복합문서를 이용한 UML 문서화

최길림* · 김태균**

*경남정보대학, **부산외국어대학교

UML Documentation Using Compound Document

Gil-rim Choi* · Tae-gyun Kim**

*Kyungnam College of Information and Technology, **Pusan University of Foreign Studies

E-mail : grchoi@kit.ac.kr

요 약

지난 10년간 소프트웨어 공학 분야의 연구개발 프로젝트에서 object oriented(OO) 기술이 생산성과 재사용성 향상에 중대하게 공헌 할 수 있었던 것을 보여주었다. 본 논문은 이들 가운데 두 가지 중요한 연구영역인 CASE 도구와 컴포넌트에 기반한 기술에 대한 이슈에 연관되어 있다. 본 논문은 CASE 도구에 컴포넌트 기반의 기술을 적용하는 예로써, COM/OLE 기술에 기반한 복합문서지원 기능이 있는 OO CASE 도구인, OODesigner를 개발하였다. 복합문서지원 기능이 있는 OODesigner가 윈도우 시스템의 다른 응용소프트웨어와 함께 상호운용 가능함으로써, 다른 CASE 도구보다 더욱 강력한 문서의 이용 환경을 제공할 수 있다는 것을 보여준다. OODesigner는 단지 UML 다이어그램을 모형화 할 수 있을 뿐만 아니라, OLE 서버와 컨테이너들과 함께 여러가지 다양한 종류의 문서를 협동하여 만들 수도 있으므로, 이 CASE 도구로 강력하고 일관된 문서화에 이용할 수 있다.

ABSTRACT

Two of major research fields in this area are concerned in issues for CASE tools and component-based technologies. This paper discusses an example of applying component-based technology to a CASE tool. This paper proposes the research experience gained while we have developing OODesigner, an OO CASE tool, with compound document support functionality based on COM/OLE technology. OODesigner can be used not only to model UML diagrams, but also to make documentation cooperatively with various kinds of OLE servers and containers. Therefore we can conduct powerful and consistent documentation with the tool. In this paper, we present design issues for incrementally implementing the compound document support facilities as a container and a server and show a brief sample for demonstrating the usability of the OLE enabled CASE tool.

키워드

CASE Tool, UML, Compound Document Support, COM/OLE

I. 서 론

지금까지 object oriented(OO) 소프트웨어의 개발을 용이하게 하기 위하여, 많은 OO 방법론들이 제안 되어졌다. 80년대 말 90년대 초에 OO 개척자에 의해 제안되어진 방법론들은 Rumbaugh의 OMT(Object Modeling Technique)[1], Coad의 OOA(Object Oriented Analysis)[2], Wirfs Brock의 RDD(Responsibility Driven Design)[3]이다. Booch, Rumbaugh와 Jacobson에 의해 제안되어진 UML(Unified Modeling Language)[4]은 90년대 초에 나타난 OO 방법들의 물결에 뒤이은 계승자이다. 90년대 말에 OMG(Object Management Group)는 OO 설계를 위한 표준 표기법으로서 UML을 채택을 하였다. OO방법들과 함께 연결된 CASE 기술은 또한 소프트웨어 생산성을 늘리는 주요한 요인 중에 하나이다. 그래서, 많은 상용 CASE 도구들 즉 Rational의 Rose와 TogetherSoft의 Together 같은 도구들은 90년대에 개발되어졌고 출시되어졌다. 이에 우리는 1996년에[5] UML 방법론을 지원하는, OODesigner라 불리는 CASE 도구를 개발했다. 현재 CASE 도구들의 대부분은 컴포넌트 기술 즉 컴포넌트 설계, 컴포넌트 적합, 컴포넌트 배포등을 지원하기 위한 방향이 거의 대부분이며, 그들 자신의 컴포넌트로 제공하지는 않는다.

하지만 우리는 CASE 도구를 위하여 복합문서 지원 같은 컴포넌트 기술 중에 하나를 사용하는

구현이 더욱 더 적절하다고 믿는다. 왜냐하면 컴포넌트 기술에 이해 구현되어지지 않은 도구들이 다른 컴포넌트와 상호 작용할 길이 없음으로, 그 도구가 다른 어플리케이션과 상호 운용하는 것은 어렵다. 그래서 컴포넌트기술 없는 CASE 도구들의 개발자는 CASE 도구들에 의해 필요 되어지는 모든 기능을 구현해야만 한다.

본 논문에서, 우리는 컴포넌트 기술 중 중요한 응용 중에 하나인, 복합문서를 지원하는 CASE 도구 즉, OODesigner를 향상시키는데 목적을 두었다. 컴포넌트 기술은, RPC(Remote Procedure Call) Automation, Active X 컨트롤들, TM(Transaction Server) 뿐만 아니라 복합문서를 지원하는 등 다양한 응용을 제공한다. 복합문서는 통합적으로 인식되어지는 하나의 환경 형태인, 유저인터페이스의 모임으로 조직되어졌으며, 복합문서는 일반 텍스트 파일, 음성파일, 영화 동영상 파일 같은 다른 데이터 형식을 담고 있는 데이터 구조를 포함한다. 복합문서는 또한 사용자에 의해 연결되고 상호작용할 수 있는, 프로그램을 담고 있는 응용 환경이다. 복합문서를 지원하는 CASE 도구는, 또 다른 복합문서기능을 지원하는 다른 어플리케이션과 내부적으로 상호작용하는 것을 지원한다.

소프트웨어 공학의 관점에서 UML 다이어그램은 일종의 문서로 간주되어질 수 있다. 그러므로 복합분서를 지원하는 CASE 도구는, 지원하지 않는 CASE 도구들보다 강력한 문서 환경을 제공할 수 있다. 다이어그램을 해석하기 위해서 음성, 비디오, 도표, 그림 같은 여러 매체는 일치하는 어플리케이션과 연결에 의해서 사용 할 수 있고, UML 다이어그램 안에 위치 할 수 도 있다. 그래서 그것은 소프트웨어 설계 문서의 이해력을 향상시킬 수 있다.

II. 시스템 디자인

2.1 OLE 복합문서

OLE 복합문서의 유용성은 꽤 알려져 있으며, Microsoft로부터 대부분의 어플리케이션은 이 기능[6]을 지원한다. 복합문서는 두개의 다른 어플리케이션에 의한 문서로부터 만들어 질 때 하나는 컨테이너로서 이고 다른 하나는 서버로서이다. OLE 컨테이너는 연결되거나 내장되어진 객체를 포함하는 엔티티이다. 복합문서들이 생성되어지고 관리되어지는 어플리케이션은 컨테이너 어플리케이션이며, 서버 어플리케이션은 object 를 구현한다. 왜냐하면 그들은 윈도우 object를 생성하고 유지하기 때문이다. OLE가 가능한 몇몇 어플리케이션은 서버와 컨테이너의 두가지 서비스를 제공하고, 다른 것들은 하나의 서비스만 제공한다. 예를들면, Microsoft의 Picture BOX와 Recorder는 서버로써 사용되어 질 수 있고, 일부 워드프로세스들은 컨테이너로서 사용되어질 수 있다. 그리고 Microsoft의 Word와 Excel은 서버뿐만 아니라 컨테이너로서도 사용되어 질 수 있다.

OLE가 가능한 어플리케이션으로서 유용한 어플리케이션을 위하여서는 두가지(서버와 컨테이너) 서비스를 다 제공해야 한다. 따라서 OODesigner는 두가지 서비스를 다 제공하기 위해 구현되어지고 설계되어진다. OLE가 가능하도록 하기 위해서, 구조적인 저장소를 지원하기위한 사용환경의 기능(모니커)과 통일된 데이터 전송 기술이 구현되어져야만 한다.

2.2. 최초 OODesigner 설계

OODesigner의 처음 버전은 1994년에 UNIX에서 개발되었다. 그것은 OMT 방법론을 지원했고, public domain에 공개되었다. 첫번째 버전이후 우리는 그것의 유용한 기능을 지속적으로 향상시켰다. 1998년에 OODesigner는 UNIX로부터 Microsoft Window 환경으로 이식작업이 수행되었다. 이식 작업 과정에서 OODesigner는 OMT 대신에 UML을 지원하도록 수정되었다.

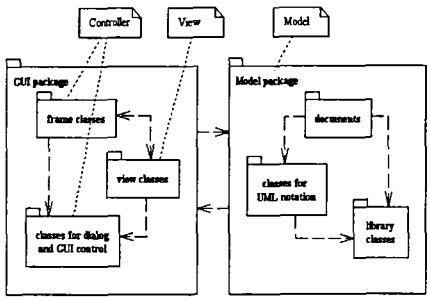

그림 1. OODesigner의 개요설계

OODesigner 설계에 MVC(Model View Controller) 패러다임[7]을 적용했다. 그림 1 은 OODesigner의 총체적인 설계를 보여준다. OODesigner는 대략 300 클래스로 구성되어있고 이들 클래스들은 그림 1에 보여주는 것처럼 패키지안에 적절히 분산되어있다. MVC 패러다임에 따라 OODesigner 사용자들은 컨트롤러에 속해있는 object를 통하여 모델링 정보를 입력 할 수 있고, 모델링 정보는 문서안에 저장되어 진다. 문서 object들 안의 모델 정보는 상응하는 view object들에 표시되어진다. 우리가 OLE기능을 지원하기 위한 OODesigner를 수정할 때 Controller와 View의 부분에서 많이 수정되지 않을 필요가 있다. 하지만 문서 클래스들에 연관되어있는 대부분의 클래스들을 수정해야만 한다.

2.3. 컨테이너 설계

컨테이너가 가능한 OODesigner를 위해 두개의 클래스에게 코드삽입을 하였다. 어플리케이션의 주요 클래스인 COODApp와 그림2처럼 수정되어진 설계인 CMainFrame이다.

그림 2에서 클래스 COODDoc의 슈퍼 클래스는 클래스 CDocument에 클래스 COleDocument로 변경되어야만 한다. 왜냐하면 OLE가 가능한 어플리케이션은 구조적 저장소를 사용해야만 하기 때문이다.

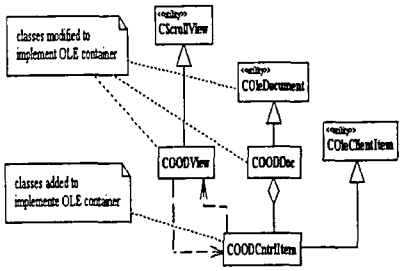

그림 2. 컨테이너 기능 관련 클래스 설계

클래스 COODCntrlItem은 OODesigner에 내장되고 연결되어진 OLE object를 관리하기 위하여 새롭게 추가되었다. 클래스 COODCntrlItem의 기능은 내장되어진 OLE object의 크기와 위치를 유지하고, 디바이스 컨텍스트에 내장되어진 OLE object를 그리고(draw), 내장되어진 OLE object를 활성화시킨다. 그리고 또한 새로운 클래스를 추가하기 위하여 여러 클래스들이 수정되어야 할 수도 있다. 클래스 COODView에 우리는 두가지 기능성을 추가했다. 하나는 OLE object를 생성하는 것이고 다른 하나는 object의 상태를 변화시키는 사용자 입력 과정이다. 클래스 COODDoc에 관해서는 세가지의 기능성을 추가했다. 추가되어진 기능성은 구조적인 저장소를 생성하는 것 과 OLE object들의 list를 유지하는 것과 OLE object들을 실현하는 것이다.

2.4 서버 설계

서버 기능을 위한 클래스다이어그램은 그림 3에서 보여준다.

그림 3에서 두 클래스들(COODView와 COODDoc)은 수정 되어 질 필요가 있고, 두 클래스들(CInPlaceFrame와 COODServerItem)은 새롭게 제공된다.

클래스 COODView와 클래스 COODDoc에 관해서, 우리는 상응하는 컨테이너 문서들에 OODesigner로부터 문서들이 내장되어질 때 수정한다. 클래스 CInPlaceFrame은 OODesigner이 메뉴 플레임과 함께, 상응하는 컨테이너의 메뉴 플레임으로 대치하기위한 목적으로 추가되어 졌다.

이경우 OODesigner로부터 내장되어진 문서는 그 위치에서 활성화 되어 진다. COODServerItem 클래스는 서버로서 실행하기 위해 활성화된 OODesigner의 인터페이스를 제공한다. 클래스 COODServerItem이 그 자신의 함수인, 컨테이너의 메타 파일에서 표시되는 문서인 OnDraw() 함수를 가진고 있음으로, 클래스 COODCntrlItem 보다 더 클래스 COODView에 결합도가 낮다.

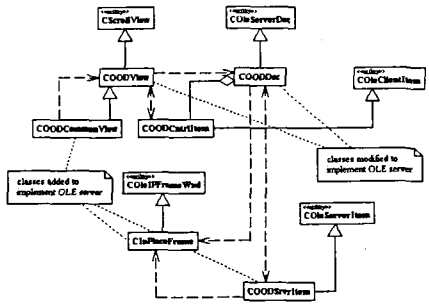

그림 3. 서버기능 관련 클래스 설계

III. OLE 가능한 OODesigner

OLE가 가능한 OODesigner 사용자들은, 다른 소프트웨어 즉 음성, 비디오, 그림, 챠트 object들을 수정할 수 있는, 다른 소프트웨어와 상호작용에 의해서 복합문서를 생성 할 수 있다. 그리고 우리는 복합문서지원의 연결가능 덕택으로 상응하는 다이어그램을 포함하는 텍스트로 된 다이어그램과, UML 다이어그램 사이의 일관성을 지킬 수 있다. 그래서 우리는 기능 없는 도구로 일하는 것 보다 쉽게 구성관리를 위한 행동을 보존 할 수 있다.

3.1 컨테이너로서 OODesigner

컨테이너가 가능한 OODesigner는 서버가 가능한 소프트웨어로부터 문서들을 포함할 수 있다. OODesigner의 유용성을 보여주기 위하여 우리는 point, lines, reactangles, cirles 그리고 TV와 같은 단순한 그래픽 엔티티를 그리는 목적인, 단순한 소프트웨어의 구현과 설계를 보인다. 그림 4는 단순한 프로그램을 설계하기위한 UML 클래스 다이어그램을 보여준다. 우리는 클래스들이 상응하는 object들을 구현하는데 드는 클래스들을 명백하게 설명하는 다이어그램과 그 다이어그램에 여러 개의 BMP 이미지들을 가지고 있다. 예를들면, 스크린의 오른쪽에 BMP 이미지로서 내장되어있는 TV object는 어떻게 그것이 보이는 바와 같이 가정되는지 보여준다. 그리고 아래에 보이는 여러 팝업이미지는 팝업 클래스가 상응하는 팝업메뉴라는 것을 말해준다. 예로서 소프트웨어의 스크린 샷을 위한 이미지의 왼쪽아래에 CDrawerApp, CMainFrame, CDrawerView 클래스들의 역할을 볼 수 있다.

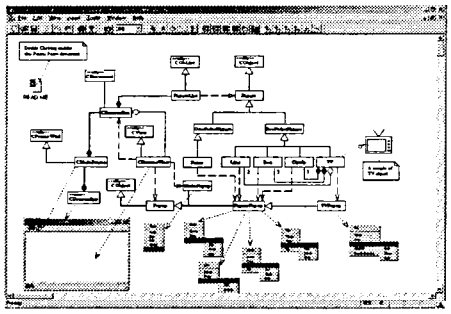

그림 4. 여러 문서객체를 포함하고있는 OODesigner의 스크린 샷

3.2 서버로서 OODesigner

서버가 내장되어있는 OODesigner의 문서들은 컨테이너가 내장되어있는 문서안에 위치 할 수 있다. Microsoft Windows 하에서 기능되어지는 대부분의 워드 프로세스들이 컨테이너로서 역할을 할 수 있으므로, OODesigner의 문서는 워드 프로세스 안에서 편집되어질 수 있다. UML 다이어그램 문서들로서 사용 되어질 수 있다. 예를 들면, 그 UML 다이어그램들은 보고서들이나 논문들에 삽입될 필요가 있다. 만약 그 CASE 도구가 서버로서 역할을 할 수 없다면, 다이어그램들의 이미지를 캡쳐한 BMP파일은 만들어져야 하고, 그것은 워드프로세스의 문서들에 삽입 되어 져야만 한다. 이 경우 만약 그 다이어그램이 수정되어 졌다면, 다이어그램들을 캡쳐하고 삽입하는 것을 반복해야 한다. 만약 CASE 도구가 OODesigner와 같이 서버로서 할 수 있다면, 워드프로세스에서 수정되어진 UML 문서는 일관성있게 유지되어질 수 있다.

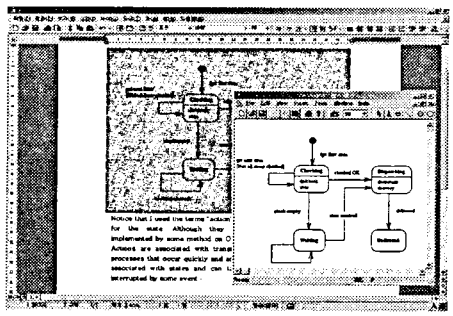

그림 5. MS 워드 문서안에 서버로서 실행되고있는 OODesigner의 스크린 샷

그림 5에서 Microsoft Word는 컨테이너로서 동작 되어지고, OODesigner는 서버로서 동작되어진다. 다른 문서들에 위치한 OODesigner의 문서는 다른 내장 객체처럼 편집되어 질 수 있다.

IV. 결론

컴포넌트에 연관된 기술을 따라 복합문서지원이 발달하였고, 1990년대에는 대부분 CASE 도구들이 이 기술을 적용하지 않았다. 만약 이들 CASE 도구중 컴포넌트 기반의 기술을 기반으로, 복합문서를 지원한다면, 다른 CASE 도구들보다 문서 환경을 더욱 강력하게 제공 할 수 있다. 예를 들면, 여러 매체와 함께하는 문서는 내부적으로 편집 가능한 멀티미디어 편집 도구들과 상호작용이 가능하고, UML 다이어그램과 UML 다이어그램을 포함하는 문서들 사이의 통일성을 쉽게 유지할 수 있다.

본 연구에서의 CASE 도구 즉, OODesigner는 OLE가 가능하도록 다시 만들어졌다. 또한 본 논문에서 OODesigner가 OLE를 가능하게 하기 위한 시스템 설계를 설명하였고 그리고 그것의 구현 결과를 예들을 통하여 OLE가 가능하다는 것이 유용 하다는 것을 보여 주었다.

향후 연구 과제로는 XMI 문서를 지원하는 OODesigner가 되도록 하는 것이다.

참고문헌

[1] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenzen, Object Oriented Modeling and Design, Prentice Hall, 1991.

[2] P. Coad, E. Yourdon, Object Oriented Analysis, Yourdon Press, 1990.

[3] R. J. Wirfs Brock and R. E. Johnson, "Surveying Current Research in Object Oriented Design", Communications of the ACM, 33(9), pp. 104 124, September 1990.

[4] M. Fowler and K. Scott, UML Distilled: Applying the Standard Object Oriented Modeling Language, Addison Wesley, 1997.

[5] Taegyun Kim and Gyusang Shin, "Restructuring OODesigner: A CASE Tool for OMT", In Proceedings of ICSE98, pp. 449 451, 1998.

[6] D. Chappell, Understanding ActiveX and OLE, Microsoft Press, 1997.

[7] Y. P. Shan, "An Event Driven Model View Controller Framework for Smalltalk",_In Proceedings of OOPSLA89, pp. 347 352, New Orleans, USA, October 1989.