

ODA에 근거한 문서 클래스 에디터 설계 및 구현

正會員 鄭 會 京*, 正會員 李 壽 淵*

Implementation and Design of Document Class Editor
based on ODAHoe Kyung Jung*, Soo Youn Lee* *Regular Members*

요 약

본 논문은 이 기종 문서처리 시스템간에 문서교환을 위해 국제 표준으로 제정된 ODA에 따른 문서 클래스(class) 에디터 설계 및 구현에 대하여 기술하였다.

ODA에서처럼 문서구조를 공통 논리구조와 배치구조로 분리하여 처리하였으며, 문서 프로파일을 작성할 수 있도록 설계하였다. 문서가 정확하게 작성되었는지를 객체(object) 단위로 확인할 수 있는 유틸리티(utility)를 구현하였다. 또한 그 문서의 ODIF 스트림(stream) 데이터가 정확한지를 확인하였다.

본 에디터는 국제 문서 응용 프로파일(DAP : Document Application Profile)인 DAP 단계 2의 제안에 따라 설계하였으며, UNIX 운영체제의 SUN 워크스테이션상에서 이식성이 좋고 일관된 사용자 인터페이스(interface)를 제공하는 X 윈도우 및 Motif 환경하에서 구현하였다. 본 연구를 통하여 구현된 에디터는 특정 문서구조를 갖는 실제 ODA 문서를 작성시 이용될 수 있다.

ABSTRACT

This paper describes an implementation of the document class editor based on ODA(Open Document Architecture).

For processing, we divided document structure into generic logical structure and generic layout structure as ODA standard. Also this editor could edit document profile. Using the utility which was implemented to investigate the composed document by object, we confirmed the document. And we could verify the ODIF stream data of the document.

We designed this editor based on DAP level 2 of international functional standard. This system was implemented in environment of the X window system and the Motif as graphical user interface. This document class editor will be used to create real document having specific document structure.

*光云大學校 電子計算機工學科
Dept. of Computer Science, Kwangwoon University
論文番號 : 92-140(接受1992. 6. 10)

I. 서 론

서로 다른 문서처리 시스템들 사이 효율적인 문서 교환이 최근들어 많이 요구되고 있다. 문자정보 뿐만 아니라 도형 및 이미지 정보를 포함한 문서는 이 기간의 원활한 정보교환을 위해 문서의 표준화된 교환 형식이 절실히 요구되고 있다.

이러한 필요에 따라 OSI 환경하에서 이 기간의 효율적인 문서교환을 위해 문서 처리 모델로 나타나는 것이 ODA로 이는 문서를 표현하는 추상적 모델이고, 문서교환 형식인 ODIF(Open Document Interchange format)는 ODA문서를 전송하기 위한 데이터 포맷이다.^(1, 2, 3) ODA는 정보처리 사무용 시스템에서 문서교환의 필요성에 대한 인식을 같이하는 ECMA(European Computer Manufactures Association), ISO와 CCITT가 공동 연구한 결과로 국제 표준으로 발표되었다.

그러나 ODA는 문서로 표현가능한 모든 구조와 표현정보를 기술하고 있으므로 너무 광범위하여 실제 구현에는 많은 어려움이 있기 때문에, PAGODA(Profile Alignment Group on ODA)를 중심으로 문서 응용 프로파일을 단계별로 제한하여 속성과 구조를 표현하는 3단계의 기능 표준을 정의하고 있다.⁵⁾ 이렇게 정의된 DAP들을 구현하는 방법으로는 첫째 변환기 형태가 있다.⁴⁾ 이는 기존에 있는 각종 워드프로세서등의 편집기 화일을 ODIF 데이터 스트림 형태로 변환하는 것으로 지금까지의 처리계를 사용할 수 있고 비교적 쉬운 장점을 가지는 반면 완전한 변환이 어려운 단점을 가진다. 둘째는 순수 ODA 에디터를 ODA 구조에 기초해 설계함으로 ODA 특성을 활용 가능하지만 새로운 처리계가 필요한 단점을 가진다. 셋째로는 ODA 툴킷으로서 변환기와 순수 ODA편집기의 효율적인 개발에 도움을 준다. 실제로 이러한 분류에 따른 개발 예로는 미국의 NSF(National Science Foundation), CMU(Carnegie Mellon University) 및 UM(University of Michigan) 등을 중심으로 수행중인 EXPRES 프로젝트에서, 유럽의 ESPRIT 프로젝트내 PODA(Piloting of ODA) 프로젝트에서 변환기 및 툴킷을 개발했고, 애플사의 WOPODA 변환기용 툴킷등을 들 수 있다. 이에 반해 국내에서는 몇몇 연구소와 일부 대학에서 ODA에 대해 연구를 하고 있으나 아직 미흡한 실정이다.

본 논문에서는 순수한 ODA 환경 구축시 요구되는

문서 클래스 구조 즉 공통 논리구조와 배치구조를 생성할 수 있는 문서 클래스 에디터를 설계 및 구현하였다. 본 에디터 출력은 문서의 동일한 성질을 갖는 공통 논리구조와 배치구조를 생성하게 되는데 이 공통구조를 이용하여 특정 논리구조와 배치구조를 갖는 실제 문서를 작성할 수 있게 된다. 또한 생성된 공통구조는 모든 문서에 따른 일반적인 클래스 정보를 전송시 피함으로써 전송효율을 증대시키며 문서 변형시 내부적인 일정성을 유지하고 특정 문서를 편집시 용이성을 제공한다. 구현은 UNIX 환경하에서 X 윈도우와 Motif를 사용하여 사용자 인터페이스를 구현함으로써 시스템의 이식성을 추구하고 사용자에게 친숙하고 일관된 사용자 인터페이스를 형성하였다.^{7, 8, 9)} 구현한 문서 클래스 에디터는 DAP단계 2인 FOD 26에 기본하여, 논리적 측면에서 텍스트, 이미지 및 도형 그래픽 정보를 가지며, 배치적 측면에서 주석, 두문, 미문 및 멀티컬럼 등이 가능하게 구현하였다.^{4, 13)} 작성된 문서 구조 결과를 객체 단위로 확인할 수 있도록 유틸리티를 구현하여 검토하였다. 또한 본 연구실에서 설계한 ODIF변환기에 의해 변환된 ODIF스트림 데이터로도 확인됨을 보였다.

2장에서는 ODA의 기본개념 및 공통구조를 기술하는 구조자에 대해 설명하고, 3장에 문서 클래스 에디터를, 4장에 실험 및 고찰 그리고 5장에 결론을 맺었다.

II. ODA/ ODIF 기본 개념

본장에서는 ODA/ODIF의 기본 개념에 대해 기술한다.^{1, 10, 11)}

ODA/ODIF는 이 기간에 문자, 라스터(raster) 및 도형 그래픽등의 멀티미디어 정보를 포함하는 문서의 상호 교환을 실현하기 위한 규격으로 문서구조와 내용구조를 규정한 ODA와 ODA에 따른 문서를 다른 시스템간에 교환할 때의 형식을 규정한 ODIF로 구성된다. ODA 문서구조 기본개념은 그림 1과 같이 논리적/배치적 관점에 따른 2개의 구조로 표현된다.

논리구조는 문서를 객체의 의미에 기본한 계층구조로 수정 가능한 문서를 표현하는데 사용되고, 배치구조는 문서 객체를 물리적 구성에 기본한 계층구조로 포맷된 문서를 표현하는데 사용된다.

또한 ODA에서는 문서의 공통 특성을 기술하는 논리적/배치적 객체들의 생성규칙 정보를 취급하는데,

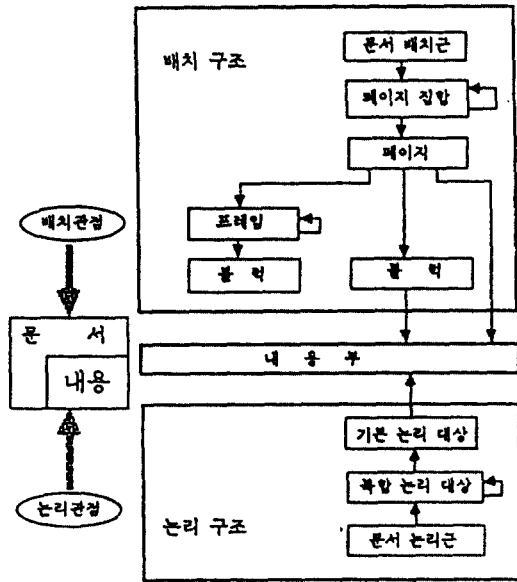


그림 1. 논리구조와 배치구조
Fig. 1. Logical structure and Layout structure.

이들의 생성규칙을 각각 공통 논리구조, 공통 배치구조라 한다. 이들 공통구조에 의해 생성되는 실제 문서의 특정 구조를 각각 특정 논리구조와 특정 배치구조라 한다.

객체들과 객체 클래스의 배치 또는 표현에 대한 정보는 스타일(style)이라는 구성요소로 지정되는데 배치 스타일과 표현 스타일로 나타낸다. 배치 스타일은 배치처리에서 논리객체가 배치객체로 변환되어 배치되는 규칙을 제공하며, 표현 스타일은 내용부의 가시화(imaging)와 배치를 위해 논리객체와 배치객체에 적용되며 이는 내용구조에 따라 각각 규정된다.

문서에는 하나의 문서 프로파일(profile)을 가지는데 이는 문서에 관계되는 모든 정보를 유지 관리한다.

ODA의 내용구조는 내용부 기본 배치객체에 관련된 최하위 구성 요소로서 그래픽 요소, 제어함수, 내용과 관련된 부호화 방법, 표현속성으로 구성되며, 배치객체에 있는 그래픽요소를 위치지정하는 규칙정의와 수정가능한 형식이 어떻게 포맷된 형식으로 변환되는가를 기술한다. 현재에는 문자, 도형 그래픽 및 라스터 그래픽 내용구조를 정의하고 있고, 앞으로는 음성, 애니메이션(animation) 등의 시변요소를 갖

는 동적 미디어와 수치데이터 등의 처리미디어를 통합적으로 표현하는 구조로 연구 진행중이다.¹²⁾

ODA의 문서처리 모델은 편집처리, 배치처리 및 가시화처리로 구성된다. 편집처리는 문서내용 편집과 공통 논리구조의 참조로 특정 논리구조 편집을 하며 결과는 배치처리에 입력된다. 배치처리는 공통 배치구조의 참조로 문서내용이 배치될 특정 배치구조를 생성하게 되며, 결과는 가시화처리로 입력된다. 가시화처리는 표현 스타일의 표현속성을 적용하여 기본 배치객체의 표현속성과 배치내용에 삽입된 제어함수의 조절에 의해 표현 매체상에 문서를 만들어 내는 것이다.

ODIF는 ODA의 구조화 문서를 교환하기 위해 사용하는 데이터 스트림의 형식을 정의하고, 이들을 코드화하기 위해 ASN.1에 의해 기술된 데이터 형식이다.⁶⁾ 또한 ODIF는 구조화된 문서의 객체, 내용 및 스타일의 부호화를 정의하여 교환될 문서에서 나타낼 수 있는 속성들 집합을 정의한다.

ODA에서는 복합객체의 구성을 위한 구성순서의 명세를 설명 가능하다. 이는 한 복합객체가 그들의 구성요소들에 의해 어떻게 구성되는지, 또는 어떤 선택자들이 구성요소들을 접근하기 위해 사용될 지를 정의하는 구조자에 의해 이루어진다. 본 시스템에서는 이들 구조자들이 문서 클래스 에디터에서 공통 논리/배치구조 생성시에 생성된다. 구조자(constructor)의 3가지 종류는 다음과 같다.

- SEQ : 객체의 구성요소들에 대한 순차적 순서를 설명하며, 구성 요소들은 같은 종류이고 순차적으로 접근될 수 있다.
- CHO : 구성요소들의 한 집합에서 단지 하나가 선택되고 접근될 수 있다. 구성요소들은 같은 또는 다른 종류일 수 있다.
- AGG : 구성요소들 사이 특정 순서없이 나타낼 수 있다.

III. 문서 클래스 에디터

본 연구의 문서 클래스 에디터는 공통 논리구조 에디터, 공통 배치구조 에디터 및 프로파일 에디터로 구성되며, 이의 전체적인 구성도를 그림 2에 보여주고 있다.

공통 논리/배치구조 에디터는 각각 논리/배치 구조의 생성규칙 화일에 의해 각각 공통 논리/배치 구

조 화일을 생성한다. 모든 ODA문서에는 그 문서의 고유 특성을 기술한 하나의 프로화일을 갖게 되는데, 이는 프로화일 에디터에 의해 생성된다. 이때 당연값을 갖는 속성은 시스템 내부에서 생성되고 사용자에 의해 생성될 속성은 여기서 편집된다. 이렇게 생성된 공통구조 화일은 유틸리티에 의해 사용자가 직접 확인하고 검토할 수 있도록 본 에디터에 구현하여 제공한다.

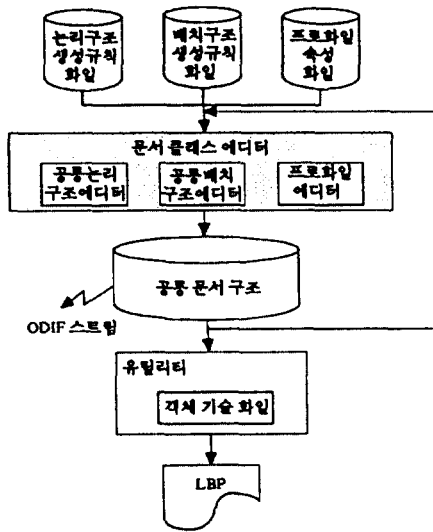


그림 2. 시스템 전체 구성도
Fig. 2. Block diagram of system.

1. 프로화일 에디터

프로화일은 문서를 처리하기 위해 필요로 하는 모든 정보를 문서의 해당 속성값에 의해 제공한다. 이들 속성값에는 공통 논리구조 및 공통 배치구조, 특정 논리구조 및 특정 배치구조, 배치 스타일, 표현 스타일, 문서구조 클래스, 내용구조 클래스, ODA 버전 및 ODIF 형식등을 포함하는데 이들 값을 에디터에 의해 편집 가능하게 하였다.

2. 공통 논리구조 에디터

논리구조는 그림 3에 나타난 것처럼 DAP단계 2에서 제안한 논리구조 생성규칙에 따라 구성한다.

문서 논리근은 문서 논리구조에서 최상위 단계로 하나이상의 패세지(passage)로 구성됨을 보이며, 각 패세지는 선택적으로 두가지 제한중 한가지로 구성

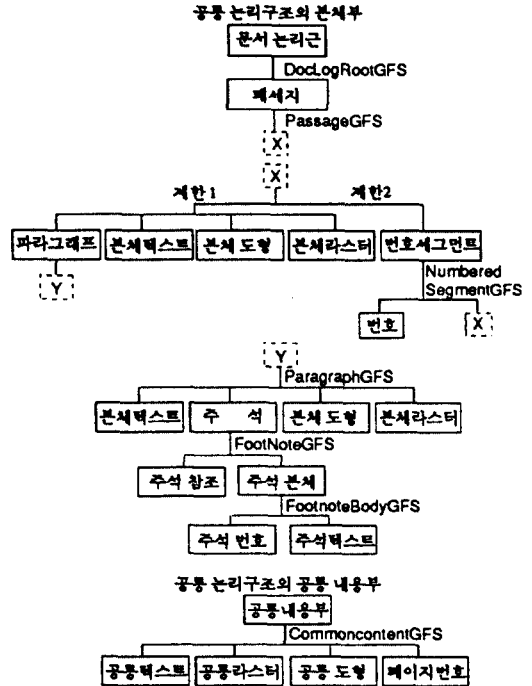


그림 3. 공통 논리구조 생성규칙
Fig. 3. Creation rule of generic logical structure.

된다. 제한 1은 파라그래프(paragraph), 도형본체, 라스터본체, 텍스트본체 중 하나이상으로 구성되며, 제한 2는 번호세그먼트가 오게된다. 파라그래프는 다시 하나이상의 본체텍스트, 주석, 본체도형 및 본체라스터로 구성됨을 보이고 있다. 공통 내용부는 일반적으로 두문 이나 미문부에 쓰이는 공통텍스트, 공통라스터 및 공통도형으로 구성된다.

그림 3의 규칙에 의해 공통 논리구조로 생성하기 위한 에디터의 공통 논리구조 구성을 그림 4에 보였다.

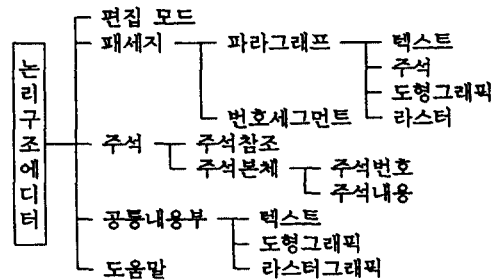


그림 4. 에디터의 공통 논리구조 구성
Fig. 4. Generic logical structure organization of editor.

에디터에 의해 논리구조 생성시 사용자가 각 단계 별로 논리구조 객체들을 선택하면 각 객체에 해당하는 클래스 객체를 생성하여 객체 속성값들을 정하게 된다. 각 클래스 객체들의 자료구조는 ODA에서의 객체에 대한 속성값과 트리운행을 위해 다음 클래스 객체 식별자 및 저장시의 접근 사슬(chain)값이 추가로 구성된다. 이때 하위 구조 생성식은 먼저 구조식 소개자(SEQ, AGG, CHO) 중 하나가 선택되고, 다음에 구조식을 이루는 항(REQ, OPT, REP, OPT_REP) 중 하나가 은 뒤 클래스 구조에서 참조되는 객체 식별값이 따르고 구조식의 종료자(ECE)로서 마치게 된다. 예를들어 파라그래프항에서 사용자가 본체텍스트와 본체라스터를 선택하면 구조식은 다음과 같이 생성된다.

```

X_SEQ :
X_REP :
X_CHO :
        X_REQ : 본체텍스트 식별자 ;
        X_REQ : 본체라스터 식별자 ;
X_ECE :
X_ECE :
    
```

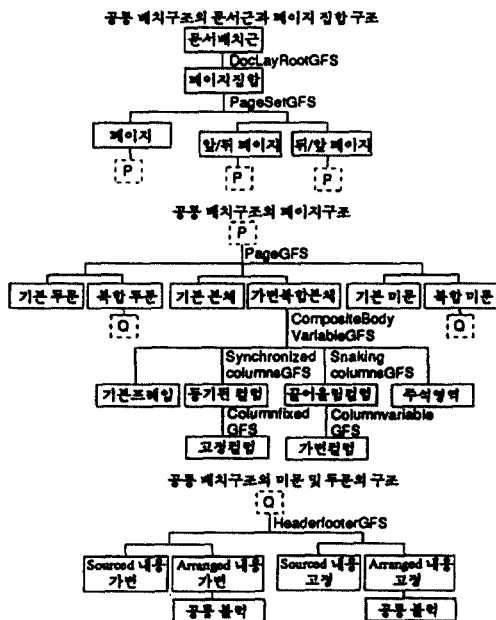


그림 5. 공통 배치구조 생성규칙
Fig. 5. Creation rule of generic layout structure.

3. 공통 배치구조 에디터

배치구조는 그림 5에 나타난 것처럼 DAP단계 2에서 제안한 배치구조 생성규칙에 따라 구성한다.

그림 5에서 페이지 집합은 페이지들을 식별하기 위한 것이고, 페이지는 문서 내용부의 위치 및 표현 형식을 정의하기 위한 직사각형 영역으로 표제 페이지, 단순 페이지 및 앞면 페이지와 뒷면 페이지 등이 있다. 각 페이지들은 두문과 본체부분과 미문이 있고, 본체부분은 다시 기본 본체와 복합 본체부로 나뉘어 구성됨을 보이고 있다. 그림 5에서 제안되고 있는 생성규칙에 따라 공통 배치구조를 생성하기 위한 에디터의 공통 배치구조 구성을 그림 6에서 보였다.

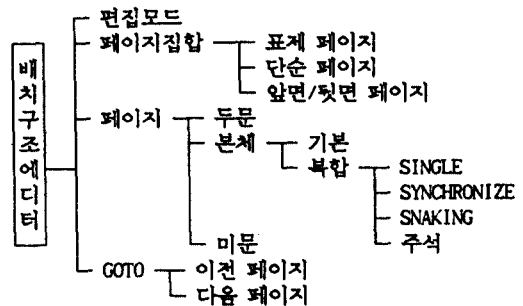


그림 6. 에디터의 공통 배치구조 구성
Fig. 6. Generic layout structure organization of editor.

에디터에 의한 공통 배치구조의 생성은 앞에서 설명한 논리구조 생성과 동일하다. 이에 대한 예로서 페이지항에서의 사용자가 두문, 기본본체, 미문을 선택하면 다음과 같은 구조식이 생성된다.

```

X_SEQ :
X_REQ : 두문 식별자 ;
X_REQ : 기본본체 식별자 ;
X_REQ : 미문 식별자 ;
X_ECE :
    
```

IV. 문서 클래스 작성 및 고찰

1. 문서 클래스 작성

본 문서 클래스 에디터는 Sun 3/80 워크스테이션에서 NCD-16 그래픽 터미널을 Ethernet으로 연결하여 사용하였다. 본 시스템의 운영체제는 UNIX

BSD 4.3이며, X 윈도우 시스템 및 Motif환경에서 프로그램 언어는 C언어를 사용하였다. 문서 클래스 에디터에 의해 프로파일, 공통 논리구조와 공통 배치구조를 작성하였으며, 작성된 구조는 IS 8613에서 정의된 ODA 구조로 저장되며, 이 구조가 작성자의 의도대로 생성되었는지 검토하기 위해 논리구조와 배치구조의 ODA 구조를 객체단위로 출력할 수 있는 유틸리티를 두어 확인하였다. 작성된 구조를 하드카피(hardcopy)하기 위한 출력장치로는 레이저빔 프린터 QMS-PS810을 이용하여 출력하였다. 문서 클래스 에디터에 의해 공통구조를 생성하는 과정을 다음에 보인다.

한 공통 문서를 만들면 문서당 하나의 프로파일을 편집해야 한다. 이에 대한 편집을 그림 7에 보이고 있다. 여기에서 당연값들은 시스템 자체적으로 값을 부여하도록 하였으며, 사용자에게 의해 편집될 사항들을 편집할 수 있도록 하였다. 문서의 공통 논리구조와 배치구조는 각각을 사용자가 만들고자 하는 공통 구조를 에디터에서 선택하면 선택된 항목을 직사각형 모양으로 화면에 보여주며, 내부적으로 공통구조와 구조식이 생성된다. 그림 8에서는 문서논리관 밑에 번호세그먼트와 파라그래프가 오고 파라그래프 밑에 본체텍스트, 본체도형, 본체라스터 및 주석으로 구성되는 공통 논리구조 작성 예를 보이고 있다. 그림 9는 단순 페이지에 두문, 멀티컬럼, 주석 및 비문으로 구성되는 공통 배치구조 작성 예를 보인다. 그리고 본 에디터에서는 생성한 공통구조를 확인하기 위한 유틸리티를 구현하였다. 이는 생성된 ODA 트리(tree) 구조를 운행하면서 각 객체가 가지는 속성들이 정확한 값으로 생성되었는지를 확인할 수 있도록 하고 있다. 이 유틸리티의 기능으로는 원하는 객체의 속성들을 볼 수 있거나, 배치구조/논리구조 각각의 구조 전체 객체들의 속성들을 볼 수 있거나, 생성된 구조에서 가지는 객체들을 통계적으로 분석한 결과를 볼 수 있는 기능이 있다. 이에 대한 예가 그림 10에 나타나고 있다. 그림 11은 앞에서 생성한 공통 문서구조를 확인하기 위해, 본 연구실에서 구현한 ODIF 변환기에 의해 ODIF 스트림으로 변환한 결과를 출력해 보여 주고 있다.

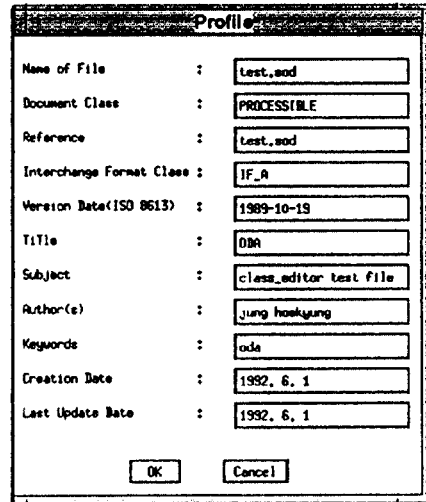


그림 7. 프로파일 작성에
Fig. 7. Compositd example of Profile.

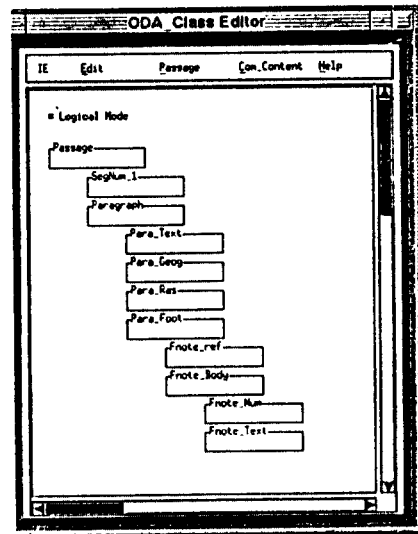


그림 8. 공통 논리구조 작성에
Fig. 8. Compositd example of generic logical structure.

OBJID	1	Profile	OBJID	44	Generic Frame
Objecttype	35	"Profile	Objecttype	"0 1 00"	"0 1 00"
Layout Class head	36	"A."	ODA identifier	"A."	"A."
Logical Class head	36	"A."	User comment	"SingleColumn"	"SingleColumn"
Application Profile	"1 3 14 2 6 0 1"	"numberstring-1"	Application comment	0	0
Default graphic subrep	0	0	Subordinates count	0	0
Default graphic char sets:	"(0..J..)"	0	Balance count	0	0
Default Content Architecture Class	NOT DEFINED (-1)	0	Position:	0	0
Document Architecture Class	FORMATTED PROCESSABLE	0	Leading offset:	0	0
Content Architecture Classes:	CP	0	Loading Separation:	0	0
	PFRC	0	Alignment	UNDEFINED (-1)	UNDEFINED (-1)
	CG	0	Fill order	NORMAL	NORMAL
Interchange Format	8	0	Dimensions:	FIXED, value 9240	FIXED, value 9240
ODA version	"ISO 8613"	0	Horizontal	RULE 8, max -1, min -1	RULE 8, max -1, min -1
ODA version date	"1989-10-19"	0	Vertical:	0	0
Title:	"ODA"	0	OBJID	41	Generic Frame
Subject:	"class_editor test file"	0	Objecttype	"0 3 00"	"0 3 00"
Keywords:	"oda"	0	ODA identifier	"A."	"A."
Document Reference	is a Character String,	0	User comment	"SynchronizedColumn"	"SynchronizedColumn"
Reference type	"test.sod"	0	Application comment	0	0
reference	"1992. 6. 1"	0	Subordinates count	0	0
Document date and time:	"1992. 6. 1"	0	Balance count	0	0
Creation date and time:	"1992. 6. 1"	0	Position:	0	0
Revision History:		0	Leading offset:	0	0
Authors	Surname: "jung hoekyung "	0	Loading Separation:	0	0
Graphic char subrep:	8	0	Alignment	UNDEFINED (-1)	UNDEFINED (-1)
Graphic Character Sets	"(0..J..)"	0	Fill order	NORMAL	NORMAL
Layout Classes:		0	Dimensions:	FIXED, value 9240	FIXED, value 9240
OBJID	55	0	Horizontal	RULE 8, max -1, min -1	RULE 8, max -1, min -1
Objecttype	Generic Layout Root	0	Vertical:	0	0
ODA identifier	"0"	0	OBJID	40	Generic Frame
Application comment	"DocLayout"	0	Objecttype	"0 3 00"	"0 3 00"
Generator for Subordinates	SEQUENCE;	0	ODA identifier	"A."	"A."
SEQUENCE;	REQUIRED; (52)	0	User comment	"ShakingColumn"	"ShakingColumn"
ECE;		0	Application comment	0	0
		0	Subordinates count	0	0
		0	Balance count	0	0
		0	Position:	0	0
		0	Leading offset:	0	0
		0	Loading Separation:	0	0
		0	Alignment	UNDEFINED (-1)	UNDEFINED (-1)
		0	Fill order	NORMAL	NORMAL
		0	Dimensions:	FIXED, value 9240	FIXED, value 9240
		0	Horizontal	RULE 8, max -1, min -1	RULE 8, max -1, min -1
		0	Vertical:	0	0
		0	OBJID	39	Generic Frame
		0	Objecttype	"0 3 4 00"	"0 3 4 00"
		0	ODA identifier	"A."	"A."
		0	User comment	"Footnote"	"Footnote"
		0	Application comment	0	0
		0	Subordinates count	0	0
		0	Balance count	0	0
		0	Position:	0	0
		0	Leading offset:	0	0
		0	Loading Separation:	0	0
		0	Alignment	UNDEFINED (-1)	UNDEFINED (-1)
		0	Fill order	REVERSE	REVERSE
		0	Dimensions:	FIXED, value 9240	FIXED, value 9240
		0	Horizontal	RULE 8, max -1, min -1	RULE 8, max -1, min -1
		0	Vertical:	0	0
		0	Logical Classes:		
		0	OBJID	26	Generic Logical Root
		0	Objecttype	"2"	"2"
		0	ODA identifier	"DocLogRoot"	"DocLogRoot"
		0	Application comment		
		0	Generator for Subordinates	SEQUENCE;	SEQUENCE;
		0	SEQUENCE;	REPEAT; (33)	REPEAT; (33)
		0	ECE;		
		0	OBJID	13	Generic Basic Logical Object
		0	Objecttype	"2 17 00"	"2 17 00"
		0	ODA identifier	"FootnoteNumber"	"FootnoteNumber"
		0	Application comment		
		0	Content Architecture Class	Character	Character
		0	Content generator expression:	UpperRoman Binding Superior "footenumber"	UpperRoman Binding Superior "footenumber"
		0	Content portion count	0	0
		0	OBJID	33	Generic Composite Logical Object
		0	Objecttype	"2 2 1"	"2 2 1"
		0	ODA identifier	"Passage"	"Passage"
		0	Application comment		
		0	Generator for Subordinates	SEQUENCE;	SEQUENCE;
		0	SEQUENCE;	REPEAT; (29)	REPEAT; (29)
		0	ECE;		
		0	Binding name	"number-1"	"number-1"
		0	Binding value:	Cardinal Number -1	Cardinal Number -1
		0	Cardinal Number	-1	-1
		0	Binding name	"footenumber"	"footenumber"
		0	Binding value:	Cardinal Number 0	Cardinal Number 0
		0	Cardinal Number	0	0
		0	Binding name	"separator-1"	"separator-1"
		0	Binding value:	String ""	String ""
		0	String	""	""
		0	Binding name	"prefix-1"	"prefix-1"
		0	Binding value:	String ""	String ""
		0	String	""	""
		0	Binding name	"suffix-1"	"suffix-1"
		0	Binding value:	String ""	String ""
		0	String	""	""
		0	OBJID	47	Generic Frame
		0	Objecttype	"0 6 0 00"	"0 6 0 00"
		0	ODA identifier	"A."	"A."
		0	User comment	"S CompositeReader"	"S CompositeReader"
		0	Application comment	0	0
		0	Subordinates count	0	0
		0	Balance count	0	0
		0	Position:	0	0
		0	Horizontal:	0	0
		0	Vertical:	0	0
		0	Dimensions:	FIXED, value 9240	FIXED, value 9240
		0	Horizontal:	FIXED, value 0	FIXED, value 0
		0	Vertical:	0	0
		0	Medium:	14030	14030
		0	Height	9920	9920
		0	Width	0	0
		0	Side	0	0
		0	OBJID	45	Generic Frame
		0	Objecttype	"0 3 00"	"0 3 00"
		0	ODA identifier	"A."	"A."
		0	User comment	"Compositebody"	"Compositebody"
		0	Application comment		
		0	Generator for Subordinates	SEQUENCE;	SEQUENCE;
		0	SEQUENCE;	REPEAT;	REPEAT;
		0	CHOICE;		
		0	REQUIRED; (44)		
		0	REQUIRED; (41)		
		0	REQUIRED; (40)		
		0	REQUIRED; (39)		
		0	ECE;		
		0	Subordinates count	0	0
		0	Balance count	0	0
		0	Position:	0	0
		0	Leading offset:	0	0
		0	Trailing offset:	0	0
		0	Alignment	UNDEFINED (-1)	UNDEFINED (-1)
		0	Fill order	UNDEFINED (-1)	UNDEFINED (-1)
		0	Dimensions:	FIXED, value 9240	FIXED, value 9240
		0	Horizontal	RULE 8, max 13200, min -1	RULE 8, max 13200, min -1
		0	Vertical:	0	0

그림 10. 생성된 문서구조의 객체단위 출력

Fig. 10. Output in object of composited document structure.

V. 결 론

본 논문은 ODA / ODIF의 기초기술 연구 목적으로 순수 ODA에디터 설계시 필요한 문서 클래스 에디터를 설계 및 구현하였다. 그래픽 사용자 인터페이스를 향상시키기 위해 X 윈도우와 Motif를 이용하여 표준 응용프로그래밍 환경 구축과 시스템의 이식성 및 사용자에게 친숙하고 일관된 사용자 인터페이스를 형성하였다. 에디터 구현 단계는 국제 기능표준인 DAP 단계 2에서 제안한 속성을 모두 수용하여 논리적 측면에서 내용 정보는 텍스트, 도형 및 라스터 그래픽 정보를 표현 가능하도록 하고 주석 참조도 가능하며, 배치적 측면에서 멀티컬럼, 주석, 미문과 두문 이 가능하게 설계되었다.

본 에디터에 의해 생성된 문서구조인 공통 논리구조와 배치구조가 국제 표준으로 제시된 IS 8613의 구조와 일치되기 때문에 ODA에서 본래 추구하는 이종간의 문서교환 즉 ODIF 스트림으로의 변환은 본 연구실에서 설계한 ODIF 변환기로 변환이 가능했다. 또한 에디터에 의해 생성된 공통 논리/배치구조를 객체 단위로 출력하여 정확히 생성되었음을 확인할 수 있었으며, 하드카피는 LBP로 출력하였다. 본 에디터에서 작성된 공통 문서구조는 ODIF 변환기의 시험용 데이터로도 사용될 수 있으며, 에디터 결과인 공통 문서구조를 이용해서 특정 문서구조를 생성시에 이용되리라 본다. 이를 이용한 실제 문서를 작성할 수 있는 특정 문서구조 에디터의 개발과 포매팅 프로세싱 처리를 위한 포맷터와 ODIF스트림 부호기와 복호기 등의 계속적인 연구로 ODA 환경구축에 크게 기대된다.

참 고 문 헌

1. ISO 8613, "Information processing - Text and Office System - Office Document Architecture (ODA) and Interchange Format (ODIF)", Sept. 1989.
2. R. Hunter, P. Kaijser and F. Nielsen, "ODA: A Document Architecture for Open Systems", Computer comm., pp.69~79, April 1989.
3. W. Horak, "Office Document Architecture and Office Document Interchange Formats: Current status of international standardization", pp.50~60, Computer, Oct. 1985.
4. Seyed N. Golkar, Peter T. Kirstein and Afzal Montaser-Kohsari, "ODA activities at university college London", Computer networks and ISDN system, 21, pp.187~196, 1991.
5. ISO /IEC JTC1 /SGFS, "pDISP 10610 - Office Document Format FOD 26 - Enhanced document structure - character, raster graphics and geometric graphics content architecture", April 1991.
6. ISO 8824, "ASN.1: Abstract Syntax Notation one and basic coding rules", Dec. 1987.
7. Oliver Jones, "Introduction to the X window system", Prentice Hall, 1989.
8. Jim Gettys, Ron Newman, Robert W. Scheifler, "Xlib-C language X interface Protocol version 11", MIT, 1987.
9. Open software foundation, "OSF/Motif programmer's guide", Eleven Carribrige center, 1989.
10. 정회경, 이수연, "ODA/ODIF", 정보과학회지, 제 8권, 제6호, pp.49-55, Dec. 1990.
11. Nobour Sonehrra, "Present and future of office document architecture", Japan IE87-96.
12. Ute Bormann and Carsten Borman, "Standards for open document processing: current state and future developments", Computer networks and ISDN system 21, pp.149~163, 1991.
13. Andy Spicely, "ODA profile: application and development," Computer networks and ISDN system 21, pp.165~173, 1991.



鄭會京(Hoe Kyung Jung) 정회원
1961년 1월 28일생
1985년 2월: 광운대학교 전자계산
기공학과 졸업(공학
사)
1987년 2월: 광운대학교 전자계산
기공학과 졸업(공학석
사)

1989년~현재: 광운대학교 전자계산기공학과 박사과정 재
학중

1990년 1월~1990년 7월: 영국 ICL 연수

※주요관심분야: ODA, SGML, Hypermedia, GUI



李壽淵(Soo Youn Lee) 正會員
1969년: 광운대학교 전자통신공학
과(공학사)
1977년: 연세대학교 전자공학과(공
학석사)
1983년: 일본 교토대학교 정보공학
과(공학박사)
1973년 3월~현재: 광운대학교 전
자계산기공학과 교수

※주요관심분야: ODA, SGML, GUI, HCI 하이퍼미디어
문서처리