Journal of Korea Multimedia Society Vol. 8, No. 4, April 2005(pp. 575-585)

객체 지향 CASE 도구 OODesigner에 대한 OLE-Container 기능의 설계 및 구현

최길림†, 김태균††

요 약

1990년대 이후로 객체 지향과 관련된 기술이 학계 및 산업계 전반에 정착됨으로써 소프트웨어의 생산성과 재사용성 향상에 큰 도움이 되고 있다. 다양한 관점에서 연구되고 있는 객체 지향 기술 중에서 객체 지향 기술의 구체적인 응용 분야로 언급될 수 있는 한가지 분야가 CASE 도구에 관련된 것이며 또 다른 하나가 컴포넌트 기술에 관한 것이다. 본 논문에서는 이러한 컴포넌트 기술을 CASE 도구에 적용한 예를 다룬다. 본 논문에서는 기존에 개발되었던 객체 지향 CASE 도구인 OODesigner에 COM/OLE 기반의 복합 문서 지원 기술을 지원하도록 추가로 개발한 연구 결과를 제시한다. 본 연구의 결과로 구현된 OLE 컨테이너 기능을 갖는 OODesigner는 윈도우 시스템에서 OLE 서버로 작동될 수 있는 응용 프로그램들과 상호 협동 작용을 하며 수행될 수 있기 때문에 기존의 CASE 도구가 지원하지 못하는 유용한 문서화 기능을 제공할 수 있다. 즉 객체 지향 관련 설계 문서의 내용을 OODesigner로 편집함과 동시에 OLE 서버 기능을 지원하는 녹음기, 엑셀과 같은 소프트웨어들을 연결하여 수행함으로써 일관성 있고 막강한 문서화 작업을 수행할 수 있다.

Design and Implementation of OLE-Container for an Object-Oriented CASE Tool, OODesigner

Gil-Rim Choi†, Tae-Gyun Kim††

ABSTRACT

Research and development projects, in the past decade, has shown that object-oriented(OO) technology could importantly contribute in productivity and reusability improvement. There are various research areas related to OO technology. Two of major research fields in this areas 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 incrementally developing OODesigner, an OO CASE tool, with compound document support functionality based on COM/OLE technology. As OODesigner with OLE container functionality has interoperability with other application programs of Windows system which can be run as a OLE server, it can provide more powerful documentation environment than other CASE tools. OODesigner can be used not only to design UML diagrams, but also to make documentation cooperatively with various kinds of OLE-enabled software like Recoder and Excel. Therefore we can conduct powerful and consistent documentation with the tool.

Key words: CASE Tool(CASE 도구), UML, COM/OLE, Compound Document Support(복합문서지원)

※ 교신저자(Corresponding Author) : 최길림, 주소 : 부산광역시 남구 대연 3동 907-7(608-023), 전화 : 051)320-1464, FAX : 051)320-1370, E-mail : grchoi@kit.ac.kr,
접수일 : 2004년 8월 26일, 완료일 : 2004년 10월 27일
† 정회원, 경남정보대학 컴퓨터정보계열 부교수
†† 정회원, 부산외국어대학교 컴퓨터공학과 교수
(E-mail : ktg@pufs.ac.kr)
※ 본 연구는 한국과학재단 목적기초연구(R05-2002-000-00497-0)지원으로 수행되었음.

1. 서 론

객체지향 소프트웨어의 개발을 용이하게 하기 위해서 다양한 객체 지향 방법론들과 이를 지원하는 CASE 도구들이 개발되었다. 1980년대 후반에 발표된 대표적인 객체지향 방법론들로는 Rumbaugh의 OMT(Object Modeling Technique)[1], Coad의 OOA(Object Oriented Analysis)[2], Wirfs-Brock의 RDD(Responsibility Driven Method)[3] 등이 있고, 이들 방법론들에 대한 통합 노력의 결과로 UML (Unified Modeling Language)[4]이 정의되었다. UML은 1997년 OMG(Object Management Group)에 의해 객체 지향 설계를 위한 표준화된 기법으로 선정되었다. 그리고 생산성을 향상시키기 위하여 UML을 비롯한 객체 지향 방법론들을 지원하는 CASE (Computer Aided Software Engineering) 도구들이 개발되었으며, Rational 사의 Rational Rose[5]나 TogetherSoft 사의 Together[6] 같은 도구가 현재 널리 사용 중이다. 본 논문에 의해 연구 개발되고 있는 OODesigner도 객체 지향 방법론을 지원하는 CASE 도구로서 현재까지 나름대로의 성과를 얻은 바 있다[7,8].

기존의 CASE 도구들이 컴포넌트의 설계와 제작을 위한 기능의 제공하기 위해 노력하고 있음에도 불구하고 CASE 도구 자체가 컴포넌트로 구현된 경우는 없다. 즉 기존의 도구들이 컴포넌트로서 사용되도록 개발되지 않았기 때문에 복합 문서 지원(Compound Document Support) 기능과 같은 유용한 기술을 제공하지 못하고 있다. 컴포넌트 기술은 현재 보편화되고 있는 기술로서, 다른 응용 소프트웨어와의 상호 운용성(interoperability)을 목적으로 한다. 그러나 기존의 CASE 도구들은 다른 응용 소프트웨어와의 연결을 목적으로 개발되어있지 않기 때문에 도구에서 필요한 모든 기능을 도구 내에서 구현해야하는 부담을 갖고 있다. 예를 들어 UML로 작성된 설계 문서에 설계자의 음성을 이용하는 문서화를 하고자 하는 경우 복합 문서를 지원하는 CASE 도구는 기존에 존재하는 녹음기 응용 프로그램을 연결해서 사용하면 되지만 복합 문서를 지원하지 않은 CASE 도구는 소리를 저장하고 재생하는 기능을 직접 구현하여 CASE 도구의 부속 기능으로 추가하여야 한다.

본 연구는 기존 개발된 OODesigner에 컴포넌트 기술을 적용함으로써 OODesigner가 복합문서를 지원하는데 목적을 두고 있다. 본 논문에서는 복합 문서를 지원하기 위한 시스템 설계와 설계 과정에서 얻은 경험, 그리고 복합 문서를 지원함으로서 얻을 수 있는 장점을 구현 결과를 통해서 기술하고자 한다.

응용 소프트웨어를 컴포넌트로 만들기 위한 기술로는 Microsoft의 COM(Component Object Model) [9], Sun의 EJB(Enterprise Java Beans)[10] 그리고 OMG의 CORBA(Common Object Request Broker Architecture)[11] 등이 있다. 이들 세 가지 기술은 효용성, 표준화, 호환성, 성능 등의 관점에서 보편적 기능과 고유의 기능을 갖고 있으며 개발 환경에 따라 그 기술을 선택적으로 사용할 수가 있다. 본 연구의 경우 기존에 구현된 OODesigner가 Microsoft 윈도우 환경의 Visual C++ 언어로 구현되었기 때문에 컴포넌트 기술의 선택 시에 Microsoft 사의 COM을 선택하였다. OLE(Object Linking and Embedding)는 COM에 기초한 구조화된 저장소나 모니커(Moniker) 그리고 통일 데이터 전송(Uniform Data Transfer) 기술들을 조합하여 만들어낸 복합 문서 지원 기술이다.[9] COM/OLE 기술을 통하여 구현된 소프트웨어는 다양한 응용 분야의 문서 작성 소프트웨어간에 삽입(embedding)과 연결(linking)을 통한 협동 작업을 가능하게 하므로 보다 막강한 문서 작성 기능을 제공할 수 있다.

소프트웨어 공학적인 관점에서 볼 때 UML을 이용한 시스템 설계 작업도 일종의 문서화 작업이라 볼 수 있으므로, UML을 지원하는 CASE 도구가 COM/OLE 기술을 지원하게 되면 동영상, 음성, 차트나 그림 파일들의 복합적인 조합으로 이루어지는 문서화가 가능해 지므로 소프트웨어의 관리에 큰 도움을 얻을 수 있게된다. 더욱이 현재 사용되고 있는 대부분의 소프트웨어 개발 도구들이 주로 다이어그램과 텍스트 자료에 기반한 문서화를 이루고 있으므로, 복합 문서 지원 기능을 갖는 도구의 개발을 통해 멀티미디어 기술과 통합된 개발 환경을 갖추게 될 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존 UML Tool들에 Container 기능의 구현여부와, 복합 문서 지원 기능을 OODesigner에 적용하기 위한 설계 내용에 대하여 기술하며 3 장에서는 OLE 컨테이너의 구현 결과를 사용 예와 실행화면 위주로 설명한다. 아울러 4 장에서는 결론과 함께 향후 연구 내용에 대하여 서술한다.

2. 시스템 설계

본 장에서는 기존 UML CASE 도구에 대해서 알아보고, OODesigner에 복합 문서 기능을 추가하기 위해 수행된 설계 내용에 대하여 다룬다. 먼저 기존 UML CASE 도구에 컨테이너 기능이 있는지 알아본다. 다음에 복합 문서의 개요에 대하여 개략적으로 알아보고, 기존 OODesigner의 설계에 대해서 논한다. 마지막으로 복합 문서 기능을 추가하기 위한 설계 내용을 논한다. 여기서는 기존의 설계 중에서 변경이 이루어졌던 부분과 새로 추가된 부분을 구분하여 기술한다.

2.1 UML CASE tool 비교

본 논문의 당위성의 확인하기 위하여 기존에 발표된 UML CASE 도구들에 대한 비교를 실시하였다. 그 결과 표 1과 같은 결과를 얻을 수 있었으며 기존 UML CASE 도구 중에서 OLE 컨테이너 기능을 제공하는 도구는 확인 할 수 없었다. 본 논문의 결과로 구현된 OODesigner 만이 컨테이너 기능을 제공한다.

2.2 OLE 복합 문서

복합문서는 한 문서 내에 다른 응용 소프트웨어에서 사용하는 정보들을 포함(Embedding)하거나 또는 정보들의 링크를 포함(Linking)할 수 있는 문서이다 [13]. OLE는 Microsoft 에서 개발된 복합문서에 대한 표준 기본 틀로서, 복합문서를 작성하고 표시하는 API들의 집합을 정의한다. OLE 복합 문서의 유용성은 이미 잘 알려져 있으며 Microsoft 윈도우에서 실행되는 상당수의 응용 프로그램들이 복합 문서 기능을 지원한다. 서로 다른 두 개의 소프트웨어에서 나온 정보가 하나의 복합 문서 안에서 합쳐질 때 소프트웨어의 한 부분은 컨테이너로, 다른 부분은 서버로서의 역할을 각각 맡는다. 개개의 소프트웨어는 그 응용 목적에 따라 컨테이너로서만 작동하거나 혹은 서버로서만 작동하거나 혹은 컨테이너이자 서버로 동시에 작동하도록 구현된다. 예를 들어 Microsoft 윈도우의 그림판이나 녹음기 등은 서버로서만 작동하며, 흔글 워드 프로세서는 컨테이너로만 작동하고, Microsoft Word나 엑셀은 컨테이너이자 서버로 작동한다. 또한 컨테이너와 서버 간의 상호 작업은 삽입이나 연결 방식을 통해 이루어질 수 있다. 본 논문은 OODesigner가 컨테이너로서 작동하도록 하는 기능을 추가 개발한 연구 결과에 대한 것이다. OODesigner에 OLE 컨테이너 기능을 추가한 이유는 UML을 이용한 시스템 설계 작업 시에 동영상, 음성, 차트나 그림 파일들의 복합적인 조합으로 이루어지는 문서화가 가능하게 하기 위함이다.

OLE 지원 소프트웨어는 COM에 기초한 여러 기술인 구조화된 저장소, 모니커, 통일 데이터 전송 기술에서 정의되어 있는 인터페이스들의 함수들을 구현해야 한다. 시스템 개발자는 복합 문서 지원을 위한 컨테이너 기능을 만들고자할 때 IUnknown, IOleInPlaceFrame, IOlePlaceUIWindow, IOleInPlaceSite, IOleClientSite, IAdviseSink와 같은 인터페이스를 구현해야만 한다. 이러한 인터페이스들에 소속된 가상 함수들의 구현 기법은 매우 복잡하다. 그러나 Microsoft의 Visual C++ 와 MFC(Microsoft Foundation Class)에서 이들 인터페이스를 쉽게 구현할

표 1. 기존 UML CASE 도구의 비교표

제품명	회사	웹사이트	OLE 기능
OODesigner		http://munjong.pufs.ac.kr/ktg/ood.htm	O
Rational Rose 2003	IBM	http://www.rational.com	X
together control center 6.2	Borland	http://www.borland.com	X
Agora plastic 2005	(주)플라스틱소프트웨어	http://plasticsoftware.com	X
poseidon for UMLcu3.0	GentlewareAG	http://www.gentleware.com	X
enterprise architecture 4.5	Sparx Systems	http://www.sparxsystems.com	X
artiso visual case 2.7	Artiso visualcase	http://www.visualcase.com	X
visual paradigm for UML 4.0 pro	visual paradigm	http://www.visual-paradigm.com	X
Ameos 9.1.5	Aonix	http://www.aonix.com	X

수 있는 API들을 제공하기 때문에 OLE 기능의 구현자는 이들 API를 사용하여 상대적으로 수월하게 OLE 기능을 구현할 수 있다.

2.3 MVC 관점의 개략 설계

본 절에서는 OODesigner에 대한 연구 배경에 대하여 간략히 소개하고 OLE 기능이 추가되기 이전의 OODesigner 구조에 대한 설계를 개략적으로 기술한다. OODesigner의 처음 버전은 1994년에 OMT를 지원하기 위한 목적으로 Unix 운영체제 하에서 구현되어 public domain에 공개된바 있다. OODesigner의 첫 번째 버전 발표 이후에 계속적인 기능 추가와 refactoring 작업이 수행되었으며[7], 1998년 이후에 Unix 환경으로부터 Microsoft 윈도우 환경으로의 이식 작업이 수행되었다[8]. 이식 작업 과정에서 Microsoft 윈도우 버전의 구현 목표가 UML을 지원하도록 수정되어 UML의 각종 다이어그램 작성 기능과 C++/Java 코드 생성 기능을 제공하도록 구현된 바 있다. 그림 1은 OODesigner의 실행 화면 예이다.

OODesigner는 MVC 패러다임을 기반으로 설계되었다. 그림 2는 OODesigner의 패키지 구조를 보여주는 개략 설계로서 크게 사용자 인터페이스 관련 패키지와 모델 관련 패키지로 설계되었다. 그림 2에서 사용자 인터페이스 관련 패키지는 Controller 기능을 담당하는 프레임 관련 클래스들, 대화상자 및 컨트롤 클래스들을 위한 패키지와 View 기능을 담당하는 뷰 관련 클래스들을 위한 패키지로 구성된다. 모델 관련 패키지는 Model 기능을 담당하는 도큐먼트 관련 클래스들, 표기법 관련 클래스들과 라이브러리 클래스들로 구성된다.

그림 2의 클래스들의 기능에 대한 설명은 다음과 같다.

- 프레임 관련 클래스들: 사용자 화면과 메뉴를 구성하고 사용자 입력에 대한 제어를 담당한다.
- 대화 상자 및 컨트롤 클래스들: 도구 사용 시에 각종 자원 값을 입력하기 위한 대화 상자들과 도구의 이용을 용이하게 하는 트리, 툴바, 상태바 등과 같은 부수적인 컨트롤들을 구성하는 기능을 담당한다.

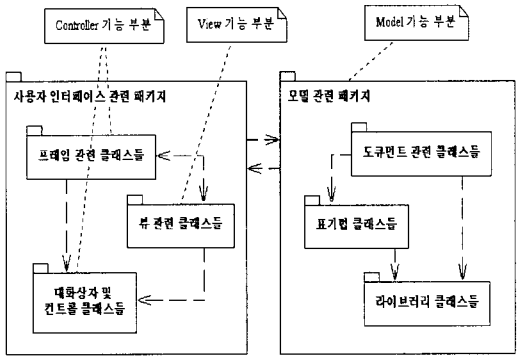


그림 2. OODesigner의 시스템 개략 설계

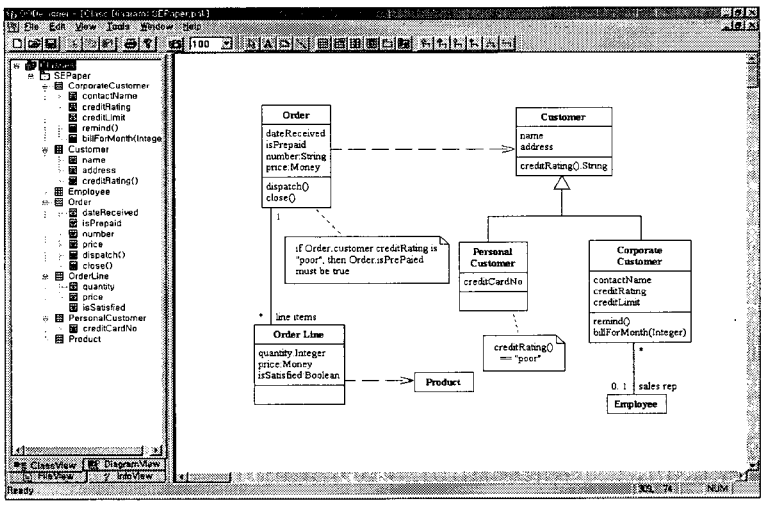


그림 1. OODesigner의 실행 화면

- 뷰 관련 클래스들: 도큐먼트에 저장되어 있는 모델 정보를 화면으로 디스플레이 하는 기능을 담당한다. OODesigner에 포함되는 OLE 객체들은 뷰 객체를 통해서 보여지므로 컨테이너 기능을 구현하기 위해서는 이 클래스들에 대한 기능의 추가가 필요하다.
- 도큐먼트 관련 클래스들: UML 모델과 관련되는 물리적인 데이터를 저장하는 기능을 담당한다. 도큐먼트에서 관리되는 정보에는 도구 상에서 명시되는 모든 표기법 객체의 좌표 값, 자원 값, 도구의 상태 값 등이 있다. 도큐먼트 클래스들의 중요 기능 중에 하나는 모델을 저장소에 직렬화(serialization) 하는 것이므로 OODesigner에 OLE 컨테이너 기능을 추가할 때 도큐먼트 클래스들에 대해 많은 수정이 필요하다.
- 표기법 클래스들: UML 다이어그램에 속하는 모든 표기법을 지원하는 기능을 담당한다. 즉 UML 클래스 다이어그램에 속하는 클래스, 패키지, 인터페이스, 일반화 관계, 집합화 관계 혹은 시퀀스 다이어그램에 속하는 객체, 생명선(life lane), 메시지 등 모든 UML 표기법을 모델링한 클래스들이다.
- 라이브러리 클래스들: 소프트웨어 구현 시에 보편적으로 사용되는 스택, 리스트, 배열 등을 구현하는 것들이다.

OODesigner는 약 200개의 클래스들로 구성되었다. 그러나 그림 2와 같은 MVC 패러다임에 따른 구조 설계에 따라 클래스 간의 결합도가 느슨하게 구현되었기 때문에 시스템의 보수유지가 용이하게 개발되었다. MVC 구조에 따라 도구 사용 시에 사용자는 컨트롤러에 속하는 객체들을 이용하여 모델 정보를 입력하고 그 정보는 도큐먼트에 저장되며 저장된 내용이 뷰에 디스플레이 되는 방식으로 도구가 작동된다. 이러한 설계 내용 중에서 OLE 기능의 추가 시에 컨트롤러와 뷰는 크게 영향을 받지 않았으나 도큐먼트 부분에서는 많은 수정이 이루어졌다. 그림 3은 OLE 기능 추가 이전에 설계되었던 모델 기능 부분의 클래스 다이어그램이다.

그림 3에 나타나있는 클래스들의 기능은 다음과 같다.

- COODView: MFC의 CScrollView 클래스로부터 상속되며 OODesigner에서 사용되는 모든 다이어그램들을 보여주는 클래스들의 상위 클래스로서 GetDocument() 함수를 이용하여 도큐먼트의 모델 정보를 가져온 후 화면에 디스플레이 하는 기능을 갖는다.

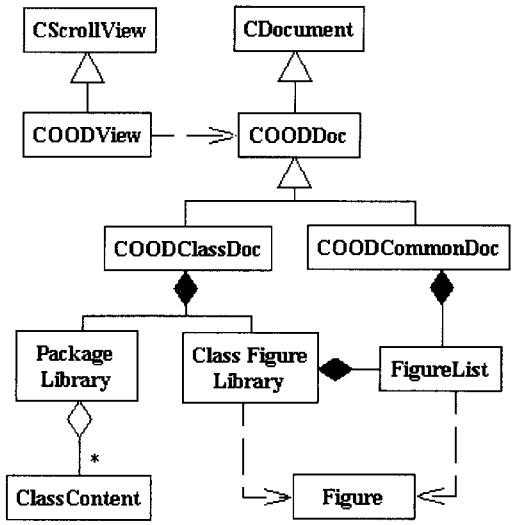


그림 3. 모델 관련 클래스 설계

- COODDoc, COODClassDoc, COODCommonDoc: OODesigner에서는 UML 다이어그램들의 특성을 고려하여 두 가지 종류의 도큐먼트를 분리해서 관리한다. 그 하나인 COODClassDoc 클래스는 UML 클래스 다이어그램을 위한 것이고 다른 하나인 COODCommonDoc 클래스는 클래스 다이어그램을 제외한 유즈 케이스 다이어그램, 시퀀스 다이어그램, 상태 다이어그램 등을 위한 것이다. 도큐먼트의 종류를 두 가지로 구분한 이유는 클래스 다이어그램의 경우는 논리적으로 정보 저장소의 역할을 하기 때문에 그림 정보와 클래스 내용에 대한 정보를 복합적으로 관리해야 하기 때문이고 나머지 다이어그램의 경우는 단순히 그림 정보만을 관리해도 되기 때문이다. 이들 두 클래스의 상위 클래스인 COODDoc 도큐먼트는 두 가지 경우 모두에 대하여 공통적으로 저장해야 하는 내용을 관리한다. OLE 기능을 추가 하기 이전의 설계 문서에서 특징적인 내용은 도큐먼트 클래스들의 상위 클래스가 MFC의 CDocument라는 점이다.
- PackageLibrary, ClassContent: 이들 클래스들

은 클래스 다이어그램들을 위한 논리적인 정보 저장소 역할을 한다. 즉 그림의 좌표와 관계 없이 도구 상에서 명시된 각 클래스의 명세에 대한 내용을 저장하는 클래스가 ClassContent이며 이들 객체를 전체적으로 관리하는 객체가 PackageLibrary이다.

- ClassFigureLibrary, FigureList: 이들은 다이어그램에 나타나 있는 UML 표기법 들의 좌표값과 관계되는 그림 정보들을 관리하는 클래스들로서 ClassFigureLibrary는 도구 상에서 여러 개의 뷰로 작성되는 여러 패키지들에 속하는 클래스 다이어그램 정보를 관리하기 위해 여러 개의 FigureList 객체를 이용한다.
- Figure: 이 클래스는 UML에서 사용되는 모든 표기법 객체들을 구현하는 클래스들의 최상위 클래스이다. 이 클래스의 하위 클래스에는 선, 점, 사각형, 삼각형 등 기본 표기법의 조합으로 구성되는 UML 표기법들이 존재한다.

2.4 컨테이너 기능 설계

OLE 컨테이너 기능 추가를 위해 핵심적으로 변경된 설계 사항은 그림 4와 같다. 기존의 클래스들 중에서 수정이 이루어진 클래스들은 COODView 클래스와 COODDoc 클래스이고, 새로 추가된 클래스는 COODCntrlItem 클래스이다.

COODDoc 클래스의 수정된 사항은 다음과 같다. COODDoc 클래스의 상위 클래스가 CDocument 클래스로부터 COleDocument 클래스로 바뀌었는데 그 이유는 OLE 기능의 구현 시에 구조화된 저장소(storage)를 사용하기 때문이다. 그리고 COODDoc 클래스에서 모델 정보를 저장하거나 읽어오는 기능을 하는 함수들인 Serialize(), OnOpenDocument(), OnNewDocuemnt()가 OLE 객체들의 리스트를 관리하며 구조화된 저장소를 처리할 수 있도록 변경되었으며, 루트 저장소를 생성하거나 초기화시키기 위하여 SetRootStg(), OnNewEmbedding(), OnOpenEmbedding()과 같은 함수들이 추가되었다.

COODView 클래스의 수정 사항은 다음과 같다. COODView 클래스에는 OLE 객체를 생성하고 객체의 상태 변화를 일으킬 수 있는 사용자 입력을 처리하는 기능들이 추가되었다. 즉 OnInsertObject() 함수를 통하여 OLE 객체를 삽입하고, OnObjectEdit(),

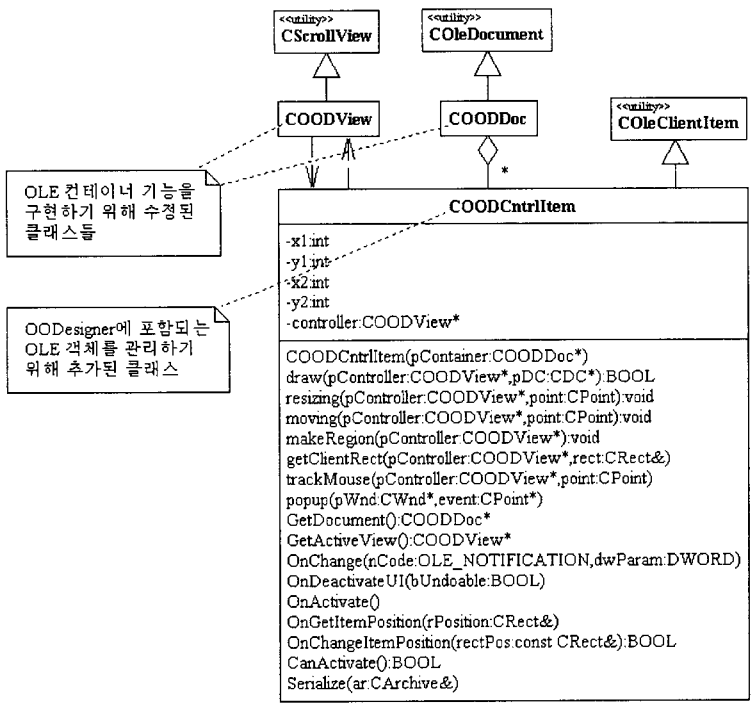


그림 4. 컨테이너 기능 설계

OnObjectOpen() 함수들을 통하여 삽입된 OLE 객체가 활성화 될 수 있게 하였으며, 사용자의 마우스 입력을 통해 OLE 객체들의 크기나 상태를 변경할 수 있도록 기존의 이벤트 처리 함수들이 변경되었다.

새로 추가된 COODCntrlItem 클래스는 OODesigner에 삽입되거나 연결되는 OLE 객체를 관리하기 위한 목적을 갖는다. 이 클래스의 기능은 삽입된 OLE 객체의 위치와 크기를 관리하고 필요한 경우 디바이스 컨텍스트에 그 객체를 그리거나 객체를 활성화하는 것이다. 그림 3의 COODCntrlItem 내용은 전체 구현 내역 중에서 핵심적인 멤버들만을 제시한 것으로 대문자로 시작되는 함수 이름들은 MFC에서 제공되는 오버라이드된 함수들이고 나머지 멤버들의 의미는 멤버 이름을 통하여 그 역할을 이해할 수 있다.

마지막으로 이루어진 변경사항에는 OLE 컨테이너 기능의 추가를 위하여 메인 클래스인 COODApp에 OLE 라이브러리를 초기화시키기 위한 코드 추가가 있었으며, 프레임 관련 클래스인 CMainFrame 클래스에 OLE 객체를 위한 메뉴 기능의 추가가 있었다.

현재까지 기술된 바와 같이 OLE 기능 추가 과정에서 기존에 존재하던 200 여개의 클래스 중 적은 수의 클래스들에 대해서만 수정이 필요하였는데 이는 기존에 구현된 클래스들이 MVC 패러다임 관점에서 역할 분담이 잘 분산되도록 설계되었던 점에 기인한다.

3. 시스템 구현 결과

OODesigner에 OLE 컨테이너 기능을 구현함으로써 OODesigner는 오디오, 비디오 등을 이용한 문서화가 가능해졌으며 아울러 Window에서 제공되는 그림판, 엑셀, 그래픽 편집기 등의 객체를 UML 문서에 연결할 수 있게 되었다. 따라서 OODesigner는 음성, 동영상, 그림, 워드 문서를 편집할 수 있는 소프트웨어와 함께 실행되면서 소프트웨어 개발자에게 막강한 문서화 환경을 제공할 수 있다. 본 절에서는 OLE 컨테이너 기능의 구현 결과를 사용 예와 실행화면 중심으로 제시한다.

OLE 컨테이너 기능을 통해 제공된 기능 중의 하나는 그림 5에서와 같이 UML 문서에 그림 파일을 삽입하는 것이다. 그림 5는 Animal 클래스를 설계하면서 Animal 객체의 예가 어떠한 것이 있는지를 알기 쉽게 보여주고 있다. 그림 5의 화면을 봄으로써 Animal 클래스를 설계한 설계자가 어떠한 객체들을 위해 모델링을 수행하였는가를 쉽게 알 수 있다.

그림 6의 내용은 그림 5의 문서에 삽입되어 있는 그림 객체를 제자리 활성화(in-place activation) 방식으로 활성화하는 상황을 보여 준다. OODesigner의 사용자는 그림 객체 위에서 마우스를 더블 클릭함으로써 제자리 활성화를 수행할 수 있다. 이 그림에서 볼 수 있다시피 제자리 활성화가 이루어지고 난 다음에 도구 프레임의 메뉴나 툴바 상태바 등이 그림

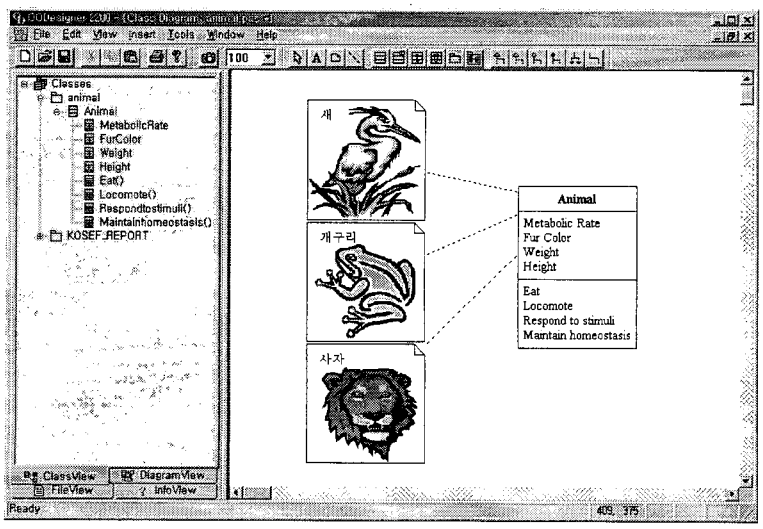


그림 5. UML 문서에 그림 객체들이 삽입된 예

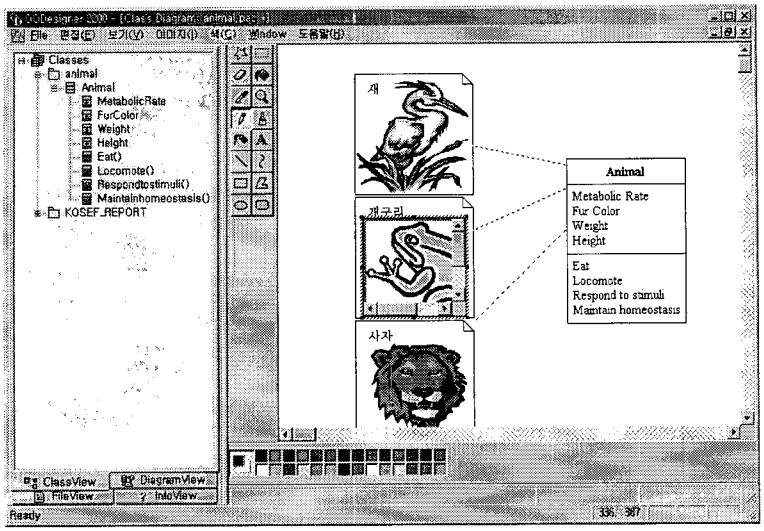


그림 6. 그림 객체의 제자리 활성화 예

판을 위한 것으로 교체되었다.

그림 7의 경우는 오픈 방식에 의한 OLE 객체 활성화 상황을 보여준다. 오픈 방식으로 객체를 활성화하기 위해서는 삽입된 OLE 객체가 제공하는 <open> 메뉴를 이용한다. 혹은 삽입된 객체가 원본 도큐먼트와 연결(link)되어 있는 경우에는 객체 위에서 더블 클릭을 하는 경우에 무조건 오픈 방식으로 활성화된다. 그림 7의 경우에 호랑이 그림은 원본 그림 파일과 연결되어 있는 경우의 예이며 화면에서와 같이 호랑이의 눈 부분을 그림판을 통해 수정하면 OODesigner의 뷰 상에서 수정이 이루어짐과 동시에 원본 파일이 변경됨을 확인 할 수 있었다.

OLE 컨테이너 기능을 통해 제공된 다른 기능은 녹음기나 엑셀과 같은 도구를 이용하여 작성된 객체를 삽입하는 것이다. 그림 8의 화면에는 두 가지의 OLE 객체가 삽입되어 있는데 그 하나는 우측 상단에 스피커 형태의 아이콘으로 표시된 음성 객체이며 다른 하나는 우측하단에 있는 엑셀로 작성된 그래프 객체이다.

그림 9의 화면은 녹음기로 작성된 보이스 도큐먼

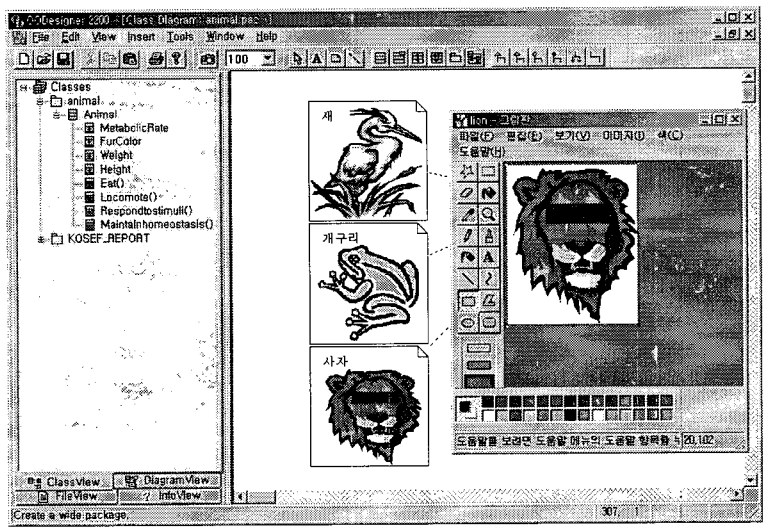


그림 7. 그림 객체의 오픈 방식 활성화 예

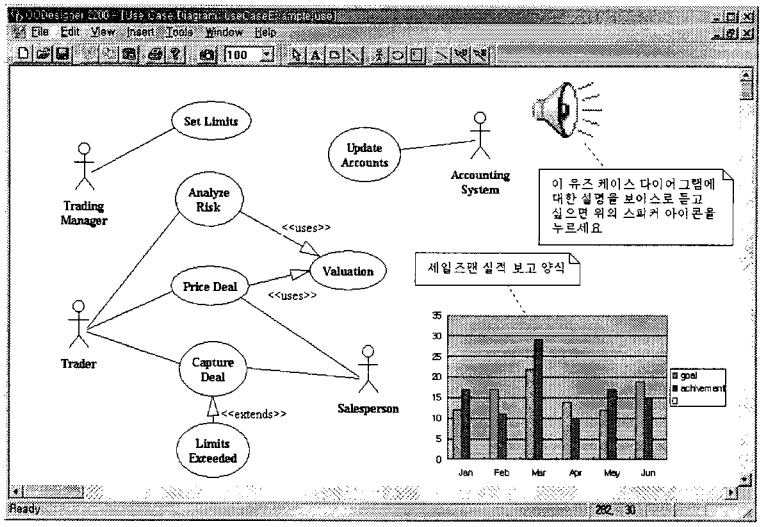


그림 8. UML 문서에 삽입된 음성 객체와 엑셀 객체의 예

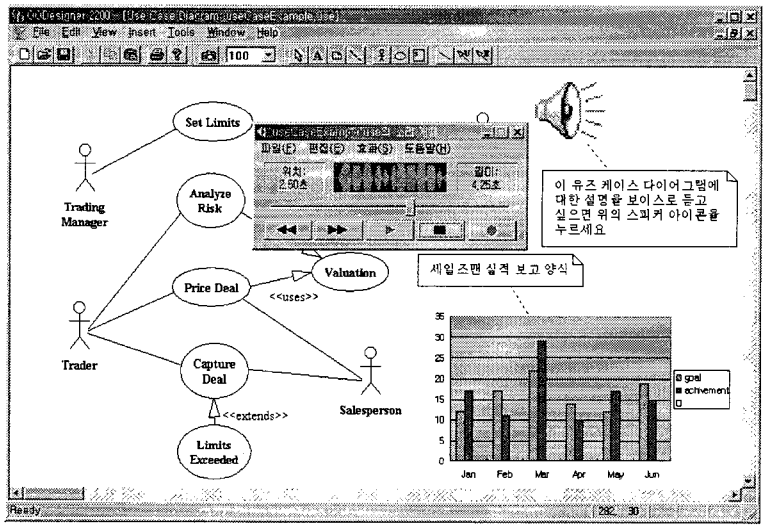


그림 9. 음성을 이용한 문서화의 예

트가 실행되는 상황을 보여준다. 현재 대부분의 CASE 도구가 텍스트 기반의 문서화 기능만을 제공하고 있는 점에서 볼 때 음성을 이용한 문서화는 소프트웨어 문서의 이해력과 사용용이성을 매우 높여줄 수 있을 것이다.

그림 10의 화면은 엑셀로 작성된 그래프 객체를 활성화하는 상황을 보여준다. 이 화면에서 활성화된 엑셀은 그래프를 그리기 위한 데이터 쉬트의 편집 상황을 보여준다. 화면에서와 같이 OODesigner가 기존에 Window에서 제공되는 도구들과 연동되어 실행될 수 있기 때문에 문서 작성의 편의성과 일관성을 달성할 수 있다.

마지막으로 제시된 그림 11의 화면은 캠코더로 작성된 비디오 도큐먼트의 활성화 상태를 보여준다. 현재 멀티미디어와 관계된 기술 수준이 컴퓨터와 관련된 모든 응용 분야에서 엄청난 파급 효과를 미치고 있는 점에서 볼 때 그림 11의 화면에서와 같이 비디오 영상을 이용한 설계 문서의 작성 환경은 소프트웨어 개발 환경에 상당히 긍정적인 영향을 미칠 것이다.

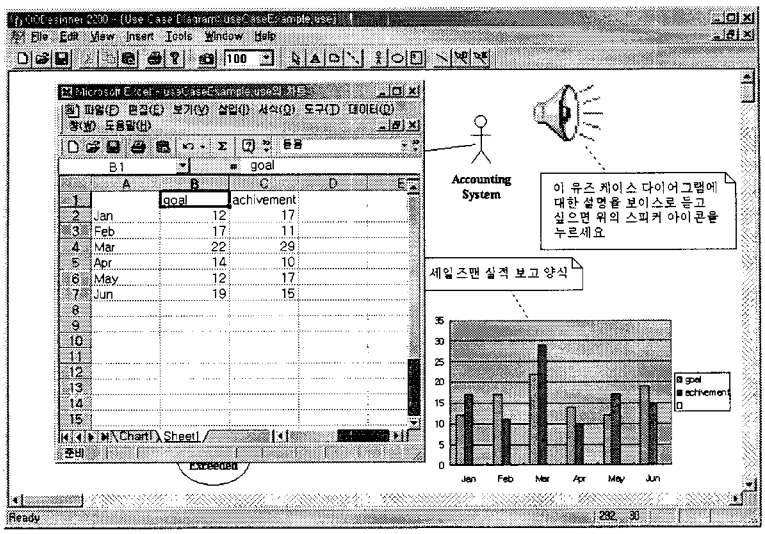


그림 10. 엑셀 객체의 활성화를 통한 엑셀 문서 편집 예

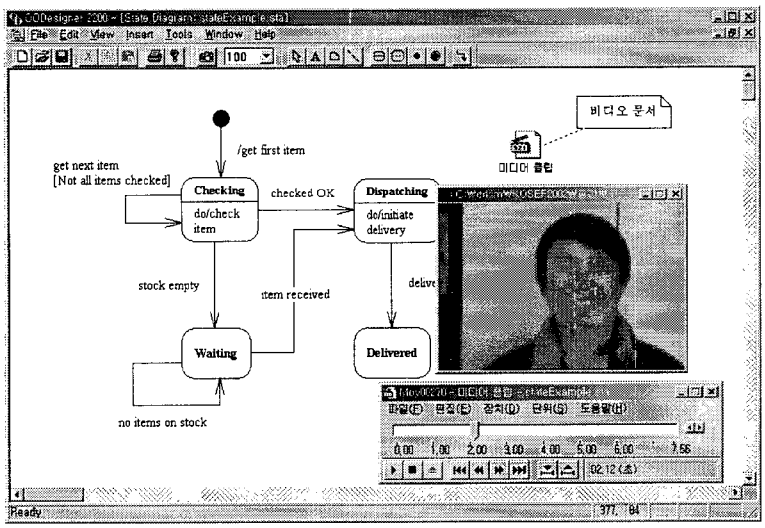


그림 11. 비디오 문서 활성화의 예

4. 결 론

재사용성 확보를 위한 복합 문서 지원 기술이 보편화되어 있음에도 불구하고 현재 사용되고 있는 대부분의 CASE 도구에는 이러한 기술을 지원하고 있지 못하다. CASE 도구에 컴포넌트 기술을 도입하여 복합 문서 지원 기능을 적용하게되면 소프트웨어 개발시의 문서화 과정에서 많은 이득을 얻을 수 있다. 예를 들어 멀티미디어 저작 도구와 연계된 문서화를 수행할 수 있으며 문서와 UML 설계 다이어그램의 일관성을 확보하는 것도 용이하다. 본 논문에서는 기존에 개발되었던 CASE 도구 OODesigner에 복합 문서 지원 기능을 적용하기 위한 목적으로 OLE 컨테이너 기능을 추가로 개발한 연구 내용을 기술하였다. 본 논문에서 논의된 주된 내용은 OLE 기능의 추가를 위한 설계 결과와 도구의 유용성을 보여주는 구현 결과에 대한 것이다. 현재 OODesigner에 대한 테스팅을 수행 중이며 시스템의 안정성이 확보되면 UML 모델링을 위한 유용한 도구로서의 역할을 할 수 있을 것이다. 본 연구의 이후에 수행되어야 할 향후 연구 내용으로는 OODesigner에 OLE 서버 기능을 구현하는 것이다. 서버 기능을 구현하게되면

OODesigner가 OLE 컨테이너 기능을 제공하는 다른 소프트웨어에 삽입되어 작동될 수 있기 때문에 좀더 유용한 객체 지향 설계 문서 환경을 제공할 것으로 보인다. 서버 기능 구현 전의 OODesigner는 도큐먼트 구조가 불안정하기 때문에 도구가 공개되지 못하고 있으나 서버 구현이 이루어지고 난 이후의 OODesigner는 인터넷에 공개될 예정이다.

참 고 문 헌

[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, R. E. Johnson, "Surveying Current Research in Object-Oriented Design," Communications of the ACM, 33(9), pp. 104-124, September 1990.

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

[5] http://www.rational.com

[6] http://www.toghethersoft.com

[7] 조장우, 김태균, "객체지향 CASE 도구에 대한 재구조화 실험", 한국정보처리학회 논문지, 제6권, 제4호, pp.932-940, 1999.

[8] 홍의석, 김태균, "객체 지향 CASE 도구 OODesigner의 플랫폼 이식 사례 연구," 한국정보처리학회 논문지, 제7권, 제9호, pp. 2857-2866, 2000.

[9] Richard Grimes, et al., *Beginning ATL COM Programming*, Wrox, 1999.

[10] E. Roman, *Mastering Enterprise Java Beans*, John Wiley & Sons, 1999.

[11] T. J. Mowbray, R. Zahavi. *The Essential CORBA*, John Willey & Sons, 1995.

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

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

최 길 림

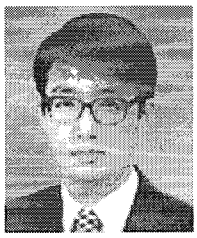
1989년 동아대학교 공과대학 전산공학과 공학사

1994년 부산수산대학교 산업대학원 전산정보학과 이학석사

2003년 부산외국어대학교 컴퓨터전자공학부 박사과정 수료

1996년~현재 경남정보대학 컴퓨터정보계열 부교수

관심분야 : 객체지향 소프트웨어공학, CASE도구, 인터넷프로그래밍



김 태 균

1985년 서울대학교 전산학과 이학사

1987년 서울대학교 전산학과 이학석사

1995년 서울대학교 전산학과 이학박사

1988년~현재 부산외국어대학교 컴퓨터공학과 교수

관심분야 : 소프트웨어공학, 객체지향소프트웨어공학, CASE도구