를 기반으로 검증에 기초가 되는 공통 내부 표현을 프레임을 이용하여 설계하는 과정을 제시하였다. 설계된 검증 모델은 다양한 객체지향 다이어그램에 대하여 적용이 가능하며 기존의 다이어그램 도구와도 결합이 가능하도록 구성되어 있다.

1. 서론

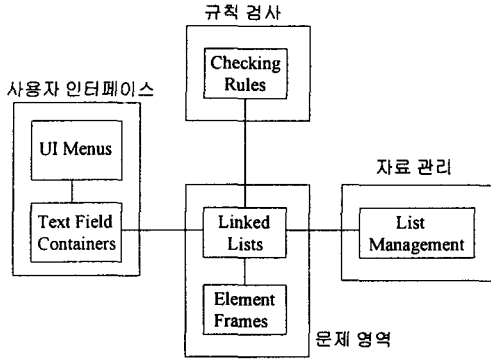
객체지향 개발방법을 사용하여 소프트웨어를 개발할 때, 전체 시스템에 대한 객체지향 설계는 많은 종류의 다이어그램으로 표현될 수 있다. 이러한 다이어그램들을 효과적으로 작성하고 관리하기 위하여 개발자들은 다이어그램 도구들을 사용할 수 있다. 이러한 도구들은 다이어그램을 그릴 수 있게 해주고 수정할 수 있게 해주며 외부 파일로 저장도 가능하게 해준다. 본 논문에서 설계한 검증 모델은 이러한 도구를 통하여 그려진 다이어그램에 대한 필요한 정보를 유도할 수 있게 해주고 이렇게 유도된 정보를 프레임을 이용하여 공통적인 내부 표현으로 저장하게 해준다. 이러한 다이어그램에 대한 공통적인 내부 표현에 대하여 검증 모델은 [1]에서 제시한 일관성과 완전성을 검사하는 규칙들을 호출하여 일관성

과 완전성을 검사하고 그 결과를 메시지로 출력하도록 하였다. 본 논문에서 설계한 검증 모델은 다양한 객체지향 다이어그램에 대하여 적용이 가능하며 기존의 다이어그램 도구와도 결합이 가능하도록 구성되어 있다. 2장에서는 검증 모델의 각 서브시스템들을 객체지향 다이어그램으로 표현하고, 3장에서는 다이어그램으로 표현된 각 서브시스템들에 대한 공통 내부 표현을 프레임을 이용하여 표현하고, 4장에서 결론을 맺는다.

2. 검증 모델의 구성

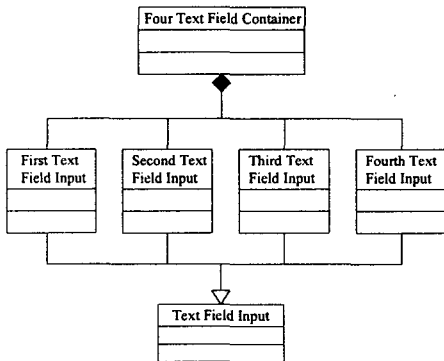
본 논문에서 제안한 검증 모델은 다음 <그림 1>과 같이 크게 4개의 서브 시스템들로 구성된다. 첫 번째는 사용자 인터페이스 서브시스템으로 Text_Field_Containers와 UI_Menu로 구성되는데 Text

_Field_Containers는 각 항목이 텍스트 필드를 갖는 텍스트 필드의 항목들의 집합이라고 할 수 있고 UI_Menus는 사용자의 입력을 도와주는 기능들을 제공한다.



<그림 1> 검증 모델의 구성

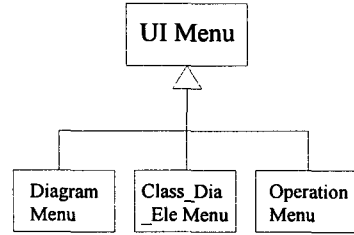
두 번째는 문제영역 서브시스템으로 Linked_Lists와 Element_Frames로 구성되는데 Linked_Lists는 single-linked-list를 생성하기 위해서 Element_Frames의 서비스를 이용하고 Text_Field_Containers로부터 받은 정보를 single-linked-list의 요소에 저장한다. 세 번째는 자료관리 서브시스템으로 그 요소인 List_Management는 single-linked-list를 외부 파일로 저장하기도 하고 저장된 파일로부터 다시 생성하기도 한다. 마지막 서브시스템은 규칙 검사 서브시스템으로 그 요소인 Rule_Based_Checking은 single-linked-list로부터 자료를 얻기 위해서 Linked_Lists의 서비스를 이용한다.



<그림 2> Text_Field_Containers에 대한 클래스 다이어그램

Text_Field_Containers는 <그림 2>와 같이 네 종류의 텍스트 필드 항목들을 가지고 있으며 프레임의 슬롯에 값을 넣는 것에 따라 종류가 결정된다.

사용자 인터페이스 서브시스템의 UI_Menu는 <그림 3>에서와 같이 다이어그램 메뉴, 다이어그램 요소 메뉴, 오퍼레이션 메뉴와 같은 클래스들로 구성되어 있다.



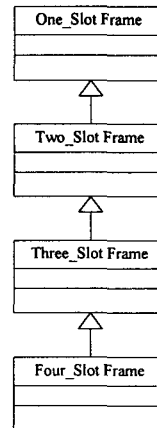
<그림 3> UI_Menu에 대한 클래스 다이어그램

문제 영역 서브시스템의 Linked_Lists는 <그림 4>와 같이 SL-List 클래스와 SL_List_Element 클래스로 구성되어 있다.



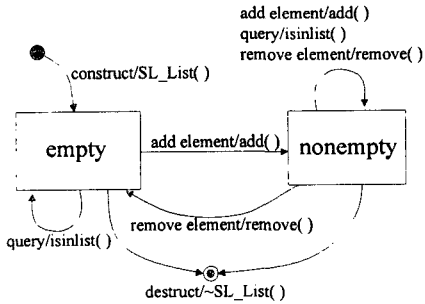
<그림 4> Linked_Lists에 대한 클래스 다이어그램

문제 영역 서브시스템의 Element_Frame은 <그림 5>와 같이 네 클래스로 구성되어 있다.



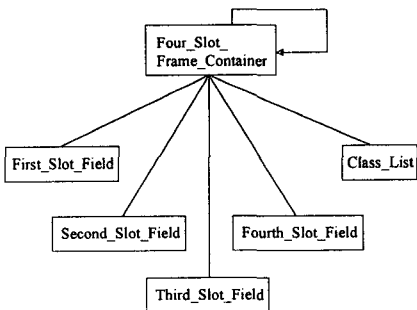
<그림 5> Element_Frame에 대한 클래스 다이어그램

SL_List 클래스는 <그림 6>과 같이 empty와 nonempty의 두 상태를 가지는데 클래스 생성자 함수인 SL_List()가 수행되면 비어있는 single-linked-list가 생성되고 그 상태는 empty 상태가 된다.



<그림 6> SL_List 클래스에 대한 상태 다이어그램

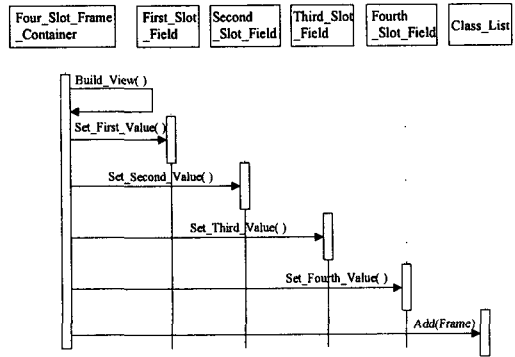
먼저 다이어그램 메뉴에서 클래스 다이어그램이 선택되고, 다이어그램 요소 메뉴에서 클래스 다이어그램의 요소인 relationship이 선택되고, 오퍼레이션 메뉴에서 오퍼레이션 add()가 선택되면, 네 개의 슬롯을 갖는 하나의 프레임이 생성된다. 먼저 사용자로부터 관련성의 이름을 입력받아 프레임의 처음 슬롯에 넣고 다음으로 관련성의 subset 이름을 받아 프레임의 두 번째 슬롯에 넣는다. 관련성이 시작되는 클래스의 이름을 입력받아 관련성 프레임의 세 번째 슬롯에 넣고 마지막으로 관련성이 종료되는 클래스의 이름을 입력받아 관련성 프레임의 네 번째 슬롯에 넣으면 이 프레임은 single-linked-list인 ClassList에 하나의 요소로 추가된다. 다음 <그림 7>은 이러한 시나리오를 보여주는 객체 다이어그램이다.



<그림 7> 관련성을 추가하는 객체 다이어그램

ClassList 리스트에 하나의 관련성을 추가하는 시

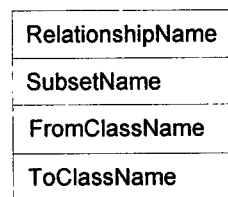
나리오는 다음 <그림 8>과 같이 순차 다이어그램으로도 나타낼 수 있다.



<그림 8> 클래스를 추가하는 순차 다이어그램

3. 프레임을 이용한 검증 모델의 설계

본 논문에서 설계한 검증 모델은 다이어그램 도구들을 통하여 그려진 다이어그램에 대한 필요한 정보를 유도할 수 있게 해주고 이렇게 유도된 정보를 프레임을 이용하여 공통적인 내부 표현으로 저장하게 해준다. 각 다이어그램의 정보를 공통적인 내부 표현으로 저장하기 위하여 프레임이 필요한데 다음 <그림 9>는 Relationship 타입에 대한 프레임으로 RelationshipName, SubsetName, FromClassName, ToClassName으로 구성되어 있다.



<그림 9> Relationship 타입에 대한 프레임

이와 유사하게 검증 모델의 각 다이어그램에서 제공되고 있는 타입들에 대한 프레임은 다음 <그림 10>과 같다.

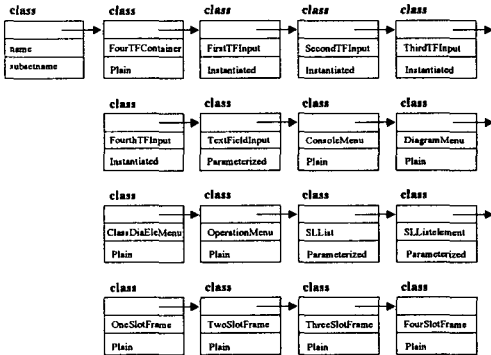
검증 모델의 각 다이어그램에 대한 일관성과 완전성을 검증하기 위하여 각 다이어그램들은 single-linked-list 형태로 표현될 수 있으며, <그림 11>에서 제공하고 있는 표현은 <그림 2>, <그림 3>, <그림 4>, <그림 5>의 클래스 다이어그램에서 유도된 정보를 가지고 있는 검증 모델의 공통된 내부 표현

이라고 할 수 있다.

Class	Relationship	ClassAttrOper
Name	Name	Name
SubsetName	SubsetName	SubsetName
StateAction	FromClassName	ClassName
Name	ToClassName	State
TransitionAction	Transition	Name
Name	Name	SubsetName
LinkMessage	FromStateName	ClassName
Name	ToStateName	Link
Object()	TransitionAction	Name
Name	Object	ClassName
ObjectAttr	Name	Interaction
Name	FromObjectName	Name
ObjectName	ToObjectName	FromObjectName
InteractionMessage	TransitionAction	ToObjectName
Name		TransitionAction

<그림 10> 각 타입에 대한 프레임들

객체 다이어그램, 상태 다이어그램, 순차 다이어그램에 대한 공통 내부 표현도 프레임에 이용하여 <그림 11>과 같은 형태로 표현될 수 있다.

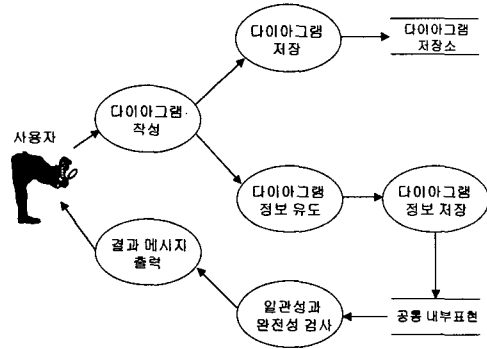


<그림 11> 검증 모델에 대한 공통 내부 표현

본 논문에서 설계한 검증 모델은 다이어그램 도구들을 통하여 그려진 각 다이어그램에 대한 필요한 정보를 유도하고 유도된 정보를 <그림 11>과 같은 공통 내부 표현으로 저장한 다음 이러한 내부 표현들을 입력으로 하여 일관성과 완전성을 검사하는 규칙들을 호출하여 일관성과 완전성을 검사하고 그 결과를 메시지로 출력하도록 하고 있다. 다음 <그림 12>는 검증 모델의 자료흐름도를 보여주고 있다.

4. 결론

본 논문에서 설계된 검증 모델은 다이어그램에서 유도된 정보를 기반으로 일관성과 완전성을 검증하



<그림 12> 검증 모델의 자료흐름도

는데 필요한 다이어그램의 정보를 공통적인 내부 표현으로 저장할 수 있도록 하기 위하여 프레임에 이용하였다.

본 논문에서는 이러한 과정을 보이기 위하여 다이어그램의 일관성과 완전성을 검증할 수 있는 검증 모델에 대한 각 다이어그램을 대상으로 정보를 유도하였고 유도된 정보를 기반으로 검증에 기초가 되는 공통 내부 표현을 프레임에 이용하여 설계하는 과정을 제시하였다. 설계된 검증 모델은 다양한 객체지향 다이어그램에 대하여 적용이 가능하며 기존의 다이어그램 도구와도 결합이 가능하도록 구성되어 있다.

참고문헌

- [1] 김진수 외, "UML 다이어그램간의 일관성과 완전성을 위한 검증 규칙 생성에 관한 연구", 한국멀티미디어학회 논문지, 제3권, 제3호, 290-298, 2000
- [2] Terry Quatrani, *Visual Modeling with Rational Rose and UML*, Addison-Wesley, 1998
- [3] Edwards, J. M. and Henderson-Sellers, B., "A graphical notation for object-oriented analysis and design", *Journal of Object-Oriented Programming* 4(9), 53-74, 1994
- [4] Robert H. Bourdeau and Betty H. Cheng, "A Formal Semantic for Object Model Diagrams", *IEEE Trans. on Software Engineering*, Vol. 21, No. 10, pp. 799-821, 1995
- [5] Paul Hramon, Mark Watson, *Understanding UML: The Developer's Guide*, Morgan Kaufmann Publishers, Inc., 1997