인터넷 기반의 공동 작업을 위한 UML CASE 도구의 동작환경 구성방법

최환복* · 김윤호*

*안동대학교

Environment Configuration of UML CASE Tool for Internet based Collaboration Works

Hwan-bok Choi* · Yun-ho Kim*

*Andong National University

E-mail : choibeta@gmail.com

요 약

본 논문에서는 공동 작업을 지원하는 CASE 도구를 위한 동작환경 구성방법을 제안하고자 한다. 분산된 위치에서 모델 공유를 위한 공유방법을 정의하고 공동 모델 저장소를 구성한다. 또한 시간 경과에 따른 작업 상황 비교를 위한 히스토리 기능 및 작성한 모델에 대해 책임을 부여하기 위해 사용자 인증을 통한 작업자 관리를 설계한다. 본 논문에서 제시하는 동작환경 구성방법은 분산된 위치에서 협업이 가능한 CASE 도구 개발에 기여할 것으로 기대된다.

ABSTRACT

In this paper, we present an environment configuration of UML CASE tool for internet based collaboration work. We define a method of model sharing in distributed location and construct shared repository. Also we design for model history among work processes and to give responsibility using user authentication. Its result will contribute development of CASE tool supporting collaboration work.

키워드

UML, Object-oriented, Collaboration work, CASE tool

Ⅰ. 서 론

1997년 UML이 처음 발표된 후 안정적이며 일반적인 통합 모델링 표기법으로써 널리 사용되었다. 그 후 UML을 통한 효과적이고 편리한 모델링을 지원하기 위해 다양한 통합 모델링 CASE 도구[1][2][3]들이 개발되었다. 하지만 시간이 지남에 따라 프로젝트의 규모가 커지고 분산된 위치에서 여러 사람이 참여하게 되었다. 이렇게 분산된 형태의 참여는 공동 작업 측면에서 기존 CASE 도구의 한계를 가져왔으며, 분산된 위치에서 공동 작업이 가능한 인터넷 기반의 CASE 도구가 제안 되었다. 현재의 인터넷 기반의 공동 작업을 지원하는 CASE 도구[4][5][6]는 여러 가지가 있지만 이들은 UML로 작성한 모델에 대해 분산된 위치에서 공동의 저장소를 통한 단순한 공유만 가능하다. 공동 작업 CASE 도구를 위해서는 공동의 작업공간뿐만 아니라 프로젝트 관리, 작성자 관리와 같은 동작 환경 구축이 선행되어야 한다. 따라서 본 논문에서는 인터넷 기반의 공동 작업을 위한 UML CASE 도구의 동작환경 구성방법을 제시하고자 한다. 또한 공동 작업을 위한 모델 공유방법과 모델의 효과적인 관리를 위한 프로젝트 및 모델 작성자 관리 그리고 모델링 과정에서 이전 모델 검토를 위한 모델 히스토리를 제시하고자 한다.

II. 공동 작업을 위한 동작환경구성

2.1 공동작업 환경 구성

분산된 위치에서 여러 사람이 참여하는 공동 작업을 위해서는 원격지 사이에 공통의 모델이 공유되어야 하고 일관성 있는 모델 관리가 필요하다. 이를 위해서는 한곳에서 모델이 유지되고 관리되어야 한다. 따라서 본 논문에서는 모델 공유를 위해 그림 1과 같이 클라이언트-서버 구조를 사용한다.

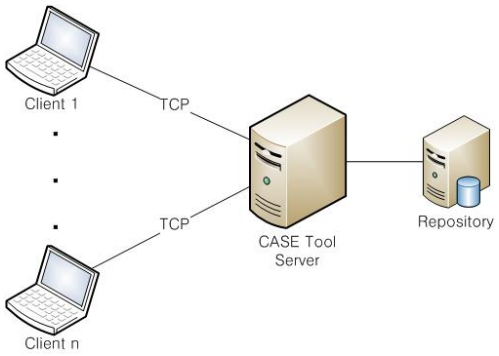

그림 1. 공동 작업을 위한 UML CASE 도구의 동작환경 구조

공동작업 환경은 모델을 작성하는 클라이언트, 클라이언트가 작성한 모델을 수신하고 모델 공유를 위해 클라이언트에게 모델을 전송하고 모델을 저장하는 서버(CASE Tool Server) 그리고 모델 저장을 위한 저장소(repository)로 구성한다.

클라이언트는 로컬 상에서 모델을 작성하고 원하는 시점에 TCP를 이용해 모델공유를 서버에 요청한다. 서버는 클라이언트로부터 모델을 수신하고 이를 저장소에 저장한다. 그리고 모델 공유를 위해 저장한 모델을 다른 클라이언트에게 전송한다. 클라이언트는 서버로부터 모델을 수신하고 이를 이용해 모델을 다시 표현한다.

2.2 모델 히스토리

객체지향 모델링은 순차적인 폭포수 모델과는 다르게 전체 프로세스 내에서 보다 나은 모델을 찾기 위해 지난 모델을 검토하는 유연한 모델링이 수행된다[7]. 따라서 본 논문에서는 이와 같은 유연한 모델링 지원을 위해 모델 히스토리 기능을 제공한다. 본 논문에서 제시하는 공동작업 환경에서는 클라이언트가 작성한 모델이 다른 클라이언트와 공유됨과 동시에 모델 유지를 위해 저장소에 저장된다. 이때 과거 모델을 덮어 쓰는 것이 아니라 이전 모델과 구분해서 별도로 저장함으로써 모델 히스토리 기능을 제공한다.

저장되는 모델간의 구분을 위해서는 모델을 유일하게 식별할 수 있는 식별자가 필요하다. 클라이언트가 직접 입력할 수 있지만 편의성을 위해 서버에서 저장시간을 식별자로 이용한다. 또한 여러 종류의 모델이 있을 경우 모델 종류 구분자 역시 필요하다. 따라서 모델 히스토리 관리를 위해 필요한 정보는 다음과 같다.

- 모델 작성 시각
- 모델 종류 식별자

2.3 프로젝트 관리 및 작성자 흔적 관리

여러 사람이 참여하는 공동 작업에서는 특정 시점에 하나의 프로젝트만 있는 것이 아니라 다수개의 프로젝트가 있을 수 있다. 프로젝트 간 모델을 구분하지 않으면 모델이 뒤섞여 의미가 모호해지기 때문에 프로젝트를 관리해야 한다. 이를 위해서 다이어그램을 저장할 때 소속 프로젝트를 같이 저장하도록 하여 프로젝트별 다이어그램을 관리한다.

프로젝트 관리에서 프로젝트를 사용자가 임의로 프로젝트를 입력하면 무분별한 프로젝트 생성을 통한 혼란을 야기할 수 있다. 따라서 권한을 부여받은 몇몇 사용자를 통해 프로젝트를 생성하고 등록된 프로젝트에 대해서만 모델을 저장하도록 하여 혼란을 줄이고 효율성을 높이도록 한다.

공동 모델링은 여러 이해관계자들이 참여한다. 여러 사람의 의견이 반영된 모델링은 이해관계자의 요구사항을 반영하지만 추후 모델링에 대한 책임을 명확하지 않으면 모델에 대한 혼란이 생긴다. 이러한 문제를 해결하기 위해서는 작성자에 대한 흔적 관리가 필요하다. 작성자 흔적 관리는 모델을 서버 측에 저장할 때 작성자에 대한 정보를 같이 저장함으로써 해결할 수 있다. 작성자에 대한 정보는 여러 방법을 통해 획득할 수 있지만, 일반적으로 많이 사용되는 로그인을 통해 획득한 정보를 사용하도록 한다.

따라서 프로젝트 관리 및 작성자 흔적 관리를 위한 요소는 다음과 같다.

- 등록된 프로젝트 리스트
- 작성자 정보
- 소속 프로젝트 식별자

2.4 모델 유지를 위한 정보 구성

원격지상의 공동 작업을 위해서는 작성한 모델을 여러 클라이언트와 공유해야 한다. 기존의 공동 작업 지원 도구는 작성한 모델을 그림형태로 변환해 유지하였지만 본 논문에서는 모델의 CRUD를 용이하게 하기 위해 모델에서 사용되는 요소들을 독립된 형태로 구성하고, 이러한 요소를 이용해 모델을 표현하기로 한다.

그림 2는 클래스 다이어그램을 이용해 UML 모델을 구성하는 요소를 나타낸 것이다. UML 모델을 구성하는 요소는 어떠한 항목을 의미하는 노드(node)와 노드를 연결하는 연결선(edge)로 분류할 수 있다. 따라서 본 논문에서는 모델을 구성

하는 요소를 노드와 연결선으로 구분하고 이들을 모델을 공유하고 유지하도록 한다. 하나의 모델에는 다수개의 노드와 연결선이 존재할 수 있기 때문에 이들을 collection 형태로 묶어서 관리할 필요가 있다. 이를 위해 일반적으로 가장 많이 사용하는 list 형태를 이용해 노드와 연결선을 유지하도록 한다.

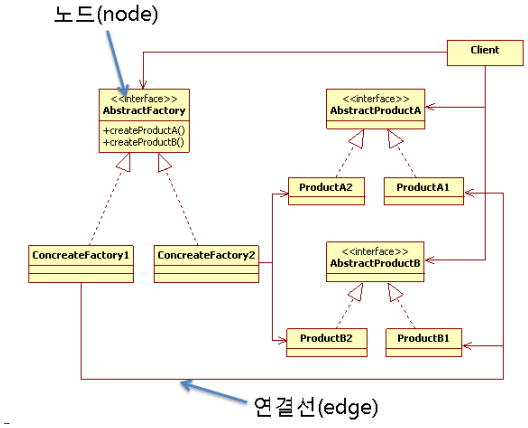

그림 2. UML 모델을 구성하는 요소의 예

2.5 사용자 편의기능

객체지향 모델링은 프로세스 내에서 모델 검토뿐만 아니라 작성중인 모델에 대해서도 여러 번의 검토가 발생한다. 이렇게 작성중인 모델에 대해서도 모델 작성의 편의를 위해서 되살리기 기능을 제공하도록 한다. 되살리기란 가장 마지막에 수행된 작업을 취소하는 것으로 볼 수 있다. 이는 LIFO(Last In First Out)의 형태를 가지며, 자료구조 중에서 스택(stack)[stack]이 이에 해당한다. 따라서 명령어를 저장하는 "명령어 스택"(command stack)을 구성하고 사용자의 명령을 저장한다. 이후 사용자가 되살리기를 요청을 했을 때 명령어 스택에서 가장 마지막에 저장된 명령을 삭제함으로써 되살리기 기능을 제공한다.

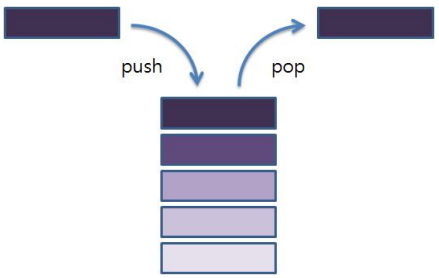

그림 3. 스택(stack)의 동작 과정

그림 4는 사용자가 다이어그램 작성 후 node 및 edge 리스트와 command stack 상태를 나타낸 것이다. node와 edge list에 사용자가 작성한 node와 edge가 저장되어 있으며 괄호는 작성한 순서를 나타낸다. 그림 4에서 N1, N2, E1, E2, N3, E3, N4, E4 순서로 작성되었으며, command stack에는 삽입된 순서를 저장하고 있다. 그림 5는 한 번의 되돌리기 후 node, edge, command stack의 상태를 나타낸 것이다. command stack에서 가장 마지막에 작성된 edge가 pop되었으며 이에 해당하는 edge E4가 edge list에서 삭제되었다.

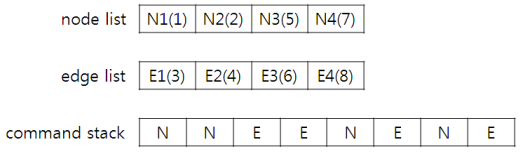

그림 4. 다이어그램 작성 후 node, edge, command stack 상태

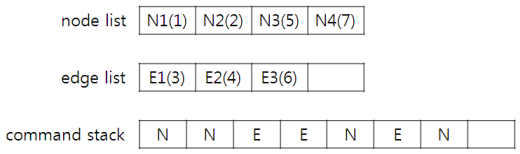

그림 5. 되돌리기 후 node, edge, command stack 상태

III. 공동 작업을 위한 UML CASE 도구의 동작환경 구현

본 장에서는 공동 작업을 위한 UML CASE 도구의 동작환경을 실제로 구현한 내용을 기술한다.

그림 6은 2.1절에서 2.4절까지의 내용을 바탕으로 공동작업 환경을 위한 저장소를 구성한 것이다.

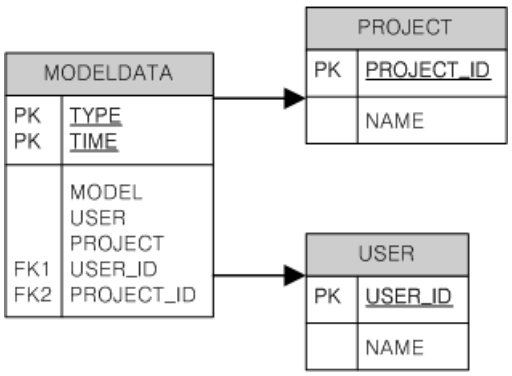

그림 6. 공동 작업 환경을 위한 저장소 구조

USER 테이블은 사용자의 정보인 ID와 이름을 저장하며, 모델을 저장소에 저장할 때 작성자 흔적으로써 사용된다. PROJECT 테이블은 등록된 프로젝트의 리스트를 의미하며, 프로젝트 별 모델

관리에 사용된다. MODELDATA는 클라이언트에 의해 작성된 모델 저장에 사용하는 테이블로 모델타입, 저장시간 그리고 실제 모델 정보가 저장되며, 모델의 작성자와 모델과 연관된 프로젝트를 참조한다.

그림 7은 2장에서 제안한 내용을 바탕으로 노드와 연결선을 이용하여 모젤 작성이 가능하도록 구현한 것이다. A영역은 모델작성에 필요한 노드와 연결선 그리고 작성한 모델을 서버로 전송하기 위한 전송 명령을 포함하고 있다. B영역은 실제 모델이 작성되는 영역으로, A영역의 노드와 연결선을 이용해서 B영역에서 모델을 작성하고 A영역의 전송 명령을 수행하여 작성한 모델을 공유한다.

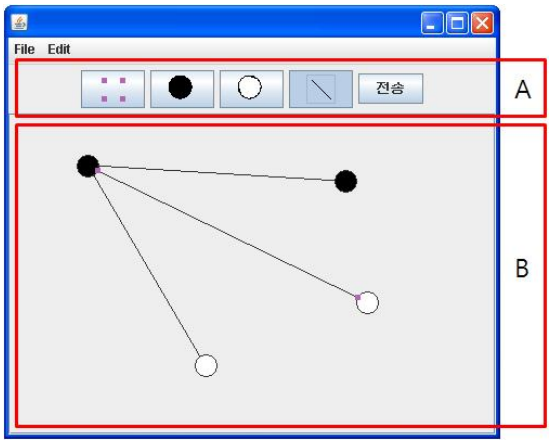

그림 7. 인터넷 기반의 공동 작업 도구의 프로토타입 및 모델 작성 예

Ⅳ. 결 론

UML이 발표된 후 UML 모델링을 지원하기 위한 다양한 CASE 도구가 개발되었다. 기존의 CASE 도구는 UML 모델링에 초점을 두고 로컬 상에서의 모델링을 지원하고 있다. 하지만 프로젝트의 규모가 커지고 다수의 이해관계자들의 원격지상에서 프로젝트에 참여함으로써 공동작업 측면에서 한계를 가져왔다. 이를 해결하고자 원격지상의 공동 작업을 지원하는 CASE 도구가 제안되었지만 이들은 원격 참여에만 초점을 두고 있어 동작환경에 대한 언급이 부족하다. 즉 원격지 상의 공동 작업을 위해 선행되어야 할 프로젝트 관리, 모델 작성자 관리와 같은 동작환경 구성방법이 존재하지 않는다. 본 논문에서는 인터넷 기반의 공동 작업을 위한 UML CASE 도구의 동작환경 구성을 제시하였다. 인터넷 기반의 원격지 상의 모델링을 지원하기 위한 동작환경을 구성하였고, 공동 모델링을 위해 공유되어야 하는 모델의 구성요소를 식별하였다. 또한 유연한 모델링을 지원하기 위한 모델 히스토리 관리 방법 및 모델의 효과적인 구성을 위한 프로젝트 별 모델관리 방법을 제시하였다. 그리고 작성한 모델에 대한 책임 관리를 위해 모델 작성자 관리 방법 및 모델링에서 사용자의 편의성을 위한 되돌리기 기능을 위한 명령어 스택을 제시하였다. 마지막으로 저장소 구성 및 제시한 내용을 바탕으로 사용 예를 보였다.

참고문헌

[1] IBM, Rational Rose, www.ibm.com
[2] Borland, Together, www.borland.com
[3] Visual Paradigm International, Visual Paradigm for UML, www.visual-paradigm.com
[4] Cao, S., Grundy, J., Hosking, J., Stoeckle, H., Tempero, E. and Zhu, N., "Generating Web-based User Interfaces for Diagramming Tools", *Proceedings of the Sixth Australasian conference on User interface*, Vol. 40, pp63-72, 2005
[5] Khaled, R., Mackay D., Biddle, R., Nobble, J., "A Lightweight Web-Based Caase Tool for Sequence Diagrams", *Proceedings SIGCHI-NZ Symposium on Computer-Human Interaction*, pp.55-60, 2002
[6] Mackay, D., Noble, J., Biddle, R., "A Lightweight Web-Based case tool for class diagrams", *Proceedings of the Fourth Australasian user interface conference on User interfaces*, Vol. 18, pp95-98, 2002
[7] Wirfs-Brock, *Object Design : Roles, Responsibilities, and Collaboration*, Addison-Wesley, 2002