

OSI 환경에서 문서 화일링 및 검색 시스템의 설계 및 구현

(Design and Implementation of Document Filing and Retrieval System in an OSI Environment)

林宰弘, 朴容震

(Jae Hong Yim and Yong Jin Park)

要約

본 논문에서는 국제 표준화 기구(ISO: International Standards Organization)에서 표준화 작업을 하고 있는 분산 사무 응용시스템 모델(DOAM: Distributed Office Application Model)의 한 응용 분야인 문서 화일링 및 검색(DFR: Document Filing and Retrieval) 시스템의 설계 및 구현에 대하여 논한다.

국제 표준안을 기초로 하여 SUN 워크스테이션과 PC/386 상에서 C 언어로 DFR 시스템을 구현하였으며, 클라이언트와 서버간의 동작 시험시 연계 기술자(association descriptor)와 서비스 요소들의 프리미티브를 추적함으로써 구현 타당성을 검증하였다. 본 연구의 결과는 국제 표준안을 기초로 한 DFR 시스템의 구현 가능성을 보였으며, DFR 시스템에 대한 기능표준을 제정하는데 크게 기여할 것이다.

Abstract

This paper describes a design and implementation of the DFR(Document Filing and Retrieval) system, one of applications of DOAM(Distributed Office Application Model) which is the international standard in ISO(International Standards Organization). On the basis of the international standard, the DFR system is implemented on SUN workstation and PC/386 with C language, and its implementation is verified by tracing the association descriptor and primitives of service elements when its operation is tested between client and server. The result of this study shows that the DFR system can be implemented on the basis of the international standard, and makes a contribution toward the establishment of functional standards for the DFR system.

1. 서론

분산 사무 환경은 PC, 워크스테이션 및 다른 여러 정보기기들을 통신망으로 연결하여 분산된 자원의 효

율적 공유를 가능케 한다. 따라서 분산 사무 환경에서는 PC와 워크스테이션들의 통신망 구축에 의한 정보의 공유 및 시스템의 효율적인 사무 처리능이 제 공되어야 하는데, 이러한 분산 사무 환경에서 사용자가 사무 정보를 효율적이고 편리하게 처리하기 위해서는 다음과 같은 기능들의 지원이 필요로 한다.

- 데이터 전송 기능
- 개인 상호간의 메시지교환 기능
- 사용자 그룹간의 통신(group communication)

*正會員, 漢陽大學校 電子工學科

(Dept. of Elec. Eng., Hanyang Univ.)

※ 본 논문은 한국전자통신연구소 위탁과제 연구결과임.

接受日字 : 1993年 5月 19日

기능

- 서버(server)에의 원격 로그인(remote login)

기능

- 디렉토리 서비스 기능
- 문서의 화일링과 검색 기능 등

이러한 요구에 따라 국제 표준화 기구(ISO: International Standards Organization)에서는 분산 사무 처리에 필요한 클라이언트/서버 모델의 여러 응용시스템들을 통합 처리할 수 있는 분산 사무 응용시스템 모델(DOAM: Distributed Office Application Model)을 제안하고 이에 대한 표준화 작업을 진행하고 있다. 분산 사무 응용시스템 모델에서는 클라이언트와 서버간의 통신 프로토콜인 접근(access) 프로토콜과 서버와 서버간의 통신 프로토콜인 시스템 프로토콜을 정의하고 있으며, 응용 분야로써 메시지 처리 시스템(MHS: Message Handling System), 디렉토리 시스템(directory system), 문서 화일링 및 검색(DFR: Document Filing and Retrieval) 시스템^{13, 14}, 문서 인쇄 응용(DPA: Document Printing Application)^{5, 6} 등을 규정하고 있다.^{1, 2, 7}

DFR 시스템은 디스크 용량이 제한된 PC나 워크스테이션에서 원격 동작(remote operation)으로 대용량의 서버에 화일링되어 있는 문서를 관리하는 응용 시스템으로써, 문서 내용에 무관한 통신 프로토콜을 규정하여 일반 텍스트 문서나 멀티미디어 문서 또는 개방 문서 구조(ODA: Open Document Architecture) 표준안에 따른 문서를 속성을 이용하여 화일링하거나 검색, 수정, 삭제 등의 동작을 수행한다. 즉 메시지 처리 시스템이 OSI(Open System Interconnection) 계층 구조를 따르는 전자 우편 시스템의 국제 표준안으로 제안되었듯이 DFR 시스템은 일종의 분산 화일 시스템에 대한 국제 표준이라 할 수 있으나, 화일 저장소(file store)의 구조에 대한 표준화보다는 원격 동작 서비스 요소의 표준화에 주안점을 두고 있다.³ DFR 시스템에 관한 구현 동향으로는 일본 Fuji Xerox 사에서 연구 개발을 위한 실현검토를 하고 있으며⁸, 아시아-태평양지역 OSI 워크샵(AOW: Asian-Oceania OSI Workshop)에서는 DFR 시스템 기능 표준을 제정하기 위한 그룹을 결성하였으나¹⁰, 국내에서는 아직 미비한 실정이다.

DFR 시스템의 서버 역할을 하는 문서 저장기(DS: Document Store)의 확장성과 계층적 화일링 기능을 제공하기 위하여 UNIX 화일 시스템을 이용하고, 클라이언트와 서버의 원격 동작 서비스 요소간의 통신을 지원하기 위하여 ISODE 소프트웨어를 이

용해서 SUN 워크스테이션과 PC/386 상에서 C 언어로 DFR 시스템을 구현하여 시험하였다. 본 연구의 결과는 국제 표준안을 기초로 한 DFR 시스템의 구현 가능성을 보였으며, DFR 시스템에 대한 기능 표준을 제정하는데 크게 기여할 것이다.

본 논문의 II장에서는 DOAM과 DFR 시스템의 개요를 설명하고, III장에서는 DFR 시스템의 설계에 대하여 기술한다. IV장에서는 DFR 시스템의 구현과 동작 시험 및 고찰을, 마지막으로 V장에서는 결론을 서술한다.

II. DOAM과 DFR 시스템의 개요

1. DOAM

DOAM의 목적은 여러 분산 사무 응용시스템들을 상호 접속하기 위한 전체적 구성 및 설계원리를 정의하는 것이다. DOAM에서는 클라이언트/서버 모델을 이용하여 응용 시스템을 나타내는데, 클라이언트는 사용자와의 인터페이스를 담당하며 사용자에게서 받는 서비스 요청을 서버에게 전달해 주는 역할을 하고, 서버는 그 서비스를 수행하여 결과를 클라이언트에 전달해 준다.^{11, 7}

그림 1에 나타난 바와 같이 분산 응용 시스템에서는 클라이언트와 서버가 다른 시스템에 존재하므로 클라이언트가 서버에 접근하기 위해서는 접근 프로토콜이 필요하게 되며 ISO에서는 이러한 접근 프로토콜에 대하여 표준화 작업을 진행하고 있다. 그러나 사용자와 클라이언트 사이의 인터페이스에 대해서는 표준화 작업을 하지 않고 시스템 구현자의 선택에 따라 임의적으로 설계, 구현토록 하고 있다. 그림 1에서 x는 임의의 응용 시스템을 나타낸다.

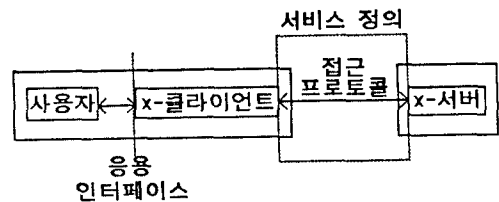


그림 1. 분산 응용 시스템

Fig. 1. Distributed application system.

그림 2는 분산 사무 응용시스템의 기능 모델을 나타낸 것이다. 그림 2에서 보면 알 수 있듯이 클라이언트와 서버는 접근 프로토콜을 통해서 통신을 하고, 서버와 서버는 시스템 프로토콜에 의해서 통신을 하

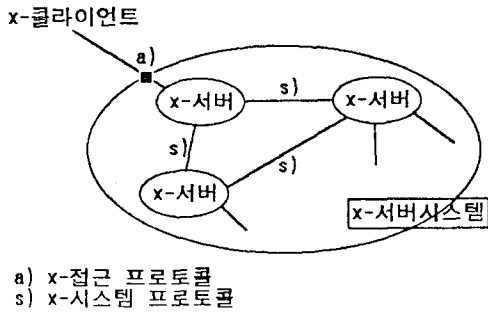


그림 2. 분산 사무 응용시스템의 기능 모델
Fig. 2. Functional model of distributed office application.

게된다. 그림 2의 모델을 메시지 처리 시스템 모델에 대응시켜 보면 메시지 전송 처리기(MTA: Message Transfer Agent)는 'x-서버'에 해당하고 원격 사용자 처리기(RUA: Remote User Agent)는 'x-클라이언트'에 대응이 되며^[14], 반면에 디렉토리 시스템 모델에 대응시켜 보면 디렉토리 시스템 처리기(DSA: Directory System Agent)가 'x-서버'에 디렉토리 사용자 처리기(DUA: Directory User Agent)는 'x-클라이언트'에 각각 해당된다.^[15] 이와 같이 모든 분산 응용 시스템을 위의 모델에 따라 대응시킬 수가 있으며, 분산 사무 응용시스템 모델을 OSI(Open Systems Interconnection) 계층 구조에 대응시킨다면 클라이언트와 서버는 모두 응용 계층에 해당된다.

2. DFR 시스템

DFR 시스템은 디스크 용량이 제한된 PC나 워크스테이션에서 원격 동작으로 대용량의 서버에 화일링 되어 있는 문서를 관리하는 응용 시스템으로써, 디스크 용량이 제한된 사용자가 디스크 용량이 큰 호스트의 시스템을 이용하는 DOAM의 대표적인 경우이다. DFR 시스템은 여러가지 속성을 이용하여 사용자가 작성한 문서를 검색하거나 관리하고, 사용허가를 받지 못한 사용자에게는 서비스를 제공하지 않는다.

DFR 시스템의 기본환경은 DOAM에 따라 통신망으로 연결된 DFR-서버(DFR-Server)노드와 DFR-사용자(DFR-User)노드로 구성되며, 클라이언트 기능을 하는 DFR-사용자는 DFR-포트(DFR-Port)를 통해 DFR-서버에 동작을 요청한다. 그러므로 DFR-포트는 DFR-서버가 제공하는 서비스를 정의하여 두 노드간의 인터페이스를 제공하게 된다.^[16] 그림 3은 DFR 시스템의 추상적 모델을 나타낸 것이다.

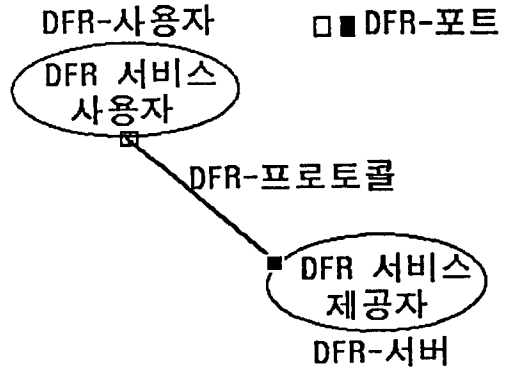


그림 3. DFR 시스템의 추상적 모델
Fig. 3. Abstract model of DFR system.

DFR 시스템은 사용자에게 서비스를 제공하기 위하여 하나의 서버안에 하나의 문서 저장기(DS: Document Store)를 두고 있는데, 그 구조는 그림 4와 같다.

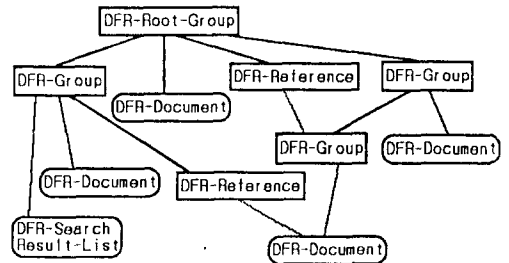


그림 4. DFR 문서 저장기의 구조
Fig. 4. DFR Document Store structure.

그림 4에서 보는 바와 같이 문서 저장기를 이루는 DFR-객체(DFR-Object)는 트리(tree) 구조로 이루어져 있다. 각 DFR-객체는 속성부(attribute)와 내용부(content)로 구성되고, DFR-객체를 고유하게 지정하는데 사용되는 고유 식별자(UPI: Unique Permanent Identifier)를 가지고 있으며, 그 종류는 다음과 같다.

- 루트 그룹(root group) : 문서 저장기의 최상위에 위치하며 모든 DFR-객체를 가리키고 있는 포인터의 집합으로 이루어져 있다.
- 일정 그룹(proper group) : 루트 그룹내에서 여러 DFR-객체를 포함하는 단위로서, 일정 그룹 자체의 정보를 나타내는 속성부와 그룹내에 포함되어 있는 DFR-객체들의 리스트를 저장하는 내용부로 이루어

어진다. 일정 그룹에 접근할 수 있는 사용자들을 제한함으로써 그룹내의 DFR-객체는 사용허가를 받지 못한 사용자로부터 보호받을 수 있다.

- 참조(reference) : 참조는 문서 저장기내의 루트 그룹과 또 다른 참조를 제외한 DFR-객체를 가리키는 포인터로서, DFR-객체를 링크시켜서 그에 대한 정보를 저장하고 있다.

- 탐색 결과 목록(search result list) : DFR 동작중의 하나인 탐색(search) 동작의 결과를 저장하는 DFR-객체이다. 그러므로 자주 탐색되는 DFR-객체인 경우 탐색 동작을 반복할 필요가 없다.

- 문서(document) : 실제 문서의 데이터를 저장하는 곳이며, 사용자는 문서를 에디터로 작성한 후 문서 저장기의 최하위의 위치에 저장한다.

Ⅲ. DFR 시스템의 설계

DFR 시스템은 분산 환경에서 문서의 화일링 및 검색 기능에 관련된 다양한 기능을 제공하며, ISO DFR 표준안에 명시된 주요 기능을 요약하면 다음과 같다.

- 분산 사무 응용시스템 환경에서 다수의 사용자가 이용 가능

- DFR-객체를 그룹화하여 화일링하고 계층적으로 관리

- 문서의 내용과 독립적인 속성을 부가
- 속성을 이용한 DFR-객체의 탐색, 열람 기능 등

DFR 표준안에는 위의 기능과 DFR 동작에 대한 개요만을 명시하고 있으며, DFR-객체의 구조를 추상 구분 표기(ASN.1: Abstract Syntax Notation One)^[11] 형식으로 기술하고 있다. 따라서 실제 DFR 시스템을 설계, 구현하기 위해서는 다음과 같은 사항을 고려하여야 한다.

- 사용하기에 편리한 사용자 인터페이스

- DFR 서비스 요소(DFRSE: DFR Service Element)와 원격 동작 서비스 요소(ROSE: Remote Operations Service Element), 연계 제어 서비스 요소(ACSE: Association Control Service Element)와의 인터페이스

- 서버 시스템내에서 DFR-객체를 처리, 관리하기 위한 루틴

- 서버에 대한 접근 인증 방법
- DFR-객체에 대한 접근 권리의 관리 등

위의 사항을 고려하여 DFR 시스템을 문서 저장기, 서버와 클라이언트 모듈별로 나누어 설계하였으며, 그 내용은 다음과 같다.

1. DFR 시스템 구성 모델

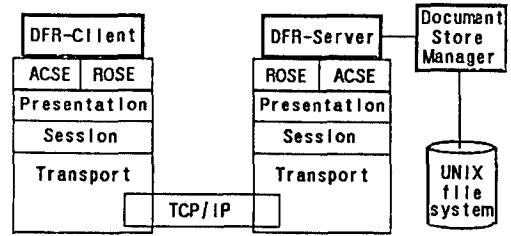


그림 5. DFR 시스템 구성 모델

Fig. 5. DFR system configuration model.

DFR 시스템 구성 모델은 그림 5와 같다. DFR 동작이 OSI 환경에서 제공되기 위해서는 DFR-클라이언트와 DFR-서버의 각 응용 개체(AE: Application Entity)에 연결된 포트상에 응용 서비스 요소(ASE: Application Service Element)가 제공되어야 하며, DFR 서비스 요소(DFRSE)는 두 응용 개체간의 인터페이스를 제공하는 포트를 지원하게 된다. ROSE^[12]는 분산 개방 환경에서 응용 시스템간의 상호동작을 지원해주기 위한 것으로 DFR 동작들은 ROSE에 의해 제공되는 추상 서비스(abstract service)로 매핑되어 동작한다. 그리고 ACSE^[13]는 원거리에 있는 응용 개체와의 연계(association)의 설정과 해제를 담당하게 된다. DFR 동작과 ACSE, ROSE와의 인터페이스를 위하여 구성된 매핑 관계를 표 1에 나타내었다.

2. 문서 저장기

DFR-객체를 화일링하는 문서 저장기의 구조는 다음 세 가지의 경우가 가능하다.

(1) 평탄 구조

DFR-객체중에서 일정 그룹을 제외한 객체, 즉 문서, 참조 등이 직접 루트 그룹에 포함되는 형태이며 일정 그룹이 없는 경우를 말한다. 이 구조는 구현하기는 쉬우나 확장성이 없으며 사용자가 문서를 분류 저장할 수 없다.

(2) 고정 계층 구조

미리 일정 그룹의 구조가 정해져 있어 이에 따라 DFR-객체들이 저장되고 사용자는 문서를 생성, 수정, 삭제할 수 있으나 일정 그룹을 생성, 수정할 수 없는 경우를 말한다. 문제점으로는 일정 그룹을 만들수 없기 때문에 확장성에 문제가 있어 사용에 제한이 따른다.

(3) 임의 계층 구조

DFR 문서 저장기내에 모든 DFR-객체, 즉 문서, 참조, 일정 그룹을 생성, 수정, 삭제할 수 있다. 따라

표 1. DFR 동작과 응용 서비스 요소와의 매핑 관계

Table 1. Mapping relation between DFR-operations and ASEs.

분류	동작명	기능	응용 서비스명	응용 요소
연계의 설정과 해제	bind	응용 기체의 연계 설정	A_Associate	ACSE
	unbind	응용 기체의 연계 해제	A_Release	
관리	create delete copy move reserve	DFR-객체의 생성 DFR-객체의 삭제 DFR-객체의 복사 DFR-객체의 이동 DFR-객체의 예약및 해제	RO_Invoke	ROSE
	검색과 수정	read modify list	DFR-객체의 읽기 DFR-객체의 수정 DFR-객체의 속성 리스트 작성	
동작 정지		abandon	이전의 동작 정지	

서 확장성이 매우 뛰어나며 응용분야가 많다.

위에서 설명한 세 가지 방법중에서 본 연구의 문서 저장기로는 임의 계층 구조를 선택하였으며, 문서 저장기의 확장성과 계층적 화일링 기능을 위하여 UNIX 화일 시스템을 이용하였다. DFR-객체의 특성을 고려한 UNIX 화일 시스템과의 매핑 관계를 표 2에 나타내었다.

표 2. DFR-객체와 UNIX 화일 시스템의 매핑 관계

Table 2. Mapping relation between DFR-Objects and UNIX file system.

DFR-객체	UNIX 화일 시스템
DFR-그룹	디렉토리
DFR-문서	일반 화일
DFR-탐색 결과 목록	
DFR-참조	심볼 링크

문서 저장기에 화일링되는 DFR-객체는 속성부(attribute)와 내용부(content)로 이루어지며, 그 구조체를 그림 6과 같이 정의하였다.

```

struct DfrEntry {
    struct DfrEntryAttributes * attributes;
    struct DfrObjectContent * content;
};
    
```

```

struct DfrEntryAttributes {
    struct DfrAttribute * Attribute;
    struct DfrEntryAttributes * next;
};

struct DfrObjectContent {
    int offset;
#define DfrObjectContent_document_content 1
#define DfrObjectContent_root_group_content 2
#define DfrObjectContent_proper_group_content 3
#define DfrObjectContent_reference_content 4
    union {
        struct DfrGroupContent * root_group_content;
        struct DfrGroupContent * proper_group_content;
        struct DfrReferenceContent * reference_content;
        struct DfrDocumentContent * document_content;
    } un;
};
    
```

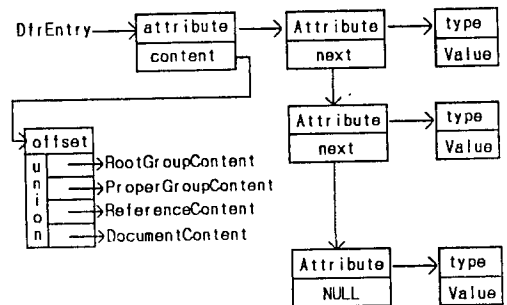


그림 6. DfrEntry 구조체
Fig. 6. DfrEntry structure.

그림 6에서 보듯이 내용부는 객체의 종류에 따라 달라지므로 메모리의 효율적인 이용을 위하여 union 구조체로 정의하였다. 속성부에 있어서 고유 식별자(UPD)는 UNIX 화일 시스템의 inode를 이용하고, 경로명, 소유주명, 생성 시간 등 UNIX 화일 시스템에서의 디렉토리 및 화일 구조체에서 제공되는 속성들은 그대로 이용하며, 그 이외의 속성은 별도의 화일에 저장하게 하였다.

3. DFR 클라이언트와 서버

1) 클라이언트

클라이언트는 사용자와의 인터페이스를 담당하며 사용자에게서 받는 서비스 요청을 서버에게 전달해 주는 역할을 한다. 본 연구에서는 그림 7과 같이 DFR-클라이언트를 구성하였으며, 그 기능은 다음과 같다.

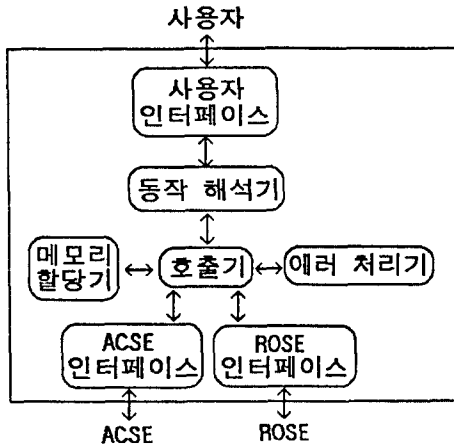


그림 7. DFR-클라이언트 구성
Fig. 7. DFR-Client configuration.

- 사용자 인터페이스: 사용자와 DFR 시스템과의 인터페이스 기능을 제공하며, 현재는 대화형(interactive) 방식으로 구현하였으나 X 윈도우를 이용한 그래픽 사용자 인터페이스(GUI: Graphic User Interface) 방식으로 개발중에 있다.
- 동작 해석기: 사용자가 요청한 동작을 해석하여 호출기에게 알려준다.
- 호출기(invoker): 사용자가 요청한 동작을 서버에 호출하며, 서버로부터 받은 결과를 사용자에게 통보한다.
- 메모리 할당기: 원격 동작에 사용되는 인수(argument)에 대한 메모리를 할당해 준다. 메모리의 효율적인 이용을 위하여 수행시 할당(run-time allocation) 방식을 채택하였다.
- 에러 처리기: 원격 동작에 대한 서버로부터의 에

러를 처리한다.

- ACSE, ROSE 인터페이스: DFR 서비스 요소와 ACSE, ROSE와의 인터페이스를 담당한다.
- 2) 서버

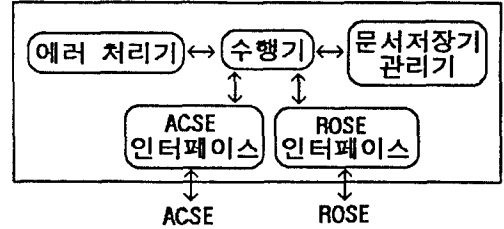


그림 8. DFR-서버의 구성
Fig. 8. DFR-Server configuration.

서버는 클라이언트로부터 요청받은 서비스를 수행하여 그 결과를 알려주는 역할을 한다. 그림 8은 DFR-서버의 구성을 나타내며, 그 기능은 다음과 같다.

- 수행기(performer): 클라이언트로 요청받은 동작을 수행하여 그 결과를 클라이언트로 넘겨준다.
- 문서 저장기 관리자: 문서 저장기에 화일링되는 DFR-객체를 관리하며, 이를 위하여 UNIX 화일 시스템과의 인터페이스 기능도 제공한다.

IV. DFR 시스템의 구현과 동작 시험

1. 구현 환경

DFR 시스템의 구현 환경은 서버에 SUN 3/50 SunOS release 4.1을, 클라이언트에 PC/386 SystemV/386 release 3.0을 사용하였으며, 서버와 클라이언트를 통신망으로 연결하기 위하여 PC/386 System V 상에 Excelan 사의 EXOS 205E 카드와 EXOS 8014-01 TCP/IP 네트워크 소프트웨어^[16, 17]를 이식하였다. 또한 ACSE, ROSE 및 그 하위 계층의 서비스들을 제공하기 위하여 ISODE 버전 6.0을 각각 사용하였다 [표 3 참조].

표 3. 구현 환경
Table 3. Implementation environment.

	서 비	클라이언트
응용 소프트웨어	ISODE-6.0	ISODE-6.0
운영체제	SunOS 4.1	System V/386 3.0
하드웨어	SUN 3/50	PC/386
네트워크 프로토콜	TCP/IP	

2. 문서 저장기

문서 저장기에 화일링되는 DFR-객체는 III장의 그림 6과 같이 속성부와 내용부로 이루어 지는데, 이를 UNIX 화일 시스템에 화일링하는 경우 i)속성부와 내용부를 하나의 화일에 저장하는 방법, ii)내용부와 속성부를 별개의 화일에 저장하는 방법, iii)UNIX 화일 시스템을 확장하여 UNIX 화일 속성을 수정하는 방법 세 가지가 가능하나, DFR 시스템의 이식성을 높게하기 위하여 iii)의 방법은 제외하고 i), ii)의 방법을 사용하였다. 즉 DFR-문서의 경우 접근 시간을 줄이기 위하여 i)의 방법을 택하고, DFR-그룹의 경우는 ii)의 방법을 사용하였다. 문서 저장기내에서 DFR-객체를 다루기 위하여 구현한 루틴은 표 4에 나타낸 바와 같다.

3. DFR-클라이언트와 서버

1) 고려 사항

(1) 원격 동작의 동기성

클라이언트와 서버간의 원격 동작은 동기성에 따라

동기적(synchronous) 동작 또는 비동기적(asynchronous) 동작으로 구현할 수 있는데, 동기적 동작은 동작을 요청한 후 서버로부터 수행 결과를 기다리는 방식이고, 비동기적 동작은 수행 결과를 기다리지 않고 새로운 동작을 요청하는 것이 가능한 방식이다. 따라서 비동기적 동작은 동시성(concurrency)이 가능한 반면에 동작의 동기화를 위해서 버퍼 기능이 제공되어야 한다. 본 연구에서는 DFR 시스템의 특성상 사용자는 요청한 결과를 즉시 받아야 하므로 원격 동작의 구현시 동기적 동작을 택하였다.

(2) 서버의 연계성

서버는 클라이언트와의 연계성에 따라 동적인(dynamic) 서버 또는 정적인(static) 서버로 구현할 수 있는데, 동적인 서버는 클라이언트로부터의 연계(association) 요청이 올 때마다 새로운 서버 프로세스가 생성되어 처리하는 것이고, 정적인 서버는 하나의 서버 프로세스가 여러 개의 연계에 관계하여 처리하는 방식이다. 동적인 서버에 있어서는 서버 프로세스가 하나의 연계에만 관계하여 서버의 구조가 간단

표 4. DFR-객체 처리 루틴
Table 4. DFR-Objects management routines.

함 수 명	기 능
struct DfrDocumentContent *str2ia5list()	문서의 내용부를 서버에 전달하기 위해 리스트 구조체를 만들어 주는 루틴
struct DfrDocumentContent *Input_content()	사용자가 명시하는 화일로부터 문서의 내용부를 읽어 들이는 루틴
void add2ent()	요청된 엔트리들 DfrEntryList 구조체에 맞도록 메모리를 할당하고 추가하는 루틴
struct DfrEntryList *ent2list()	추가된 엔트리들 리스트 구조체로 만들어주는 루틴
void add2att()	DfrEntryList에 포함될 속성들을 하나씩 추가하는 루틴
struct DfrAttribute *att2list()	속성을 속성 리스트 구조체로 만들어주는 루틴
struct DfrDocumentContent *Read_content()	문서 저장기내의 지정된 문서로부터 내용부를 읽어 들이는 루틴
struct DfrGroupContent *Read_group()	문서 저장기내의 지정된 그룹으로부터 내용부를 읽어 들이는 루틴
int GenTaskId()	작업 식별자(task-id)를 생성하는 루틴
struct DfrEntryAttributes *Read_attr()	지정된 DFR-객체로부터 속성을 읽어 들이는 루틴
void remove_attr()	지정된 DFR-객체에서 사용자 요청 속성을 삭제하는 루틴
void cp_attr_from()	지정된 DFR-객체의 속성을 사용자 요청 객체의 속성으로 복사하는 루틴
void cp_cont_from()	지정된 문서의 내용부를 사용자 요청 문서 객체의 내용부로 복사하는 루틴

해지므로 본 연구에서는 동적인 서버를 택하였다.

2) 클라이언트

DFR-클라이언트를 구성하는 각 모듈을 구현하기 위하여 다음과 같은 C 구조체를 정의하였다.

```

static struct dispatch {
char *op_name: /* operation name */
int op_code: /* operation code */
IFP do_operation: /* address of routine to
prepare argument */
IFP arg_free: /* address of routine to free
argument */
IFP op_result: /* address of routine for
result */
IFP op_error: /* address of routine for error */
char *op_help: /* operation description */
};

```

위의 구조체에서 'op_name'은 동작명, 'op_code'는 동작의 정수 코드값을 나타내며, 'do_operation'은 동작을 수행하는데 필요한 인수(argument)의 구조체에 맞도록 메모리를 할당해주고 값을 대응시켜 주는 함수를 나타낸다. 'arg_free'는 할당된 메모리를 다시 풀어주는 함수이며, 'op_result'와 'op_error'는 그 동작에 대해 서버에서 되돌려주는 결과나 에러를 처리하는 함수를 나타낸다. 'op_help'는 동작에 대한 도

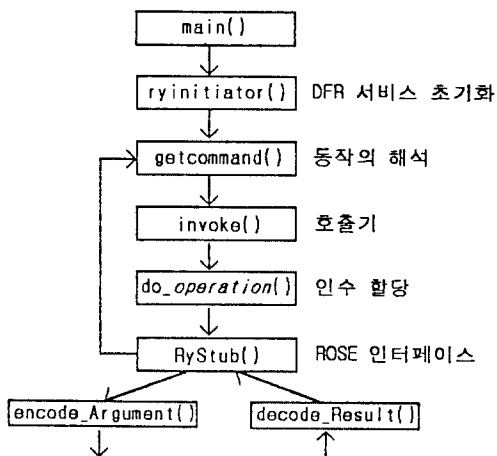


그림 9. DFR-클라이언트 동작의 흐름
Fig. 9. Operation flow of DFR-Client.

움말을 나타낸다.

그림 9는 클라이언트에서의 각 모듈의 흐름을 나타낸 것이다. 그림 9에서 사용자가 DFR을 초기화하면 ryinitiator 루틴이 서버와 연계 설정을 하고 getcommand 루틴이 사용자로부터 명령어를 받아, 그에 해당하는 동작 루틴을 호출하게 된다. RyStub 루틴은 각 동작의 인수를 인코딩 루틴을 통하여 서버에 넘겨주며, 그 결과를 서버로부터 받아 처리하여 준다.

표 5에는 본 연구에서 구현한 클라이언트 동작의 루틴과 그 기능을 나타내었다.

3) 서버

서버에는 클라이언트로부터의 요청을 수행키 위한 모듈의 구성을 위하여 다음과 같은 C 구조체를 정의하였다.

```

static struct dispatch {
char *op_name: /* operation name */
int op_code: /* operation code */
IFP op_operation: /* address of routine for
response */
};

```

위의 구조체에서 'op_name'과 'op_code'는 클라이언트에서의 의미와 같고, 'op_operation'은 요청 받은 동작 서비스를 수행해 주는 함수를 나타낸다.

그림 10은 서버 동작의 흐름을 나타낸 것이다. 클라이언트로부터 연계 설정 요청이 오면 ryresponder 루틴이 연계 설정을 하여주고, RyDispatch 루틴이 인수를 받아 요청 동작을 수행한 후 그 결과나 에러를 클라이언트에 넘겨준다.

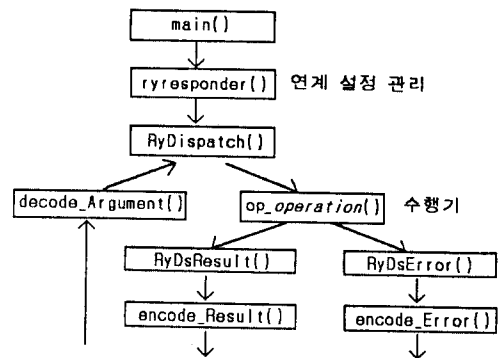


그림 10. DFR-서버 동작의 흐름
Fig. 10. Operation flow of DFR-Server.

표 5. DFR-클라이언트 루틴
Table 5. DFR-Client routines.

함수명	기능
do_bind()	클라이언트와 서버간의 연계 설정을 요청하는 루틴이며, 사용자의 이름과 패스워드를 인수로 받아 서버에 있는 문서 저장기에의 접근 여부를 인증하게 된다.
do_create()	문서 저장기에 새로운 DFR-객체를 생성 요청하는 루틴이며, DFR-객체의 종류와 속성, 내용을 인수로 받아 서버에 넘겨주게 된다.
do_delete()	DFR-객체를 삭제 요청하는 루틴이며, 삭제할 엔트리를 인수로 받아 서버에 넘겨주게 된다.
do_copy()	DFR-객체를 복사 요청하는 루틴이며, 복사할 엔트리와 복사되어질 엔트리를 인수로 받아 서버에 넘겨주게 된다. 복사할 DFR-객체가 일정 그룹일 경우 그룹에 속한 모든 객체가 복사된다.
do_move()	DFR-객체를 이동 요청하는 루틴이며, 이동되어질 엔트리를 인수로 받아 서버에 넘겨주며 복사 요청과 마찬가지로 일정 그룹일 경우 그룹에 속한 모든 객체가 이동된다.
do_read()	DFR-객체의 속성과 내용을 읽기 위한 루틴이며, 읽히질 DFR-객체의 종류와 엔트리, 속성 또는 내용의 선택을 인수로 받아 서버에 넘겨준다.
do_modify()	DFR-객체의 수정을 요청하는 루틴이며, 수정할 엔트리와 속성, 내용을 인수로 받아 서버에 넘겨준다.
do_list()	일정 그룹에 속한 DFR-객체의 속성 리스트를 요청하는 루틴으로써, 인수로써 일정 그룹의 엔트리를 인수로 받아 서버에 넘겨준다. 읽히질 DFR-객체의 수를 제한할 수도 있다.
do_reserve()	DFR-객체의 예약 레벨[3]을 수정 요청하는 루틴이며, 인수로써 예약할 엔트리와 예약 레벨을 인수로 받아 서버에 넘겨준다.
do_abandon()	사용자가 이전에 요청한 동작을 취소 요청하는 루틴이며, 인수로써 작업 식별자 또는 요청 식별자(invoke-id)를 받아 서버에 넘겨준다.
do_unbind()	연계 설정을 해제 요청하는 루틴이다.

표 6에는 본 연구에서 구현한 DFR-서버 루틴과 그 기능을 나타내었다.

4. 동작 시험 및 고찰

1) 동작 시험

본 연구에서 구현한 DFR 시스템의 소스 코드는 C 언어로 약 18,000 라인이며, 문서 저장기 관련 부분은 6,000 라인, 클라이언트 서버 부분은 12,000 라인이다.

DFR 시스템의 동작 시험을 위하여 문서 저장기의 루트 그룹으로는 "/tmp/DS" 디렉토리를 사용하고, DFR 시스템을 처음 사용하는 사용자에게는 자기 자신의 일정 그룹을 루트 그룹내에 자동적으로 생성시 키도록 하였다. 그리고 DFR-클라이언트의 호스트명은 'creohyu' 이고, DFR-서버의 호스트명은 'hyuee' 이다. 먼저 DFR 시스템을 사용하려는 사용자는 서버 시스템에 등록되어 있어야 하며, 서버는 '/etc/passwd' 파일에 등록된 패스워드로 문서 저장기의 접근 여부를 체크하게 된다. 사용자는 사용하고

자 하는 문서 저장기가 위치하는 서버명을 지정한 후 이름과 패스워드를 입력한다. 이름과 패스워드가 맞으면 연계 설정이 되어 그 이외의 동작을 사용할 수 있다.¹⁹⁾

2) 고찰

DFR 시스템은 일종의 분산 화일 시스템으로써 SUN 사의 NFS(Network File System)나 AT&T의 RFS(Remote File System)와 기능은 비슷하나 OSI 참조 모델에 따른 국제 표준이라는 데 의의가 있다. 원격 동작에 있어서 NFS의 경우는 RPC(Remote Procedure Call) 메카니즘을 사용하고, DFR 시스템의 경우는 ROSE 서비스를 이용한다. 따라서 OSI 참조 모델 계층간의 인터페이스에 의한 시간이 많이 소요되므로 기존의 분산 화일 시스템보다는 성능면에서 떨어지며, 이것이 가장 큰 문제점이라 할 수 있으나 이기간의 시스템간에도 상호운용 가능하다는 장점이 있다. 본 연구에서 구현한 DFR 시스템의 구현 타당성은 클라이언트와 서버간의 동작 시험시 연계 기술자와 호출되는 각 서비스 요소의 프

표 6. DFR-서버 루틴

Table 6. DFR-Server routines.

함수명	기능
op_bind()	클라이언트와 서버의 인계물 설정하고 사용자 이름과 패스워드를 검사하여 DFR 서비스를 초기화한다. 이 때 사용자가 문서 저장기에 처음 접근하는 사용자이면 그 자신의 일정 그룹이 자동적으로 생성되게 된다.
op_create()	문서 저장기내에 새로운 DFR-객체를 생성하고, 그 결과로 생성된 DFR-객체의 고유 식별자를 클라이언트에 넘겨준다.
op_delete()	DFR-객체를 삭제하고, DFR-객체가 속해있던 그룹의 속성부를 수정한다.
op_copy()	DFR-객체의 속성과 내용을 새로운 DFR-객체로 복사해주고, 새로운 DFR-객체가 포함되는 그룹의 속성부를 수정한다.
op_move()	DFR-객체의 속성과 내용을 다른 DFR-객체로 이동시키며, 이 때 원래의 속성이 새로운 DFR-객체에 맞도록 수정된다.
op_read()	DFR-객체의 속성과 내용을 요청자에게 알려준다.
op_modify()	DFR-객체의 속성 또는 내용을 수정하여 준다.
op_list()	일정 그룹에 속한 DFR-객체의 속성 리스트를 요청자에게 알려주며, 요청자가 DFR-객체의 수를 제한하였을 경우 이 동작에 대한 작업 식별자를 생성하여 알려준다.
op_reserve()	DFR-객체의 예약 레벨을 수정하여 준다. 연계 설정의 해제는 예약 레벨을 최소화시키지 않는다.
op_abandon()	작업 식별자 또는 요청 식별자의 값을 가지고 있는 동작을 정지 취소시킨다.

리미티브를 추적함으로써 검증하였다. 그림 11은 동작 시험시 서버 시스템의 콘솔에 나타난 추적값의 한 예를 보인 것이다.

그림 11에서 "4"는 클라이언트와 서버간의 연계 기술자이며, 이 값을 가지고 A-ASSOCIATE와 A-RELEASE 프리미티브를 이용하여 연계 설정과 해제를 수행하며 그 이외의 원격 동작은 RO-INVOKE 프리미티브를 이용함을 알 수 있다.

다른 DFR 시스템과의 상호 운용성 시험은 현재 발표된 사례가 없어 시행하지 못 하였으며, 본 연구에서 DFR 시스템 구현시 ISODE 소프트웨어를 이용하였지만 표준안을 충실히 따랐으므로 ACSE, ROSE와의 인터페이스 부분만 조절하여 준다면 다른

소프트웨어를 사용하더라도 무방하다.

V. 결론

본 논문에서는 현재 ISO에서 표준화 작업을 진행 하고 있는 분산 사무 응용시스템 모델의 한 분야인 DFR 시스템의 구현에 대하여 기술하였다. DFR 시스템의 구현은 크게 문서 저장기와 DFR 동작 구현의 두 가지로 나누어지는데, 문서 저장기의 확장성과 계층적 화일링 기능을 위하여 UNIX 화일 시스템을 이용하였으며 DFR 동작 서비스 요소의 구현을 위하여 ISODE 소프트웨어를 이용하였다.

본 연구의 결과는 표준안에 따라 DFR 시스템을 구현함으로써 ISO에서 제안한 문서 인쇄 시스템, 폰트 서비스(font service)시스템 등의 다른 분산 사무 응용시스템 개발에 기초가 될 것이며, 기존에 개발된 메시지 처리 시스템, 디렉토리 시스템 등을 통합시킨 분산 사무 응용시스템 개발의 초석을 마련할 수 있리라 생각된다. 또한 PC나 워크스테이션이 주를 이루는 분산 사무 환경, 즉 분산 사무 시스템이나 이동 분산 시스템에서의 정보 구축에 중요한 역할을 할 것이다.

향후 연구 방향으로서는 DFR 시스템의 성능을 향상시키는 문제, X 윈도우를 이용한 그래픽 사용자 인터

```

T-CONNECT.INDICATION: <4, Internet=128.134.64.4,
U410H/Internet=128.134.64.1, 0, 2041>
starting
A-ASSOCIATE.INDICATION: <4, "isode document dfr",
<,,>, <1.17.4.1.7,,>, 0>
RO-INVOKE.INDICATION/4: list
RO-INVOKE.INDICATION/4: delete
RO-INVOKE.INDICATION/4: create
RO-INVOKE.INDICATION/4: list
A-RELEASE.INDICATION/4: unbind
    
```

그림 11. 추적 상태의 예
Fig. 11. Example of trace status.

페이스(GUI: Graphical User Interface)의 개발, DFR 시스템과 이미 개발된 메시지 처리 시스템 및 디렉토리 시스템과의 연계 구동을 이룰 수 있는 분산 사무 응용시스템을 개발하는 것이다.

參 考 文 獻

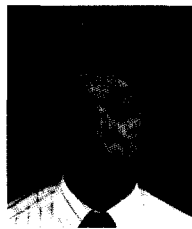
- [1] ISO 10031-1. Distributed Office Application Model (part 1: General Model).
- [2] ISO 10031-2. Distributed Office Application Model (part 2: Distinguished object reference and associated procedures).
- [3] ISO 10166-1. Document Filing and Retrieval (part 1: Abstract service definition and procedure).
- [4] ISO 10166-2. Document Filing and Retrieval (part 2: Protocol specification).
- [5] ISO/DIS 10175-1. Document Printing Application (part 1: Abstract service definition and procedure).
- [6] ISO/DIS 10175-2. Document Printing Application (part 2: Protocol specification).
- [7] P. Chilton, "Current status of DOAM," International Open Systems 89, Blenheim Online Publication, pp.207-218, 1989.
- [8] Yuji Miyawaki, "A Feasibility Study on DFR," 일본 정보 처리 학회 멀티 미디어 통신과 분산 처리 연구회지, vol. 47, no. 7, pp.1-8, 1990.
- [9] AOW Scope and Scenario of PCI ISP ADFR11, Document Filing and Retrieval, 1992.
- [10] Marshall T. Rose, The ISODE: User's Manual (Version 6.0), Performance Systems International Inc., 1989
- [11] CCITT Recommendation X.208, X.209, Specification of ASN.1, 1988.
- [12] CCITT Recommendation X.219, Remote Operations: Model, Notation, and Service definition, 1988
- [13] CCITT Recommendation X.227, Association Control Protocol specification, 1988.
- [14] CCITT Recommendation X.400, Message Handling System, 1988.
- [15] CCITT Recommendation X.500, The Directory, 1988
- [16] EXOS 8014-01 TCP/IP Network Software Reference Manual, Excelan Inc., 1987.
- [17] EXOS 205T and 205E Intelligent Ethernet Controllers Reference Manual, Excelan Inc., 1987.
- [18] Jae-Hong Yim, Yong-Jin Park, "Implementation of DFR(Document Filing & Retrieval) System - A Status Report -," Proceedings of 5th JCCW, pp.143-152, 1990.
- [19] 임재홍, 김문희, 박용진, 최종연구보고서 "문서 보관 및 검색 시스템의 개발에 관한 연구," 한양대학교 산업과학연구소, 1992.

著 者 紹 介



林宰弘(正會員)

1986年 서강대학교 전자공학과 졸업(공학사). 1988年 한양대학교 대학원 전자공학과 석사학위 취득. 1990年 ~ 현재 한양대학교 대학원 전자공학과 박사과정.



朴容震(正會員)

1969年 와세다 대학교 전자통신학과 졸업. 1971年 와세다 대학원 석사학위 취득. 1978年 와세다 대학원 박사학위 취득. 1979年 ~ 현재 한양대학교 전자공학과 교수. 1983년부터 1년간 Univ. of Illinois, Urbana 전산학과 방문 부교수. 1991년부터 1년간 영국 Kent 대학 방문교수. 주관심 분야는 컴퓨터 통신, 분산 시스템, 그룹 웨어 등임.