

DICOM 툴킷 소프트웨어의 개발에 관한 연구

연세대학교 대학원

의 공 학 과

김 동 선

김동선의 박사 학위논문을 인준함

심사위원 김 동 윤 인

심사위원 조 호 성 인

심사위원 홍 인 국 인

심사위원 한 봉 수 인

심사위원 김 영 호 인

연세대학교 대학원

2003년 12월 일

감사의 글

부족한 논문입니다. 그나마 받을 영광이 있다면 아무런 공로가 없는 저에게 늘 복을 주시는 하나님께 돌립니다. 오랜 기간동안 저를 완전히 신뢰하여 주시고 무능력한 제자로부터 많은 것들을 이끌어내셨으며 학문과 인생을 고루 지도하여 주신 김동운 교수님께 감사드립니다. 또한 의공학과와 발전을 이루시어 저와 동문들에게 많은 기회를 주시고 바쁘신 가운데에도 늘 관심을 가져주신 윤형로 교수님께 감사를 드립니다.

학과 행정을 돌보시느라 바쁘신 와중에도 논문을 지도해주신 김영호 학과장님과 방사선과 조효성 학과장님, 실무 경험으로 다양한 조언을 해주신 방사선학과 한봉수 교수님, 귀중한 시간을 내주시어 임상적 조언을 해주신 원주의과대학 진단 방사선과학 홍인수 주임 교수님께 깊은 감사를 드립니다.

부족한 저의 행실을 감싸주신 이윤선 교수님, 학문과 더불어 하나님의 말씀을 전해주시는 이경중 교수님, 경제적으로 큰 도움이 되는 강의를 믿고 맡겨주신 윤영로 교수님, 대학원 과정 중에서 많은 가르침을 주신 안우희 교수님, 신태민 교수님, 김법민 교수님, 김한성 교수님, 이철규 교수님께 감사를 드립니다. 또한 늘 큰 형님 같아 언제나 찾아가도 편안하게 맞아주시고 많은 도움을 주신 안종수 선생님께도 감사드립니다.

막다른 문제에 봉착할 때마나 늘 해결책을 마련해주고 저의 코드를 자신의 코드에 신뢰하고 사용하여준 박사과정 신동규 학형이 아니었다면 아무런 발전이 없었을 것입니다. 항상 뒤에서 굳은 일들을 하며 도와준 사랑하는 후배들에게도 감사를 드립니다. 무슨 일을 맡겨도 항상 훌륭히 해내어 저의 부담을 덜어준 이경무 학형, 선배를 잘 챙겨준 사려 깊은 김영대 학형, 좋은 회사를 떠나 학교로 돌아온 정광영 학형과 조정진 학형, 귀여운 막내 김영윤 학형과 이제 막 입학하는 이용환 학형과 최석우 학형, 모두 신세만 졌습니다. 이제는 졸업하여 각자의 분야에서 최고가 된 신동조, 정정원, 김혁, 김정한, 권의철, 박정근, 윤용현, 송지웅, 조대식 동문들에게도 감사를 드리며 특별히 저의 빈번한 부탁에 언제나 정성껏 도움을 준

박순만, 김대영, 정진덕 동문에게 감사할 드립니다.

풍부한 방사전 실무 경험으로 많은 도움을 주신 리스텍 DR 사업부의 윤종민 과장님, 이종인 대리님께 감사드리고 논문에 반드시 필요한 임상 데이터를 제공하여 주시고 툴킷의 문제점에 대하여 상세히 알려주신 (주) 피플넷커뮤니케이션즈의 연구원님들께 감사할 드립니다.

저의 미래를 늘 걱정해 주시고 기도해 주시는 장인, 장모님께 감사드리고 항상 격려해주신 원광희, 송정은, 안재성, 송운아 형님 동생들에게도 감사드립니다. 또한 저의 졸업을 가족보다 더 기뻐해주신 원주제일감리교회 단계 20속 식구들에게 감사할 드립니다.

어려운 일이 있을 때마다 항상 힘이 되어주시는 가장 가까운 가족이신 박영우 교수님과 사모님께 참 감사드리며 예쁜 동생들인 재연, 주연이도 참 고맙습니다. 제가 어릴 때부터 많은 사랑을 베풀어 주신 김영자, 김차남 이모님, 김재식 외삼촌님과 가족 분들께 감사할 드립니다. 미국에 계시며 응원해 주신 두 작은 아버님과 어머님께 감사드리고 동환 형님, 명선, 현경, 한나, 동일 동생들에게도 참 신세를 많이 졌습니다.

저의 졸업이 학문과 인생의 선배이시고 늘 주시지만 하시는 부모님께 작은 보람이 되었으면 합니다. 또한 이제 저의 신앙의 선배가 되어 저를 위해 저보다 더 많이 기도해주시는 명아 누나에게 감사드립니다.

너무나도 많은 분들에게 감사할 드려야 하나 저의 기억은 부족하기만 합니다. 다시 한번 저에게 도움을 주신 모든 분들께 감사할 드리며 끝으로 저의 작은 세계에 마법을 걸어준 현지와 수빈에게 좋은 남편과 아빠가 될 것을 약속하며 끝을 맺습니다.

2003년 12월

김동선 드림

차 례

그림 차례	iii
표 차례	vi
국문 요약	vii
제 1 장 서 론	1
제 2 장 재료 및 방법	5
2.1 DICOM 개요	5
2.1.1 정보 객체 클래스	5
2.1.2 DICOM 통신	8
2.1.3 DICOM 사용 예	10
2.2 병원 환경	11
2.2.1 PACS의 보급	11
2.2.2 PACS의 구성	11
2.2.3 방사선 검사 작업의 흐름	15
2.2.4 병원 환경의 특징	17
2.3 개발 환경	18
2.3.1 개발 도구	18
2.3.2 테스트 영상 및 환경	18
2.3.3 밸리데이션 도구	20
2.4 설계	21
2.4.1 데이터 모듈군	22
2.4.2 영상 모듈군	24
2.4.3 네트워크 모듈군	29
제 3 장 결과	33
3.1 데이터 모듈	33
3.1.1 데이터 사진 모듈	33

3.1.2 디코딩	35
3.1.2 인코딩	36
3.2 영상 모듈	38
3.2.1 영상 출력 관련 기능	41
3.2.2 영상 처리 기능	47
3.2.3 주석	48
3.3 네트워크 모듈	50
3.3.1 DIMSE	50
3.3.2 DICOM 서비스 클래스	51
3.4. 테스트 PACS 구현	58
3.4.1 워크리스트 서버	58
3.4.2 영상 획득 서버	59
3.4.3 데이터베이스 서버	60
3.4.4 뷰어	62
제 4 장 고찰	71
4.1 툴킷의 성능 고찰	71
4.2 툴킷 개발의 장애 요소	71
4.3 응용 사례	72
4.4 개선 방향	75
제 5 장 결론	77
참 고 문 헌	79
ABSTRACT	81

그림 차례

그림 2.1 CT영상 전송 과정의 예	9
그림 2.2 PACS의 구성	12
그림 2.3 방사선 검사의 흐름	16
그림 2.4 툴킷의 구성	22
그림 2.5 데이터 집합 모듈군의 ERD	23
그림 2.6 영상 모듈군의 ERD	25
그림 2.7 네트워크 모듈군의 ERD	29
그림 2.8 메시지 교환 절차의 예	30
그림 2.9 연합 과정	31
그림 2.10 DICOM Print 과정	32
그림 3.1 데이터 사전 파일 편집기	35
그림 3.2 데이터 집합 디코딩 코드의 예	36
그림 3.3 영상 인코딩 코드의 예	37
그림 3.4 메시지 인코딩 코드의 예	38
그림 3.5 영상 출력 코드의 예	40
그림 3.6 오버레이	41
그림 3.7 영상 방위	42
그림 3.8 원영상	43
그림 3.9 보간이 적용된 영상	43
그림 3.10 확대	44
그림 3.11 이동	44
그림 3.12 윈도우 조정	45
그림 3.13 부분 확대	45
그림 3.14 반전	46
그림 3.15 커브 데이터의 출력	46

그림 3.16 흐름 처리	47
그림 3.17 선명 처리	47
그림 3.18 외곽선 검출	48
그림 3.19 잡음 제거	48
그림 3.20 회전	48
그림 3.21 대칭	48
그림 3.22 주석 객체	49
그림 3.23 Verification Service Class 사용 예	52
그림 3.24 Storage Service Class의 사용 예	52
그림 3.25 Query/Retrive Service Class 사용 예 (C-FIND)	53
그림 3.26 Query/Retrive Service Class 사용 예 (C-MOVE)	54
그림 3.27 Query/Retrive Service Class 사용 예 (C-GET)	55
그림 3.28 Basic Print Management Service Class 사용 예	56
그림 3.29 Basic Worklist Management Service Class 사용 예	57
그림 3.30 조회용 뷰어	58
그림 3.31 워크리스트 서버	59
그림 3.32 영상 획득 서버	60
그림 3.33 데이터베이스의 ERD	61
그림 3.34 데이터베이스 서버	62
그림 3.35 윈도우 탐색기 스타일의 영상 검색	63
그림 3.36 로컬 데이터베이스 메뉴	64
그림 3.37 원격 서버의 영상 검색	65
그림 3.38 DICOMDIR로 기록된 미디어 검색	66
그림 3.39 영상 출력 창	67
그림 3.40 도구 모음 바	68
그림 3.41 영상 필터 대화상자	69
그림 3.42 프린트 대화상자	70
그림 4.1 L사의 CT 콘솔	72

그림 4.2 P사의 PACS 뷰어	73
그림 4.3 L사의 디지털 투시 콘솔	74
그림 4.4 H사의 안과용 영상 관리 시스템	75

표 차 례

표 2.1 DICOM 서비스 클래스의 예	6
표 2.2 DICOM 정보 객체 클래스의 예	6
표 2.3 Composite DIMSEs	8
표 2.4 Normalized DIMSEs	8
표 2.5 수집된 영상의 종류 및 제조사	19
표 2.6 테스트 영상	19
표 2.7 테스트 환경	20
표 2.8 영상 출력을 위해 데이터 집합으로부터 획득한 파라미터	26
표 3.1 데이터 사전의 파일 포맷	34
표 3.2 데이터 집합의 디코딩 소요 시간 (단위: ms/dataset)	36
표 3.3 영상 출력 시간 (단위 : ms/image)	39
표 3.4 디코딩 및 영상 출력의 종합	39
표 3.5 구현된 DIMSE	50
표 3.5 구현된 DICOM 서비스 클래스	51

국 문 요 약

DICOM 툃킷 소프트웨어의 개발

DICOM은 디지털 의료 영상 장비간의 통신을 위한 표준이며 DICOM 툃킷은 이러한 표준을 구현한 소프트웨어로서 의료 영상과 관련된 제품을 개발할 때 필수적인 요소이다. 기존의 툃킷들은 다음과 같은 문제점들을 가지고 있다. 1) DICOM 관련 기능만을 제공하므로 영상 출력 및 처리와 관련된 부분은 별도로 구현해야 한다. 영상 부분까지 통합된 툃킷도 있지만 의료 영상과 관련된 기능은 제공하지 않는다. 2) 유닉스를 주 기반으로 개발된 후에 윈도우즈용으로 이식된다. 이는 윈도우즈를 주로 하는 병원 환경에서 최상의 성능을 발휘할 수 없다. 3) 속도가 느리거나 메모리 소모가 심하다. 이것은 대량의 영상을 처리해야 하는 병원 환경에 적합하지 않다. 4) 툃킷의 구조가 매우 복잡하다. 이것은 DICOM의 방대함에 기인하며 효율적인 구조가 필요하다

본 논문에서는 기존 툃킷의 단점을 보완한 다음과 같은 특징을 갖는 DICOM 툃킷을 구현하였다. 1) DICOM 관련 기능과 영상 관련 기능을 모두 구현하였으며 추가로 의료 영상 관련 기능들을 구현하였다. 2) 범용 컴퓨터를 주로 사용하는 병원 환경에 적합하도록 윈도우즈 운영체제를 기반으로 하였다. 3) 빠른 영상 출력 속도를 보장하며 물리적 메모리를 거의 사용하지 않도록 설계하였다. 4) 의료 영상 저장 시스템 등의 환경에서 필요한 대부분의 의료 영상 관련 기능이 포함되어 있으며 복잡성을 줄이기 위해 객체 지향적인 구조로 설계하였다. 구현된 툃킷은 다양한 모달리티와 장비회사들에서 생성된 120장의 영상에 대해 테스트한 결과 모든 영상들을 12초 이내에 출력하였으며 메모리를 거의 소모하지 않았다. 그리고 제안된 툃킷으로 의료 영상 저장 시스템 뷰어, 디지털 투시 촬영 장비 콘솔, 전산화 단층 촬영 장비 콘솔 및 디지털 검안기용 영상 관리 시스템 등에 성공적으로 사용되었다.

핵심되는 말 : 툃킷, 의료 영상 저장 시스템

제 1 장 서 론

1970년대부터 시작된 전자 공학 기술과 정밀 의학 기술의 발달로 인하여 다양한 첨단 영상 장비들이 개발되었으며 인체 질병의 진단에 획기적인 기여를 하였다. 1980년대에는 영상 획득 시스템과 출력 장치, 저장 장치, 병원 정보 시스템이 급속하게 성장하였고, 그 결과 장비들 간의 연결과 상호 작용이 매우 빈번하게 발생되었으며 의료 영상 장비 표준화의 필요성이 대두되기 시작하였다.[1]

American college of radiology (ACR)와 national electrical manufacturers association (NEMA)은 1983년 의료 영상의 표준화에 관한 조직을 처음으로 결성하였다. 이 조직의 목적은 사용자가 원하는 의료 영상 장비의 연동을 위한 표준안 작성으로서 구체적으로 각 장비별 하드웨어간의 연결 방식은 물론 영상 및 일반 정보의 해석을 위한 표준 양식까지를 포함하고 있었다. 기존 자료를 분석한 결과 모든 것을 만족할 만한 표준은 없었지만 몇 가지 유용한 방식에 관한 정보를 얻게 되었다. 그것은 미의물리학회(American association of physicists in medicine, AAPM) 에서 사용하는 방식으로서 영상 데이터 앞에 헤더(header)를 사용하여 환자 및 검사 정보를 기록하였다. 헤더에는 가변적 길이를 가지는 태그(tag)와 키(key)를 가지고 정보를 등록하였다. 장비간의 프로토콜은 널리 사용되고 있는 transmission control protocol/internet protocol (TCP/IP)을 채택하였고 open systems interconnection (OSI)에서 정의한 네트워크 7 단계 모델을 사용하였다.[2]

1985년에는 ACR-NEMA 조직이 결성된 후 2년간의 작업을 바탕으로 ACR-NEMA 300-1985 (ACR-NEMA 1.0판)를 북미 방사선 학회(radiological society of north America, RSNA)에서 발표하게 되었으나 많은 예외와 실제 연동 적용에 적합하지 않은 점들이 발견되어 이러한 점을 보완한 ACR-NEMA 300-1988 (ACR-NEMA 2.0판)을 1988년 다시 내놓게 되었다. 그 당시 사용자들로부터 가장 요구되었던 사항은 영상 장비간의 네트워크 연동이었다. 그러나 2.0판에서도 역시 사용자들이 원하는 수준의 네트워크 지원 기능을 가지고 있지 못하

였다.[2]

1992년에는 RSNA에서 digital imaging and communications in medicine (DICOM) 표준이 일반에 최초로 발표되었다. 새롭게 발표된 표준은 설계 단계부터 객체 지향적인 방식을 도입하였으며 영상 장비를 네트워크에 직접 연결이 가능하게 되어 일반 정보와 영상 정보를 결합하여 전송하는 기능을 지원하게 되었다. 그 이후에 표준화 작업을 위하여 의 각 워크 그룹(work group)이 형성되었으며 이들 워크 그룹간의 협의가 거듭되어 1993년 RSNA 학회에서 처음으로 시연을 가졌다. 워크 그룹의 노력과 관련 업체들의 협조, 그리고 사용자들의 인식이 높아지며 DICOM은 표준으로서 발전을 거듭하여 최근에는 2001년 판인 DICOM 2000이 발표되기에 이르렀다. DICOM은 이제 의료 영상 장비의 표준으로 확고한 자리를 잡아가고 있으며 더 나아가 표준화 영역을 영상 이외의 범위로 넓혀 가고 있다.[2]

DICOM 표준의 2003년 판은 기본 16권, 추가본 82권, 수정본 약 377권으로 매우 방대한 문서로 되어 있으며 매년 새로운 문서가 추가되고 있다. 업계에서는 이렇게 빠르게 발전하는 표준에 적시에 따를 수가 없으며 업계 상호간 적절한 협의 하에 점차적으로 장비에서 준수하는 표준의 범위를 넓혀가고 있다. 의료 영상 장비 또는 컴퓨터상에서 작동되는 소프트웨어가 DICOM 표준을 따르기 위해서는 DICOM 표준에 있는 사항 중에 필요한 부분들을 각 업체마다 구현해야 한다. 90년대 초반에는 GE, Philips, Siemens 등과 같은 다국적 기업에서 각각 DICOM을 구현하였다. 그러나 표준을 준수하는 업체가 증가하며 모든 업체가 DICOM 표준을 구현해야 하는 문제가 대두되었다. 이러한 환경에서 발전된 소프트웨어가 DICOM 툴킷(toolkit)이다. 1993년 RSNA의 주도로 최초의 공개 툴킷이 발표된 이후에 수많은 상용 및 공개 툴킷이 발표되었다. 현재 사용되고 있는 대표적인 툴킷들을 살펴보면 다음과 같다.[3]

Mallinckrodt institute of radiology central test node (MIR CTN)는 DICOM 표준을 구현한 최초의 툴킷이다. RSNA의 기금으로 1992년 RSNA에서 시연되었으며 매년 기능을 향상시켜 발표하였다. MIR CTN은 오픈 소스(open source) 형태로 제공되고 있으며 유닉스(UNIX)와 마이크로소프트 윈도우즈(Microsoft

Window)용으로 개발되고 있다. 별도의 사용료가 없으므로 많은 분야에서 널리 사용되고 있으며 가장 강력한 툴킷 중 하나이다. 그러나 사용법이 복잡하고 유닉스 환경에서 개발되어 윈도우 환경에서는 기능의 한계가 있으며 DICOM 표준만을 구현하여 영상 출력 부분은 별도로 구현해야 하는 단점이 있다.[4]

Oldenburger Forschungs und Entwicklungsinstitut für informatic werkzeuge und systeme central test node (OFFIS CTN)은 MIR CTN의 발표와 비슷한 시기인 1993년 최초로 발표되었다. 유럽의 표준 기관인 European committee for standardization (CEN/TC251/WG4)이 사업을 추진하였으며 European CTN으로 잘 알려져 있다. 초기의 소프트웨어는 MIR CTN의 일부를 사용하였으나 그 이후에 많은 발전을 하여 현재 버전은 여러 측면에서 MIR CTN의 기능을 능가한다. 1996년 OFFIS DICOM toolkit (DCMTK) 라는 이름으로 다시 발표되었고 1999년에는 integrating the healthcare enterprise (IHE) 테스트용 소프트웨어로 잘 알려진 DICOMScope를 발표하였다. 유닉스와 윈도우 환경에서 사용할 수 있도록 오픈 소스 형태로 개발되었으며 역시 별도의 사용료가 없다. 매년마다 새로운 버전을 발표하며 꾸준히 발전하는 소프트웨어로 그 성능이 뛰어나다. 그러나 일부 기능은 유닉스 버전에만 존재하며 MIR CTN과 같이 영상 디스플레이 관련하여서는 빈약한 기능들을 보이는 것이 최대의 단점이다.[5]

Merge사의 DICOM 툴킷은 상용 소프트웨어들 중에 가장 널리 알려져 있으며 수많은 의료 영상 관련 업체에서 사용한다. DICOM 표준을 따르는 제품들이 생산되기 시작할 때부터 툴킷이 생산되었기 때문에 매우 오랜 기간의 기술력을 축적하였다. 상용 소프트웨어답게 사용하기가 쉽고 안정성이 우수하다. CTN과 마찬가지로 DICOM 표준만 구현하므로 영상 디스플레이 부분은 별도로 구현해야 한다. 매우 높은 구입가와 및 로열티를 지불해야하는 것이 단점이다.[6]

Lead Technologies사의 LeadTools 소프트웨어는 영상 라이브러리로 매우 명성을 쌓은 제품이다. 이 제품에 DICOM 관련 기능을 추가한 “Medical Suite” 라는 제품을 90년대 말 발표하였다. 툴킷에서 대부분의 영상 처리 기능을 제공하므로 개발자의 입장에서는 매우 편리한 제품이었다. 짧은 개발 시간을 요구하거나 구현해야 할 제품이 소규모일 때 주로 사용된다. 주로 DICOM 영상을 출력하는데 사용

되며 네트워크 관련 기능들은 충분히 검증되지 않아 널리 사용되지 않고 있다.[7]

기존 툴킷들의 단점들을 정리하면 다음과 같다. 첫 번째, DICOM 관련 기능을 제공하므로 영상 출력 및 처리와 관련된 부분은 별도로 구현해야 한다. 영상 부분 까지 통합된 툴킷도 있지만 의료 영상과 관련된 기능은 제공하지 않는다. 두 번째, 유닉스를 주 기반으로 개발된 후에 윈도우즈용으로 이식된다. 이는 윈도우즈를 주로 하는 병원 환경에서 최상의 성능을 발휘할 수 없다. 세 번째, 속도가 느리거나 메모리 소모가 심하다. 이것은 대량의 영상을 처리해야 하는 병원 환경에 적합하지 않다. 네 번째, 툴킷의 구조가 매우 복잡하다. 이것은 DICOM의 방대함에 기인하며 효율적인 구조가 필요하다.

본 논문에서는 상기한 기존 툴킷의 단점을 보완한 다음과 같은 특징을 갖는 DICOM 툴킷을 구현하였다. 첫 번째, DICOM 관련 기능과 영상 관련 기능을 모두 구현하였으며 추가로 의료 영상 관련 기능들을 구현하였다. 두 번째, 범용 컴퓨터를 주로 사용하는 병원 환경에 적합하도록 윈도우즈 운영체제를 기반으로 하였다. 세 번째, 빠른 영상 출력 속도를 보장하며 물리적 메모리를 거의 사용하지 않도록 설계하였다. 네 번째, 의료 영상 저장 시스템(picture archiving and communication system, PACS) 등의 환경에서 필요한 대부분의 의료 영상 관련 기능이 포함되어 있으며 복잡성을 줄이기 위해 객체 지향적인 구조로 설계하였다.

또한 본 논문에서는 제안된 툴킷으로 테스트 PACS를 구현하여 툴킷의 성능을 평가하였다. 개발된 테스트 PACS는 소형 PACS로 사용될 수 있으며 디지털 방사선 촬영 장비의 콘솔(console) 등 DICOM 관련 제품을 개발할 때에 테스트 도구로도 유용하게 사용될 수 있다. 테스트 PACS의 구성요소인 워크리스트(worklist) 서버, 영상 획득 서버, 데이터베이스 서버, 영상 조회용 뷰어(viewer)를 각각 구현한다.

제 2 장 재료 및 방법

본 장에서는 툴킷 개발을 위한 이론적 배경을 설명한다. DICOM의 개념에 대하여 간단한 예를 들어 설명하고 툴킷이 사용될 병원 환경의 특징에 대하여 기술한다. 또한 툴킷을 개발할 각종 도구와 테스트 영상들을 소개하며 세부적인 설계 내용과 구현되어야 할 기능들을 설명한다.

2.1 DICOM 개요

DICOM의 두 가지 기본 구성요소는 정보 객체 클래스(information object class)와 서비스 클래스(service class)이다. 객체 및 클래스라는 용어를 사용하는 이유는 DICOM이 객체 지향적으로 설계되었기 때문이다. 정보 객체는 영상에 포함되어야 하는 내용을 정의하며 서비스는 정보 객체들을 이용하여 무엇을 할지를 설명한다. 정보 객체 클래스와 서비스 클래스는 서비스 객체 쌍(service-object pair, SOP)이라 호칭되는 DICOM의 기본 요소를 구성한다. 본 절에서는 상기한 클래스들의 개념에 대하여 설명한다.[1][2]

2.1.1 정보 객체 클래스

DICOM 표준에는 두 가지 종류의 클래스인 정보 객체 클래스와 서비스 클래스가 존재한다. 정보 객체 클래스는 객체(예: 환자, 모달리티, 검사)들을 정의한다. 반면에 서비스 클래스는 서비스(예: 영상 저장, 프린트, 질의 및 응답)를 정의한다. 표 2.1과 표 2.2는 서비스 클래스와 객체 클래스의 예를 보여주고 있다.

표 2.1 DICOM 서비스 클래스의 예

서비스 클래스	설명
Verification service class	연결 여부 확인
Storage service class	영상의 저장
Query/retrieve service class	영상의 조회 및 가져오기
Print management service class	프린트 관리
Basic worklist management service class	워크리스트 관리

표 2.2 DICOM 정보 객체 클래스의 예

형태	종류	설명
Normalized	Patient	환자
	Study	검사
	Results	결과
	Image annotation	영상 주석
Composite	Computed radiography image	CR 영상
	Computed tomography image	CT 영상
	Magnetic resonance image	MR 영상
	Nuclear medicine image	핵의학 영상
	Ultrasound image	초음파 영상
	Curve	1차원 신호

(1) 정보 객체 클래스

DICOM 정보 객체 클래스는 정규 객체(normalized object)들과 복합 객체(composite object)들로 구성되어 있다. 정보 객체는 실제 세계에서 발생하는 속성들을 포함한다. 표 2.2에 열거된 검사 정보 객체와 환자 정보 객체를 예를 들어 설명하면 검사 날짜와 시간은 검사가 이루어질 때마다 발생되므로 검사 정보 객체 클래스의 속성이며 환자 이름은 검사가 이루어질 때 사용되는 환자 정보 객체

클래스에서 발생되기 때문에 환자 정보 객체 클래스에 속한다. 의료 영상 응용 프로그램에서는 이러한 정보 객체 클래스들을 사용함으로써 정보의 모호함을 배제하여 정확하게 구분할 수 있다. 이러한 이유로 DICOM에는 정보 객체들을 매우 자세히 정의하고 있다. 때로는 정규 객체 클래스들을 묶어서 복합 객체 클래스로 구성하는 것이 실용적으로 유용할 수 있다. 예를 들어서 CT 영상 정보 객체 클래스는 검사 정보 객체 클래스와 환자 정보 객체 클래스의 속성들을 모두 가지고 있기 때문에 복합 형태이다.

DICOM은 객체를 구분하기 위하여 unique identifier (UID)를 사용하며 "1.2.840.10008.X.Y.Z"와 같은 방법으로 사용한다. 숫자 부분은 기관을 의미하며 X, Y, Z는 세부적인 객체를 구분하기 위하여 사용한다. 예를 들어 DICOM에서 사용되는 "Explicit value representation little endian transfer syntax"는 "1.2.840.10008.1.2.1"이다. UID는 객체를 구분하기 위하여 사용될 뿐 정보를 포함하지 않는다.

하나의 정보 객체는 객체의 설명, 객체가 가지고 있는 값, 값의 길이, 값의 형태의 속성(attribute)을 가지고 있다. 이 속성들이 변환 문법(transfer syntax)에 의하여 인코딩(encoding)되면 실존하는 데이터 값이 되며 이것을 데이터 요소 (data element)라고 한다. 반대로 데이터 요소를 디코딩(decoding)하면 객체의 속성들을 얻을 수 있다.

정보 객체 클래스는 하나 이상의 정보 객체들로 구성되므로 데이터 요소들의 집합이 되며 이것을 데이터 집합 (data set)이라고 부른다. 일반적으로 데이터 집합은 DICOM 파일로 취급된다.

(2) DICOM 서비스

DICOM 서비스란 정보 객체 클래스를 저장하거나 출력하는 등의 작업을 수행하는 것을 의미하며 하나의 장비 안에서 이루어질 수도 있고 두 장비 간에 이루어질 수도 있다. DICOM 서비스는 DICOM 메시지 서비스 요소(DICOM message service element, DIMSE) 라는 특별한 기능들 하도록 구성된 컴퓨터 소프트웨어 프로그램을 사용하여 수행된다. DIMSE는 수행되는 객체의 형태에 따라서 표 2.3,

2.4와 같이 정규 DIMSE와 복합 DIMSE로 구분된다. DIMSE는 한쪽 장비에서 명령을 발생시키고 다른 한쪽장비에서 응답하는 구조로 되어있으므로 쌍을 이루는 것이 특징이다. DIMSE는 명령과 정보 객체로 구성되는 메시지를 교환한다. 만일 장비가 서비스를 제공한다면 서비스 클래스 제공자(service class provider, SCP)라고 호칭되고 서비스를 사용하면 서비스 클래스 사용자(service class user, SCU)라고 호칭된다. 예를 들어서 모달리티에서 DICOM 프린터로 출력할 때 DICOM 프린터는 SCP가 되고 모달리티는 SCU가 된다.

표 2.3 Composite DIMSEs

명령	기능
C-ECHO	연결 여부를 확인
C-STORE	정보 객체를 저장
C-FIND	정보 객체를 조회
C-GET	정보 객체를 가져옴
C-MOVE	정보 객체를 이동

표 2.4 Normalized DIMSEs

명령	기능
N-EVENT-REPORT	이벤트를 알림
N-GET	정보 객체를 가져옴
N-SET	정보 객체의 값을 지정
N-ACTION	정보 객체의 행동을 지정
N-CREATE	정보 객체를 생성
N-DELETE	정보 객체를 삭제

2.1.2 DICOM 통신

DICOM은 OSI에 근간한 네트워크 통신 표준을 이용한다. OSI는 가장 낮은 물

리 계층(예 : 케이블)으로부터 가장 높은 응용 계층까지 일곱 계층으로 이루어져 있다. 정보 객체는 두 장비 사이의 여러 단계를 통하여 연결되며 이를 DICOM을 이용하여 연합(association)한다고 한다. 연결된 두 장비는 메시지를 주고받는다. 예를 들어 정보 객체를 CT에서 영상 획득 서버로 전송할 때 그림 2.1과 같은 과정을 거친다. CT로부터 영상 획득 서버로 정보 객체가 전송될 때에 사용되는 통신 프로토콜은 널리 사용되고 있는 TCP/IP이다.

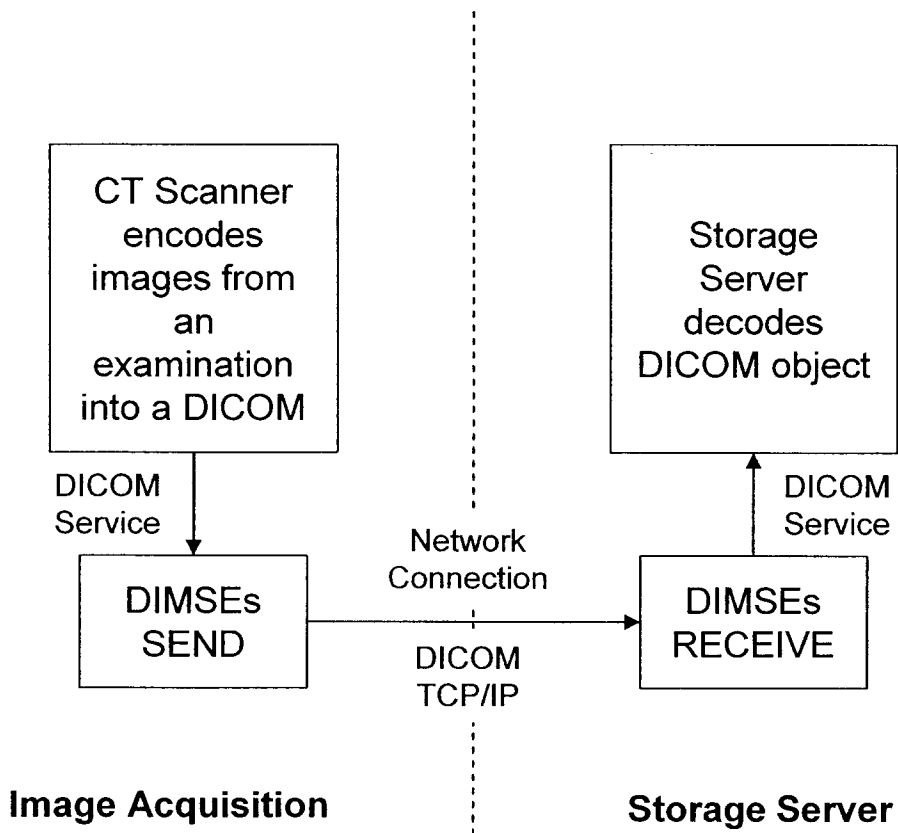


그림 2.1 CT영상 전송 과정의 예

- ① CT는 모든 영상을 DICOM 데이터 집합으로 인코딩한다.
- ② CT는 일련의 DIMSE를 호출하여 명령과 데이터 집합으로 구성된 메시지를

OSI 모델의 물리적 계층으로 보낸다.

③ 영상 획득 서버는 메시지를 물리적 계층으로 받아 대응되는 DIMSE로 보낸다.

④ 영상 획득 서버는 DICOM 메시지에서 명령과 데이터 집합을 구분하여 디코딩한다.

2.1.3 DICOM 사용 예

CT 검사에서 출력된 영상들을 PACS의 영상 획득 서버로 전송하는 과정을 예로 들면 다음과 같다. 각각의 영상들은 DICOM의 C-STORE 서비스를 이용하여 CT에서 영상 획득서버로 전송된다. 전송 과정에서 CT는 C-STORE SCU로서 클라이언트 역할을 하며 영상 획득 서버는 C-STORE SCP로서 서버 역할을 한다.

① SCU와 SCP는 DICOM의 “association request and response” 명령에 의하여 연결을 성립한다.

② SCU는 C-STORE 서비스 요청을 SCP에 전송한다.

③ SCP는 C-STORE 요청을 받아 C-STORE 허가를 SCU로 보낸다.

④ SCU는 첫 번째 영상의 첫 번째 데이터 패킷을 SCP로 보낸다.

⑤ SCP는 요청된 C-STORE 서비스를 수행하여 패킷을 받는다.

⑥ SCP는 서비스가 종료되면 SCU로 확인을 보낸다.

⑦ SCU는 SCP로부터 확인을 받은 후에 다음 패킷을 SCP로 보낸다.

⑧ ⑤~⑦ 과정을 첫 번째 영상이 모두 전송될 때까지 반복한다.

⑨ SCU는 두 번째 영상을 보내기 위하여 C-STORE 서비스 요청을 SCP로 보낸다. ②~⑧ 과정을 모든 영상에 대하여 반복한다.

⑩ SCU와 SCP는 연결을 종료하기 위하여 “association release and response” 명령을 수행한다.

2.2 병원 환경

제안된 툴킷이 최종적으로 사용될 곳은 PACS가 설치된 병원 환경이다. 병원 환경을 간단히 설명한다면 일반 사용자가 제한된 시스템에서 다량의 데이터를 처리하는 환경이다. 본 절에서는 PACS의 개요와 방사선 작업의 흐름을 설명하고 요구되는 기능들을 파악한다.

2.2.1 PACS의 보급

PACS는 디지털 영상 장비로부터 얻은 영상들을 병원의 영상 데이터베이스 시스템에 저장하고 네트워크를 통해 병원 내의 어디에서나 검색 장치를 이용하여 조회가 가능하도록 하는 시스템이다. 병원 정보 시스템(hospital information system, HIS), 처방 전달 시스템(order carrying system, OCS), 임상 정보 시스템(laboratory information system, LIS)과 함께 병원 전산화의 핵심적인 역할을 하고 있다.

PACS가 높은 초기 투자비용의 부담에도 불구하고 급속도로 보급된 이유는 여러 가지가 있지만 가장 큰 이유는 비용 절감 측면이다. 단기적으로 보았을 때에도 필름 및 필름의 보관 장소, 운반 인력, 현상 장소와 현상액, 폐기물 처리, 관리 인력 등에 소모되는 비용에 비교하여 경제력이 있다. 두 번째 이유는 빠른 조회가 가능하기 때문이다. 단기 저장 영상은 수초 이내에, 장기 저장 영상은 수분 이내에 병원내의 어디에서나 조회가 가능하다. 세 번째 이유는 우수한 영상을 얻을 수 있기 때문이다. 필름과 달리 촬영 후에도 언제나 다양한 영상 조작 및 처리에 의하여 새로운 영상을 만들 수 있다. 이 외에도 PACS를 설치함으로써 발생하는 이점은 매우 많기 때문에 보급률이 점차 증가하고 있다.[2]

2.2.2 PACS의 구성

PACS의 목적은 영상을 관리하고 전송하는 것으로서 그 구조는 자유롭게 구성할 수 있으므로 특별히 정하여진 규칙이 없고 제조사별로 상이할 수 있다. 일반적

으로 사용되는 구성법은 그림 2.2와 같으며 100Mbps 이상의 고속 데이터 전송이 가능한 네트워크로 연결되어 있다. 구성요소들의 역할을 각각 살펴보면 다음과 같다.

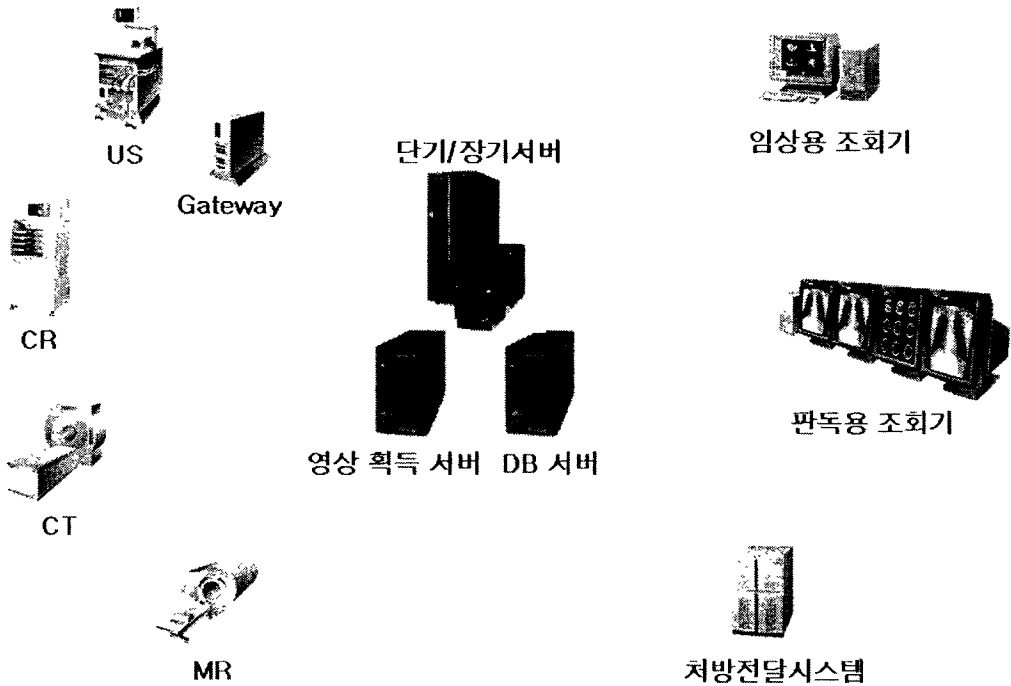


그림 2.2 PACS의 구성

(1) 모달리티

영상이 발생하는 장치이다. 일반적으로 병원에서 가장 많이 사용되는 모달리티들은 computed radiography (CT), magnetic resonance image (MRI), ultrasound (US), computed radiography (CR), digital radiography (DR), nuclear medicine (NM) 등이 있다. 모달리티가 PACS에 연동되기 위해서는 DICOM 표준을 준수하여야 한다. 즉 DICOM 서비스 클래스를 사용하여 워크리스트 서버로부터 검사 정보를 획득하고 영상을 촬영한 후에 DICOM 데이터 집합 형태로 저장하며 이를

PACS의 영상 획득 서버나 DICOM 프린터로 DICOM 서비스 클래스를 이용하여 전송한다. DICOM을 지원하지 않는 구형 장비는 게이트웨이(gateway)라는 매개체를 통하여 DICOM 형태로 변환되어 PACS와 연동된다.

(2) 영상 획득 서버

모달리티들로부터 DICOM 서비스 클래스를 사용하여 영상을 전송 받는 역할을 한다. 영상 획득 서버는 전송 받은 영상으로부터 환자, 검사, 영상 등의 정보를 추출하여 이를 데이터베이스 서버에 저장하고 영상 파일은 단기 저장 서버로 전송한다. 영상 획득 서버는 병원내의 모든 모달리티와 연결되어 매우 많은 양의 데이터를 전송 받는다. 그러므로 1대만을 설치하였을 경우 과부하가 발생하므로 일반적으로 모달리티의 수에 따라 여러 대를 적절히 설치한다. 영상 획득 서버는 수많은 제조사에서 구현된 장비와 DICOM 표준으로 연동되어야 하므로 PACS를 설치할 때에 가장 문제가 많이 발생하는 부분이다.

(3) 데이터베이스 서버

영상 획득 서버로부터 영상에 관련된 정보를 받아 이를 데이터베이스에 보관한다. 병원에 설치된 모든 조희용 단말기는 데이터베이스 서버를 이용하여 영상 정보를 조회할 수 있다. 조희용 단말기는 원칙적으로 DICOM 서비스 클래스를 사용하여 데이터베이스 서버를 조회하여야 하나 일반적으로 데이터베이스 서버와 조희용 단말기가 같은 PACS 회사의 제품이므로 회사 자체의 프로토콜을 이용하는 경우가 더 많다.

(4) 단기 저장 서버

영상 획득 서버로부터 영상을 전송 받아 저장한다. 조희용 단말기로부터 영상 전송 지시가 발생하였을 경우 즉시 영상을 전송하는 것이 단기 저장 서버의 목적이다. 그러므로 저장 미디어로 접근 속도가 가장 빠른 하드 디스크(hard disk)를 사용한다. 보관 기간은 발생하는 영상의 양과 저장 용량에 의존한다. 일반적으로 수개월에서 수년까지 보관하며 데이터의 용량을 줄이기 위하여 무손실 압축을 할

수도 있다. 단기 저장 서버는 데이터베이스 서버와 마찬가지로 DICOM 서비스 클러스터를 사용하여야 하나 PACS 회사 자체의 프로토콜을 이용하는 경우가 더 많다.

(5) 장기 저장 서버

단기 저장 서버에 있는 영상들은 일정 시일이 지나게 되면 장기 저장 서버로 이동된다. 장기 저장 서버의 목적은 사용하지는 않으나 반드시 보관해야 하는 영상들을 비교적 저렴한 미디어에 관리하는 것이다. 저장 미디어로는 가격대 성능비가 우수한 자기 테이프 등을 사용한다. 장기 저장 서버는 미디어 특성상 영상 조회 속도가 매우 느리다. 병원에 따라 손실 압축을 적용하여 영상을 저장하기도 한다. 최근에는 하드 디스크 가격의 하락으로 모든 미디어를 하드디스크로 사용하는 사례가 늘고 있으며 이로 인하여 단기 저장 서버와 장기 저장 서버의 구분이 없어지고 있다.

(6) 조회용 단말기

임상에서 판독 및 진단을 위하여 사용하는 조회 전용 뷰어가 설치된 단말기이다. 조회 전용 단말기는 임상용과 판독용이 있으며 이는 영상을 출력하는 모니터의 성능에 따라 구분된다. 판독용 조회기는 5백만 화소 이상 해상도와 700cd/m² 이상의 밝기를 가지는 방사선 영상 전용 모니터를 사용하며 가격이 매우 높아 판독실에서만 사용된다. 임상용 조회기는 일반 모니터를 사용한다. 조회용 단말기에는 일반인이 사용할 수 있도록 편리한 인터페이스와 기능을 갖춘 뷰어가 있어 영상 출력 및 진단에 관련된 모든 기능들을 제공한다.

(7) 워크리스트 서버

워크리스트란 임상 의사가 지시한 방사선 검사 의뢰서의 목록을 의미한다. 워크리스트 서버는 병원의 처방전달시스템으로부터 검사 관련 정보를 주기적으로 획득하여 보유하고 있어 모달리티에서 조회를 요청할 경우 검사 관련 정보를 알려준다. 워크리스트는 처방전달시스템이 일반적으로 표준을 따르지 않으므로 병원마다

별도의 시스템을 구현해야 하며 또한 수많은 제조사의 장비와 DICOM 서비스 클래스를 이용하여 연동되어야 하므로 PACS를 설치할 때에 문제가 많이 발생하는 곳이다.

2.2.3 방사선 검사 작업의 흐름

PACS 환경에서는 임상이가 환자를 진단한 후에 검사를 의뢰한 시기부터 영상을 관찰할 때까지의 전 과정에 DICOM이 사용된다. 방사선 검사 작업의 흐름은 그림 2.3과 같다.

(1) 방사선 검사 의뢰

임상의는 환자를 진료한 후에 방사선 검사가 필요하다고 판단될 경우 방사선 검사 의뢰서를 처방전달시스템의 단말기에서 작성하여 방사선과에 작업을 지시한다. 방사선 검사 의뢰서에는 환자 이름, ID, 생년월일, 성별, 검사할_MODAL리티, 검사 방법 등의 정보가 포함된다. 워크리스트 서버는 이 정보들을 처방전달시스템으로부터 주기적으로 가져와 보관한다.

(2) 촬영

환자가 방사선과를 방문하면 검사할_MODAL리티의 담당자는_MODAL리티의 워크리스트관련 기능을 이용하여 워크리스트 서버를 조회하여 검사 관련 정보를 획득하고 촬영을 실시한다. 촬영된 영상은_MODAL리티 고유의 영상 형태 또는 DICOM 미디어 파일 형태로 저장된다. 저장된 영상은 PACS의 오류에 대비하여 병원의 내규에 따라 일정기간 보관된다.

(3) 전송

_MODAL리티에 저장된 영상은 DICOM 서비스 클래스를 이용하여 영상 획득 서버로 전송되거나 DICOM 프린터로 출력된다. 영상 획득 서버는 전송 받은 DICOM 파일을 검사하여 이상 유무를 판단한 후에 영상 정보를 추출하여 데이터베이스

서버로 전송하며 영상은 단기 저장 서버로 전송한다.

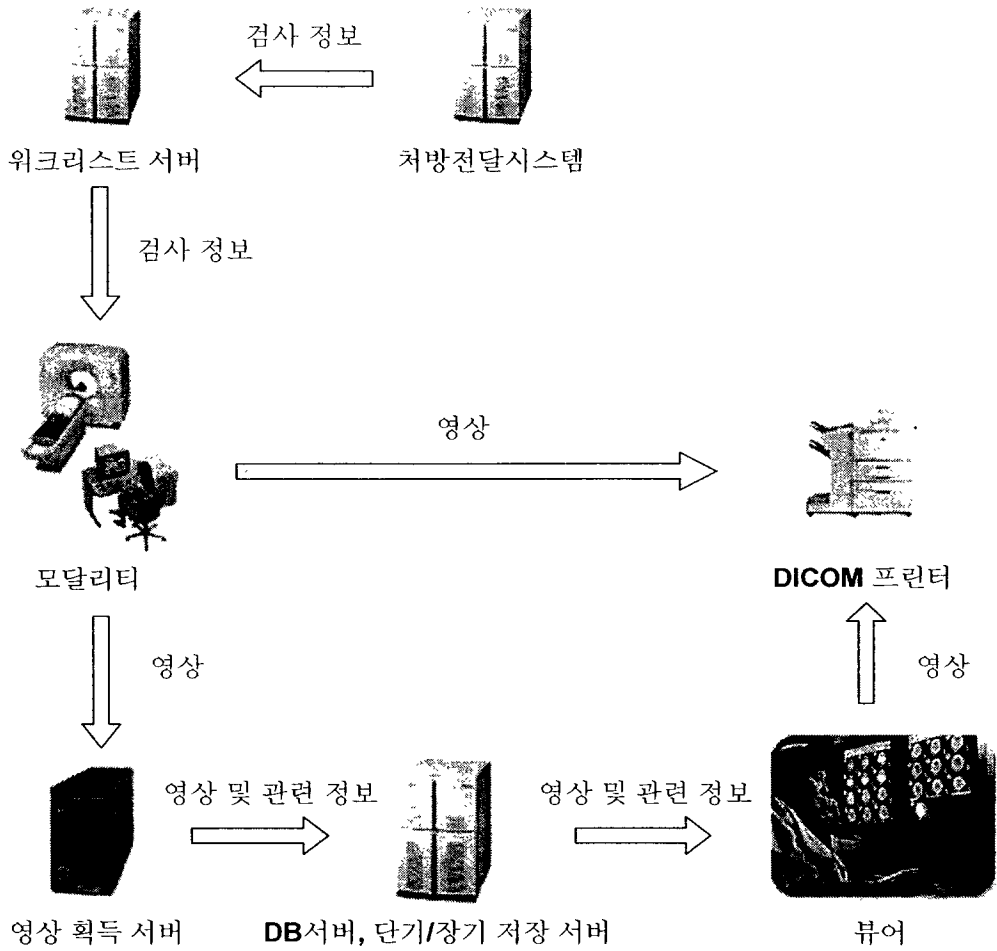


그림 2.3 방사선 검사의 흐름

(4) 조회

관독의는 조회용 단말기를 이용하여 데이터베이스 서버에 영상을 조회한다. 원하는 영상을 찾았을 경우 단기 저장 서버로부터 영상을 불러와 영상을 관독한다. 관독된 영상은 관독결과와 함께 데이터베이스와 단기 저장 서버로 재전송된다. 환

자의 담당의는 최종적으로 조희용 단말기를 이용하여 영상을 조회한다.

2.2.4 병원 환경의 특징

DICOM 관련 기능들은 병원이라는 특수한 환경에서 사용되며 이러한 환경에서 툴킷이 올바르게 동작하기 위해서는 병원 환경이 충분히 고려되어야 한다. 병원 환경의 특징을 살펴보면 다음과 같다.

(1) 윈도우즈 환경

관독실을 포함한 모든 조희용 단말기는 마이크로소프트사의 윈도우즈를 사용하는 일반 업무용 컴퓨터이다. 이러한 현상은 2002년 RSNA에 출품된 PACS 제품들이 100% 윈도우즈를 사용한 것을 반영하며 전통적으로 유닉스를 사용하였던 모달리티들도 매우 빠르게 윈도우즈로 대체되고 있다. 그러므로 DICOM 툴킷은 윈도우즈 환경에 최적화 되도록 설계되어야 한다.[8]

(2) 대용량 메모리 소모

최근에 생산되는 고성능 CT 및 MR은 한번의 검사에서 수백장의 영상이 발생한다. 또한 2000년 이후에 판매가 급증된 DR 관련 장비들에서 발생하는 영상은 9백만 화소 이상의 해상도를 가지며 한 장의 용량이 수십 Mbyte에 이른다.[9] 임상되는 이러한 영상들을 조희용 단말기에서 동시에 관찰할 필요가 있다. 이를 처리하기 위해서는 한계가 명확한 물리적 메모리만을 사용하여서는 불가능한 작업이며 파일 매핑(file mapping) 기법 등을 이용한 효율적인 메모리 관리가 필수적이다.

(3) 다중 쓰레딩(multi threading)

DICOM 관련 작업들 중 상당 부분이 네트워크를 통하여 데이터를 전송하는 작업이다. 네트워크는 속도의 한계가 있으므로 하나의 작업을 수행하며 그 작업이 종료될 때까지 기다려야 한다면 이는 매우 불편할 것이다. 그러므로 DICOM 툴킷

에서 다중 쓰레딩 기능은 필수적이며 매우 안정적으로 동작해야 한다. 예를 들어 모달리티의 경우 영상 전송, 워크리스트 조회 등과 같이 상당한 시간이 소모되는 네트워크 작업을 하며 영상 촬영 등의 모든 작업을 동시에 수행할 수 있어야 한다. 서버의 경우에는 여러 노드에서 들어오는 DICOM 메시지들을 동시에 처리할 수 있어야 한다.

2.3 개발 환경

본 절에서는 툴킷 개발에 사용된 하드웨어 및 소프트웨어 환경에 대하여 기술하고 테스트에 사용된 영상을 소개한다. 또한 툴킷의 표준 준수 여부 평가에 사용된 도구들을 소개한다.

2.3.1 개발 도구

제안된 툴킷은 마이크로소프트사의 윈도우즈 운영체제에서 동작하는 응용 프로그램을 개발할 때 사용하도록 설계되었다. 윈도우즈 98, ME, 2000, XP에서 사용할 수 있으며 윈도우즈 95 이하 버전에서는 사용할 수 없다. 추천되는 운영 체제는 윈도우즈 2000이다.

개발 도구로 마이크로소프트사의 Visual C++ 6.0판을 사용하였으며 C와 C++ 언어를 혼합하여 사용하였다. 툴킷은 Microsoft foundation class (MFC) 확장 dynamic link library (DLL) 형태로 제공된다. Jjoint picture expert group (JPEG) 영상의 디코딩을 위하여 independent JPEG group (IJG)의 코덱(codec)을 사용하였다. 툴킷은 객체 지향적인 구조를 가지며 각각의 네트워크 접속에 대하여 별도의 쓰레드를 가지는 다중 쓰레드구조로 되어있다.

제안된 툴킷은 DICOM 표준의 모두를 구현한 것이 아니며 일반적으로 병원 환경에서 요구되는 기능들을 중심으로 구현하였다.

2.3.2 테스트 영상 및 환경

툴킷 개발을 위하여 표 2.5와 같이 가능한 많은 수의 영상을 확보하였다. 영상

장비 제조업체, RSNA 등의 인터넷 사이트에서는 다양한 모달리티에서 발생하는 샘플 영상을 얻을 수 있었으며 국내 PACS 회사 및 병원에서는 일반적으로 널리 사용되는 모달리티에서 발생하는 영상들을 획득할 수 있었다.

표 2.5 수집된 영상의 종류 및 제조사

모달리티	제조사
CR	Konica, Fuji, Agfa
CT	GE, Siemens, Philips, Picker
MR	GE, Siemens, Philips, Medinus, AILab
NM	ACME
US	ACME
XA	Philips, Shimadzu
DR	Hologic, Infimed, Listem

수집된 영상 중 표 2.6와 같이 테스트 영상을 구성하였다. 테스트 영상을 수집한 장비는 국내 종합병원에서 널리 사용되는 장비로 선택하였으며 모달리티의 종류는 병원에서 가장 많이 발생하는 순서로 채택되었다. 이후 본문에서 표기할 영상에 대한 성능은 본 테스트 영상에 대한 것으로 간주한다.

표 2.6 테스트 영상

장비 종류	제조사	모델명	촬영 부위	영상 개수	Row×Col	영상 크기	Data Element 개수
CT	GE	HiSpeed	복부	50	512×512	0.5MB	111
MR	SIEMENS	Magnetom Expert	두부	50	256×256	0.12MB	252
CR	FUJI	FCR9501HQ	흉부	10	2140×1760	8MB	102
DX	LISTEM	DRS	흉부	10	3000×3000	16MB	49

테스트 영상을 평가할 테스트 시스템의 사양은 표 2.3과 같다. 범용으로 널리

사용되는 부품으로 구성된 일반 사무용 컴퓨터를 사용하였으며 출력된 영상을 정성적으로 평가하기 위하여 일반 모니터 및 의료용 모니터를 모두 사용하였다.

표 2.7 테스트 환경

항목	내용
CPU	Pentium 4 2.0GHz
RAM	512MByte
Operating System	Window XP Professional Service Pack 1
HDD	Segate ST380021A, 80G
IDE	Ultra DMA 100MHz
File System	NTFS
Network	100Mbps Ethernet
Color Monitor	Samsung SyncMaster 185T
Monochrome Monitor	EIZO G11
Printer	HP Laserjet 4V

2.3.3 밸리데이션 도구

DICOM 밸리데이션(validation)이란 DICOM 표준을 잘 준수하는지의 여부를 검사하는 것을 의미한다. DICOM 표준이 의료 영상 업계의 표준으로 자리 잡은 후에 수많은 업체에서 DICOM 호환 제품을 생산하고 있다. 그러나 많은 DICOM 생산물들이 DICOM 표준을 제대로 지키지 않고 있으며 그 결과로 임상에서 문제가 되는 경우가 빈번하게 발생한다. 일반적으로 널리 쓰이는 DICOM 밸리데이션 도구들은 다음과 같다.

(1) DICOM Validation Tool (DVT)

Agfa사와 Philips사에서 공동으로 개발한 밸리데이션 전용 소프트웨어로서 가장 많이 사용된다. 윈도우즈 및 유닉스 환경에서 모두 사용할 수 있다. DICOM 데이터 집합, 미디어 파일 등의 구문을 검사할 수 있으며 영상 전송 및 프린트 관련

기능도 검사할 수 있는 등 DICOM 과 관련된 대부분의 항목을 밸리데이션할 수 있다. [10]

(2) CTN

DICOM 툴킷이지만 같이 제공되는 다양한 유틸리티들을 이용하여 구문 검사, 영상 전송, 프린트 등의 검사를 수행할 수 있다. 특히 서버 유틸리티는 실제로 병원에서 영상 획득 서버로 사용되는 경우가 많이 있다. 윈도우즈 및 유닉스 환경에서 모두 사용할 수 있다. 1990년대 후반부터 새 버전이 갱신되지 않아 설치 및 유지보수가 어렵다.[4]

(3) OFFIS DCMTK

CTN과 거의 같은 기능을 제공한다. 그 외에 워크리스트 및 프린트 서버를 제공하여 유용하게 사용될 수 있다. 특히 프린트 서버는 밸리데이션 도구 중 유일하게 결과물을 영상 파일로 저장해 준다. 또한 DICOMScope를 사용하여 영상을 올바르게 출력했는지의 여부를 검사할 수 있다. 비교하고자 하는 대상이 DICOMScope의 영상 출력과 다르게 보인다면 잘못된 출력이라고 할 수 있다.[5]

2.4 설계

툴킷은 객체 지향적 구조로 되어 있는 DICOM 구성 요소들을 C++에서 제공되는 클래스 개념으로 구현하였다. DICOM 서비스 클래스들은 기능별로 객체화되어 기존 툴킷에 비교하여 간단하게 사용될 수 있도록 하였다. 새로운 연결이 발생할 때마다 새로운 쓰레드가 생성되며 쓰레드와의 통신을 위하여 윈도우즈에서 제공되는 메시지 기능을 사용하였다. 툴킷의 기능들은 그림 2.4와 같다.

툴킷은 데이터 모듈부, 영상 모듈부, 네트워크 모듈부로 구성된다. 영상 모듈부는 데이터 모듈부로부터 영상 관련 속성들을 추출하여 영상을 출력하고 처리하며 주석을 관리한다. 데이터 모듈은 정보 객체 클래스를 관리하며 데이터 집합의 인코딩과 디코딩에 관여한다. 네트워크 모듈은 TCP/IP 프로토콜을 구현하여

DICOM 장비와 연결되며 정보 객체를 사용하여 서비스 클래스들을 처리한다. 본 절에서는 각각의 모듈군에 대해 설명한다.

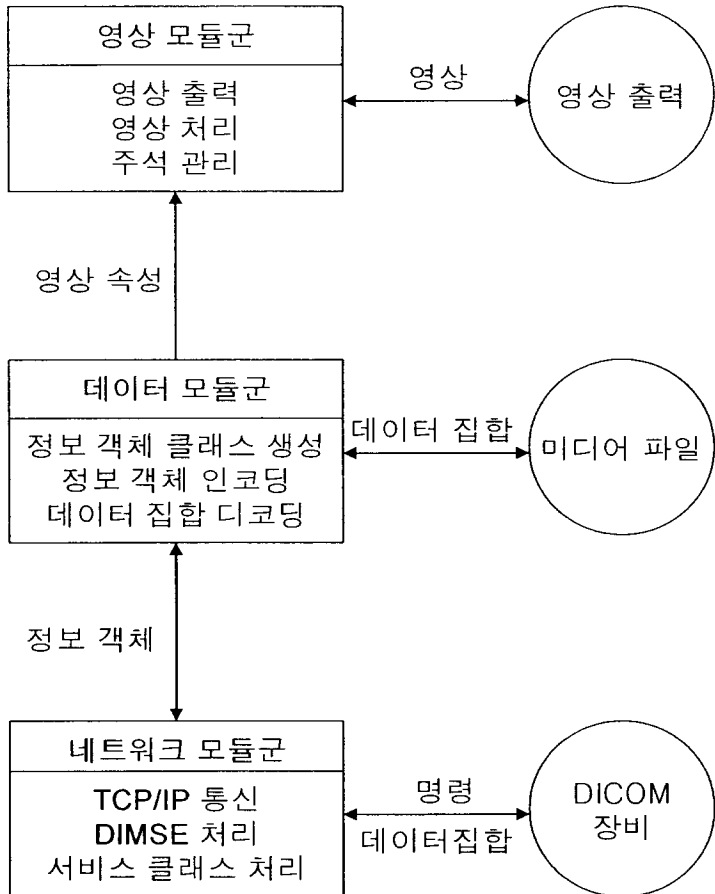


그림 2.4 툴킷의 구성

2.4.1 데이터 모듈군

데이터 모듈군은 DICOM Part 3, 5, 6에 명시된 DICOM 데이터 집합 및 데이터 요소를 처리하는 모듈들로서 데이터 집합 모듈, 데이터 요소 모듈, 데이터 사전 모듈로 구성된다. 그림 2.5는 데이터 집합 모듈군의 관계도 (entity relation diagram, ERD)이다.

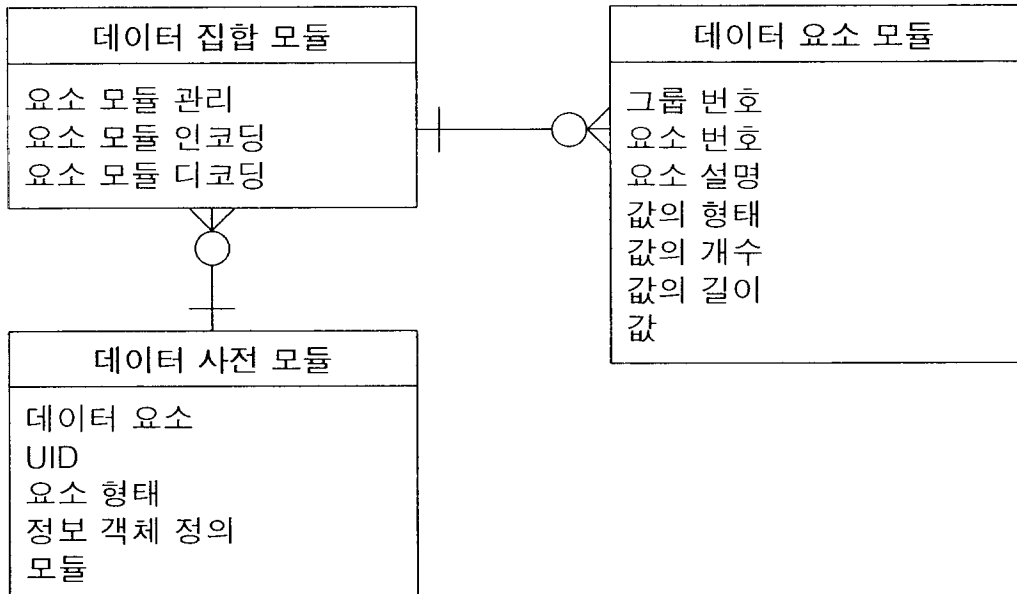


그림 2.5 데이터 집합 모듈군의 ERD

(1) 데이터 요소 모듈

DICOM 3.6에 명시된 데이터 요소의 속성들을 저장한다. 속성들은 그룹 번호와 요소 번호, 요소에 대한 설명, 저장된 값의 표현 방법(value representation, VR), 값의 개수 등이 있다. 하나의 데이터 요소 모듈은 환자의 이름을 가지고 있을 수도 있고 매우 큰 영상을 가지고 있을 수도 있다. 이러한 값을 저장하기 위한 메모리의 생성 및 삭제도 담당한다.

(2) 데이터 집합 모듈

데이터 요소 모듈을 관리하는 모듈로서 데이터 요소 모듈의 생성, 삭제, 변경을 담당하도록 한다. 또한 DICOM 3.5에 명시된 대로 데이터 집합을 파일로 인코딩하거나 파일을 데이터 집합으로 디코딩할 수 있도록 한다. DICOM 데이터 집합 파일을 디코딩 할 때에 파일 매핑(file mapping) 방법을 지원해야 한다. 이는 물리적 메모리를 사용하지 않고 파일을 메모리처럼 사용하는 방법으로 임상에서 대용량의 영상을 조사할 때 반드시 필요한 기능이다.

(3) 데이터 사전 모듈

DICOM 3.6에 열거된 데이터 요소와 UID 자료를 데이터베이스화하여 보관하고 검색 기능을 제공하도록 한다. 자료 구조가 비교적 단순하므로 별도의 데이터베이스를 사용하지 않는다. 데이터 사전 모듈은 툴킷이 사용될 때에 메모리에 상주하며 단 하나의 모듈만 존재하고 다중의 데이터 집합 모듈에서 참조할 수 있도록 한다.

2.4.2 영상 모듈군

영상 모듈군은 영상 출력과 관련된 기능들을 제공한다. 일반 영상 툴킷과 달리 12bit 이상의 그레이스케일(grayscale)을 주로 사용하는 의료 영상에 적합하도록 설계한 것이 가장 큰 특징이며 데이터 집합 모듈로부터 영상 출력과 관련된 자료를 추출하여 올바른 영상을 생성하여 화면에 출력하는 역할을 한다. 영상 모듈군은 그림 2.6와 같이 네 개의 모듈로 되어있다.

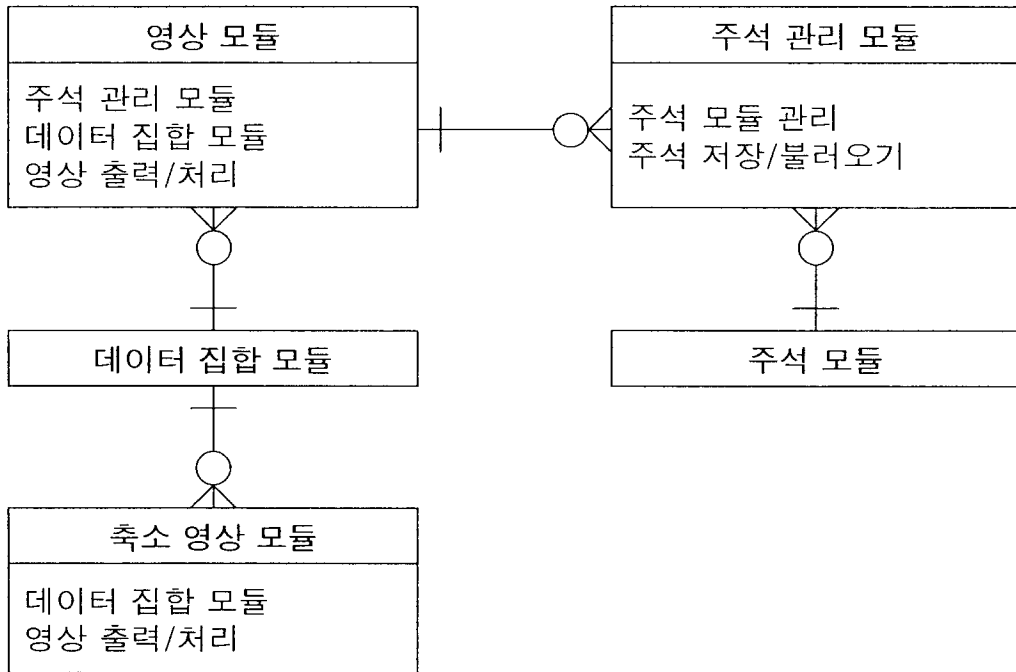


그림 2.6 영상 모듈군의 ERD

(1) 영상 모듈

DICOM 데이터 집합으로부터 표 2.8과 같이 필요한 정보를 획득하여 올바른 영상을 출력한다. 하나의 데이터 집합 모듈에 여러 개의 영상 모듈이 관계할 수 있다. 영상 모듈은 주석 출력을 위하여 하나 이상의 주석 관리 모듈을 가진다. 영상 모듈은 데이터 집합 모듈과 마찬가지로 메모리 사용을 극소화시키기 위하여 영상 출력 및 처리에 필요한 메모리를 할당할 때 임시 파일을 매핑하여 사용한다. 화면에 출력에 사용되는 메모리는 필요시에만 할당하여 사용하므로 역시 메모리가 누적되지 않아 속도가 저하되지 않도록 한다. 영상 모듈은 아래와 같이 임상에서 필요한 대부분의 기능을 수행할 수 있도록 한다.

표 2.8 영상 출력을 위해 데이터 집합으로부터 획득한 파라미터

Tag	Description	Tag	Description
0x0028,0x0002	Sample Per Pixel	0x0028,0x1053	Rescale Slope
0x0028,0x0004	Photometric Interpretation	0x0028,0x1201	Palette Red
0x0028,0x0010	Row	0x0028,0x1202	Palette Green
0x0028,0x0011	Col	0x0028,0x1203	Palette Blue
0x0028,0x0100	Bits Allocated	0x0028,0x0008	Multiframe
0x0028,0x0101	Bits Stored	0x0018,0x1063	Frame time
0x0028,0x0103	Pixel Representation	0x0028,0x0030	Pixel Spacing
0x0028,0x0006	Planar Configuration	0x0020,0x0037	Image Orientation
0x0028,0x0106	Smallest Pixel	0x5000,0x0010	Curve Length
0x0028,0x0107	Largest Pixel	0x5000,0x3000	Curve Data
0x0028,0x0120	Pixel Padding	0x0018,0x1600	Shutter
0x0028,0x1051	Window Width	0x0018,0x1610	Shutter Center
0x0028,0x1050	Window Center	0x0018,0x1612	Shutter Radius
0x0028,0x1052	Rescale Intercept		

① 영상 출력

- 그레이스케일 변환

의료 영상은 대부분 12bit 이상의 그레이스케일 영상이나 일반적으로 사용되는 모니터나 프린터 등은 대부분 8bit만을 지원하므로 변환과정이 필요하다. 그레이스케일 영상이 출력되기 전에는 두 가지 변환 과정이 적용된다. 첫 번째 변환은 모달리티 변환이다. DICOM 영상에 저장된 픽셀 데이터는 장비 제조사에 의존적일 수 있다. 이 경우 제조사에 의존적인 픽셀 값은 제조사에 독립적인 픽셀 값으로 바뀌어야 한다. 선형 변환에는 Rescale Intercept (0028, 1052), Rescale Slope (0028, 1053) 데이터 요소가 식 1과 같이 사용된다. [11]

$$\text{New Pixel} = \text{Rescale Slope} \times \text{Old Pixel} + \text{Rescale Intercept} \quad (1)$$

두 번째 변환은 관심 영역(value of interest, VOI) 변환이다. 일반적으로 윈도우 조정으로 잘 알려져 있다. Window center (0028, 1050), Window width (0028, 1051) 데이터 요소를 사용하여 식 (2)를 적용한다.[11]

$$\begin{aligned} \text{Low} &= \text{Window Center} - \text{Window Width} / 2 \\ \text{New Pixel} &= (\text{Old Pixel} - \text{Low}) / \text{Window Width} \times 255 \end{aligned} \quad (2)$$

- 보간

영상을 확대할 경우 픽셀의 각진 현상이 나타나게 된다. 이런 현상을 제거하고 양질의 영상을 제조하기 위하여 보간 기능을 지원해야 한다. 툴킷에서는 윈도우에서 비트맵을 출력할 때 제공되는 보간 기능을 사용한다.

② 영상 조작 기능

윈도우 조정, 확대, 이동, 부분 확대, 반전 등의 기능을 제공하도록 한다. 영상의 원본은 바뀌지 않고 출력되는 형태만 변화되도록 하는 기능으로서 임상에서 매우 빈번하게 사용되는 기능이므로 마우스로 편리하고 빠르게 조작할 수 있도록 설계되어야 한다.

③ 영상 처리 기능

흐림 처리, 선명 처리, 외곽선 검출, 회전, 대칭 등의 기능을 제공하도록 한다. 진단에 도움을 주기 위한 기능으로서 영상 처리된 이후에 영상 조작이나 영상 처리에 영향을 주는 기능들이다. 그러므로 반드시 백업 기능을 지원하여 이전 상태로 돌아갈 수 있어야 한다.

④ 오버레이

화면 오버레이 요소는 영상의 정보를 출력하는 텍스트, 영상의 실제 크기를 보

여 주는 자, 사용자가 추가할 수 있는 주석, 모니터를 조정하는데 참조할 수 있는 그레이스케일, 영상의 위치를 알려 주는 영상 위치, 심전도 등을 보여주는 신호 등을 출력할 수 있도록 한다.

⑤ 동영상

데이터 집합이 동영상을 포함할 경우에는 재생, 멈춤, 앞으로, 뒤로 등과 같은 다양한 동영상 모드를 제공하도록 한다. 시간 간격을 조정할 수도 있어야하며 동영상에 심전도와 같은 1차원 신호가 포함되어 있을 경우 이를 같이 출력할 수 있어야 한다.

⑥ 윈도우즈 프린터 지원

윈도우즈에 설정되어 있는 프린터를 사용하여 영상을 출력할 수 있는 기능을 제공한다.

⑦ 압축

DICOM에서 사용되는 무손실 압축 코덱은 대부분 상용이므로 별도로 구현해야 하지만 손실 압축 코덱은 공개되어 있으므로 JPEG과 RLE 두 가지 압축은 기본적으로 제공하도록 한다.

⑧ 기타 기능

일반적으로 널리 사용되는 영상 포맷인 윈도우즈 비트맵, JPEG 형태로 변환하여 저장할 수 있도록 하며 윈도우의 클립보드로 복사가 가능하도록 한다.

(2) 주석 관리 모듈

주석 모듈들의 생성, 삭제, 변경을 관리하는 모듈이다. 또한 주석의 저장 및 불러오기에 관련된 기능도 수행한다. 주석은 별도의 파일로 관리되도록 한다.

(3) 주석 모듈

주석 객체이다. 그 종류에는 선, 네모, 원, 텍스트, 자유 그리기, 거리, 각도가 있다. 주석 객체는 영상 위에 오버레이되어 출력되도록 한다. 네모, 원, 자유 그리기 객체에서는 넓이, 높이, 면적, 기술 통계량을 구할 수 있는 기능을 제공하도록 한다.

(4) 축소 영상 모듈

영상 모듈에서 파생된 모듈로서 영상 모듈을 원하는 크기로 축소하여 새로운 영상 모듈을 생성한다. 영상의 크기도 축소되므로 출력 속도가 빠르다. 미리 보기, 실시간 영상 처리 등에 사용한다.

2.4.3 네트워크 모듈군

툴킷에서는 DICOM Part 7, 8에서 권고한 TCP/IP를 이용하여 네트워크와 관련된 작업을 처리한다. TCP/IP를 이용하기 위하여 MFC에서 제공되는 소켓 관련 클래스들을 사용하며 그림 2.7과 같이 설계하였다.

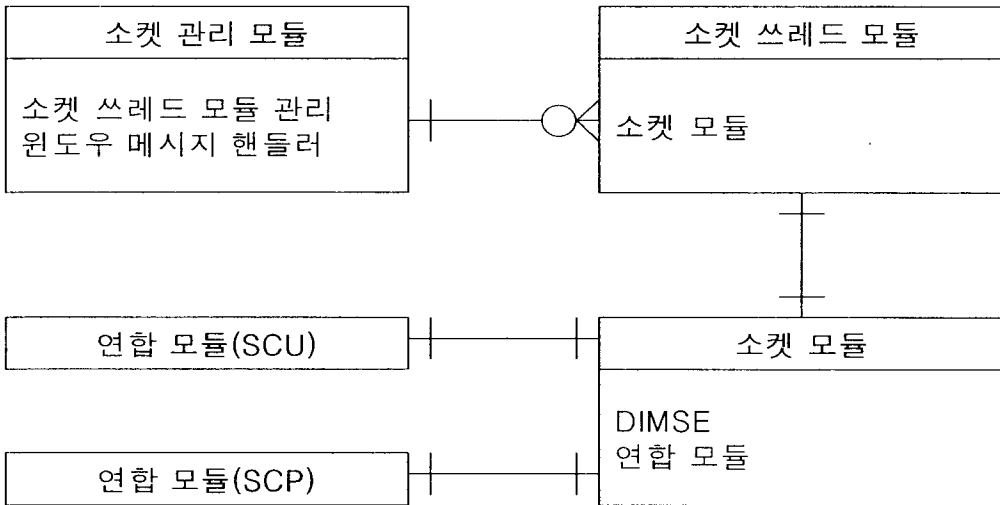


그림 2.7 네트워크 모듈군의 ERD

하나의 소켓 모듈은 항상 유일한 쓰레드에서 실행된다. 이때 사용되는 쓰레드는 소켓 쓰레드 모듈이며 이러한 쓰레드 모듈들을 관리하는 것이 소켓 쓰레드 모듈이다. 소켓 모듈은 항상 두개의 연합 모듈을 갖는다. 모듈들 간의 상호 작용에 의하여 메시지를 교환하는 예는 그림 2.8과 같다.

응용 프로그램 쓰레드 영역	소켓 쓰레드 영역
Socket Thread 모듈 생성	시작됨
작업 1 추가 지시	작업 1 추가
작업 2 추가 지시	작업 2 추가
작업 3 추가 지시	작업 3 추가
작업 시작 지시	소켓 연결
	Send Associate Request
	Receive Associate Accept
	작업 1 메시지 교환
	작업 2 메시지 교환
	작업 3 메시지 교환
	Send Associate Release
	Receive Associate Release
	소켓 종료
쓰레드 파괴	쓰레드 파괴 요청

그림 2.8 메시지 교환 절차의 예

(1) 소켓 관리 모듈

MFC의 CWnd에서 파생된 클래스로 소켓 쓰레드 모듈의 생성 및 파괴를 관리한다. 소켓 쓰레드 모듈은 개발자의 응용 프로그램과 다른 쓰레드에 존재하므로 개발자가 접근할 수 없다. 그러므로 개발자의 응용 프로그램과 동일한 쓰레드에 있는 소켓 관리 모듈을 통하여 정보를 획득한다. 즉 소켓 쓰레드 모듈은 윈도우

메시지를 통하여 소켓 관리 모듈에 정보를 전달하고 사용자는 소켓 관리 모듈의 메시지 처리 루틴을 이용하여 작업을 수행한다.

(2) 소켓 쓰레드 모듈

MFC의 CWinTread에서 파생된 모듈로서 하나의 소켓 모듈이 실행될 쓰레드 환경을 제공한다. 소켓의 생성, 삭제, 연결, 종료에 관련된 작업을 수행한다.

(3) 소켓 모듈

CSocket에서 파생된 모듈이다. 소켓 모듈은 SCU 및 SCP로 행동할 수 있도록 두 개의 연합 모듈을 갖는다. 소켓 모듈에서 발생하는 모든 이벤트들과 정보들은 윈도우 메시지를 통하여 소켓 관리 모듈로 전송된다.

(4) 연합 모듈

DICOM 장비 간의 연합을 담당하는 모듈이다. 그림 2.9은 연합 요청 과정을 보여주고 있다.[1] SCU는 연합 요청 데이터를 SCP로 보낸다. SCP는 SCU로부터 받은 데이터를 분석하여 연결 여부를 결정한다. SCP는 연결이 가능할 경우 연합 응답 데이터를 SCU로 보낸다. SCU는 SCP로부터 데이터를 받아 연합 여부를 확인한다.

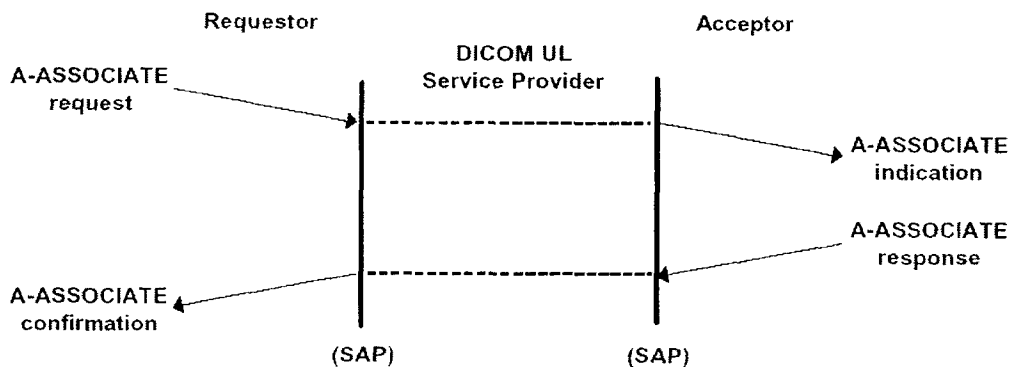


그림 2.9 연합 과정

(5) DICOM 프린트 모듈

DICOM 프린터를 사용하는 과정은 매우 복잡한 작업이다. 기존의 툴킷을 사용할 경우 툴킷을 사용하는 개발자가 그림 2.10과 같이 여러 단계의 메시지 교환과정을 복잡하게 구현해야 한다. 제안된 툴킷에서는 이러한 복잡성을 없애고 간단하게 구현할 수 있는 기능을 제공한다.

```
A-ASSOCIATE // 프린터와 연합을 수행한다.
N-GET // 프린터의 정보를 받는다.
N-CREATE // Film session SOP 객체를 생성한다.
for (각각의 film session 객체에 대하여 반복한다.)
{
    N-CREATE // Film box SOP 객체를 생성한다.
    for (각각의 image box에 대하여 반복한다.)
    {
        N-SET // 출력할 image box를 설정한다.
    }
}
if (성공적이면)
{
    N-ACTION // Film session SOP 객체의 출력을 지시한다.
    N-DELETE // Film session SOP 객체를 삭제한다.
}
N-EVENT-REPORT // 프린트 결과를 보고 받는다.
A-RELEASE // 프린터와 연합 해제를 수행한다.
```

그림 2.10 DICOM Print 과정

제 3 장 결과

제안된 툴킷은 1999년 초기 버전이 완성된 이후에 많은 수정 과정을 거쳐 2003년 현재 버전에 이르렀다. 기본적으로 벨리데이션 도구에 의한 시뮬레이션 및 검증 과정을 거쳤으며 다양한 모달리티에서 출력된 영상들을 테스트 하였고 여러 PACS 제조사들의 서버 및 뷰어 제품들과의 연결 여부를 검증하였다. 구현된 툴킷의 각 모듈별 기능 및 성능을 살펴보면 다음과 같다.

3.1 데이터 모듈

모든 DICOM 관련 작업의 기본 단위는 DICOM 데이터 집합이다. 데이터 집합 모듈의 성능은 데이터 집합의 인코딩/디코딩 여부와 속도이다. 구현된 각각의 모듈들을 살펴보면 다음과 같다.

3.1.1 데이터 사전 모듈

DICOM 데이터 집합은 데이터 요소들로 구성되어 있으며 이것을 처리하기 위해서는 각종 정보가 필요하다. 제안된 툴킷에서는 DICOM Part 3에 명시된 정보 객체 정의 및 모듈 정보들, DICOM Part 5에 명시된 VR 정보들, DICOM Part 6에 명시된 데이터 요소, UID 정보들을 표 3.1과 같은 포맷을 가지는 파일 형태로 저장하고 이를 조회하는 모듈을 구현하였다. 또한 파일을 편집할 수 있는 편집 도구를 그림 3.1과 같이 구현하였다. 편집 도구는 스프레드시트 기능을 제공하여 마이크로소프트 엑셀 등과 같은 응용 프로그램에서 편집된 내용을 복사하여 사용할 수 있다. 데이터 사전 모듈은 빈번히 사용되므로 물리적 메모리에 상주하도록 되어 있으며 툴킷이 첫 번째 데이터 집합 모듈을 최초로 생성할 때 자동으로 생성된다. 데이터 사전 모듈은 하나의 응용 프로그램에 단 하나만 존재한다.

표 3.1 데이터 사전의 파일 포맷

Section	Item	Type	Size	Description
Header	nModules	int	4	Number of Modules
	nIODs	int	4	Number of IODs
	nVRs	int	4	Number of VRs
	nAttributes	int	4	Number of Attributes
	nUIDs	int	4	Number of UIDs
	nWindows	int	4	Number of Windows
Module Data	strName	char	40	Module Name
	nAttributes	int	4	Number of Attributes
	Repeat Part			
	Tag	int	4	Element Tag
IOD Data	strName	char	40	IOD Name
	nModules	int	4	Number of Modules
	nModuleID	int	nModules * 4	Module Code Array
VR Data	strVR	char	4	VR
	strName	char	40	VR Name
	nLength	short	2	Length Of VR
	strRestrict	char	3	Restriction
	nUnitSize	short	2	Unit Size
Attribute Data	nGroup	short	2	Attribute Group
	nElement	short	2	Attribute Element
	nMaskGroup	short	2	Mask Group
	nMaskElement	short	2	Mask Element
	strName	char	48	Attribute Name
	nCode	char	4	VR
	nMinVM	int	4	Min VM
	nMaxVM	int	4	Max VM
UID Data	strUID	char	35	UID
	strName	char	80	UID Discription
	strUIDType	char	50	UID Type
Window Data	strName	char	80	Window Name
	nCTWidth	int	4	CT Width
	nCTCenter	int	4	CT Center
	nMRWidth	int	4	MR Width
	nMRHeight	int	4	MR Center

	1	2	3	4	5	6	7	8	9	10	11
1 Acquisition Context	0040:0055	0040:0043	0040:0136	0040:0030	0040:009a	0040:0121	0040:0122	0040:0123	0040:0150	0040:0158	0040:0556
2 Audio	5000:2000	5000:2002	5000:2004	5000:2006	5000:2008	5000:200a	5000:200c	0008:1140	0008:1150	0008:1155	5000:200e
3 CR Image	0028:0004	0018:0060	0018:1004	0018:1110	0018:1111	0018:1150	0018:1151	0018:1152	0018:1153	0018:1164	0018:1170
4 CR Series	0018:0015	0018:5101	0018:1150	0018:1180	0018:1190	0018:1260	0018:1261				
5 CT Image	0008:0008	0028:0002	0028:0004	0028:0100	0028:0101	0028:0102	0028:1052	0028:1053	0018:0060	0020:0012	0018:0022
6 Cine	0018:1244	0018:1063	0018:1065	0008:2142	0008:2143	0008:2144	0018:0040	0018:1066	0018:1067	0018:0072	0018:1242
7 Contrast/bolus	0018:0010	0018:0012	0018:1040	0018:0014	0018:1041	0018:1042	0018:1043	0018:1044	0018:1045	0018:1047	0018:1048
8 Curve	5000:0005	5000:0010	5000:0020	5000:0103	5000:3000	5000:0022	5000:0030	5000:0040	5000:0104	5000:0105	5000:0106
9 Curve Identification	0020:0024	0008:0025	0008:0035	0008:1140	0008:1150	0008:1155	0008:1130	0008:1150	0008:1155	0008:1145	0008:1150
10 DX Anatomy Imaged	0028:0062	0006:2218	0006:2220	0006:2228	0006:2230						
11 DX Detector	0018:7004	0018:7005	0018:7006	0018:7008	0018:700a	0018:700c	0018:700e	0018:7010	0018:7011	0018:7012	0018:7014
12 DX Image	0006:0006	0028:0002	0028:0004	0028:0100	0028:0101	0028:0102	0028:0103	0028:1040	0028:1041	0028:1052	0028:1053
13 DX Positioning	0018:5140	0018:5100	0018:5101	0054:0220	0054:0222	0054:0410	0054:0412	0054:0414	0018:1111	0018:1110	0018:1114
14 DX Series	0008:0060	0008:1111	0008:1150	0008:1155	0006:0068						
15 Device	0050:0010	0050:0014	0050:0016	0050:0017	0050:0018	0050:0019	0050:0020				
16 Display Shutter	0018:1600	0018:1602	0018:1604	0018:1606	0018:1608	0018:1610	0018:1612	0018:1620	0018:1622		
17 Frame Pointers	0028:6010	0028:6020	0028:6022								
18 Frame of Reference	0020:0052	0020:1040									
19 General Equipment	0006:0070	0006:0090	0006:0081	0006:1010	0006:1040	0006:1090	0018:1000	0018:1020	0018:1050	0018:1200	0018:1201
20 General Image	0020:0013	0020:0020	0008:0023	0008:0033	0008:0006	0020:0012	0008:0022	0008:0032	0006:1140	0008:2111	0008:2112
21 General Series	0008:0060	0020:000a	0020:0011	0020:0060	0008:0021	0008:0031	0008:1050	0018:1030	0008:103a	0008:1070	0008:1111
22 General Study	0020:0004	0008:0020	0008:0030	0008:0090	0020:0010	0008:0050	0008:1030	0008:1048	0008:1060	0008:1110	0008:1032
23 Image Histogram	0060:3000	0060:3002	0060:3004	0060:3006	0060:3008	0060:3010	0060:3020				
24 Image Pixel	0028:0002	0028:0004	0028:0010	0028:0011	0028:0100	0028:0101	0028:0102	0028:0103	7e0:0010	0028:0006	0028:0034
25 Image Plane	0028:0030	0020:0037	0020:0032	0018:0050	0020:1041						
26 MR Image	0008:0008	0028:0002	0028:0004	0028:0100	0018:0020	0018:0021	0018:0022	0018:0023	0018:0080	0018:0081	0018:0091
27 Mask	0028:6100	0028:6101	0028:6102	0028:6110	0028:6112	0028:6114	0028:6120	0028:6190	0028:1030		
28 Modality LUT	0028:3000	0028:3002	0028:3003	0028:3004	0028:3006	0028:1052	0028:1053	0028:1054			
29 Multi-Frame	0028:0008	0028:0009									
30 Multi-Frame Overlay	0060:0015	0060:0051									
31 NM Detector	0054:0022	0018:1180	0018:1181	0018:1147	0018:1149	0018:1182	0018:1183	0018:1184	0028:0032	0028:0031	0018:1145
32 NM Image	0008:0008	0054:0400	0028:2110	0018:0070	0018:0071	0018:1130	0018:1131	0018:1242	0018:1243	0018:5020	0028:0051
33 NM Image Pixel	0028:0002	0028:0004	0028:0100	0028:0101	0028:0102	0028:0030					
34 NM Isotope	0054:0012	0054:0018	0054:0013	0054:0014	0054:0015	0054:0016	0054:0300	0018:1070	0054:0302	0018:1071	0018:1072
35 NM Multi-Frame	0028:0009	0054:0010	0054:0011	0054:0020	0054:0021	0054:0030	0054:0031	0054:0050	0054:0051	0054:0060	0054:0061
36 NM Multi-gated Acqui...	0018:1080	0018:1085	0018:1086	0018:1088	0054:0062	0018:1060	0018:1064	0054:0063	0018:1063	0018:1062	0018:1081

그림 3.1 데이터 사전 파일 편집기

3.1.2 디코딩

제안된 툴킷은 2장에서 언급된 테스트 영상뿐만 아니라 임상에서 발생하는 대부분의 영상에 대하여 벨리데이션 도구들과 동일한 결과를 보여주었다. 디코딩 속도는 임상에서 사용하기에 충분하였으며 파일 맵을 사용하여 메모리를 사용하지 않으므로 일반 PC에서도 대량의 영상을 처리할 수 있었다. DICOM 데이터 집합을 디코딩하는 예제 코드는 그림 3.2와 같다.

```

// 데이터 집합 모듈을 생성한다.
KDicomDS * pDS = new KDicomDS;
// 데이터 집합 파일을 파일맵을 사용하여 디코딩한다.
pDS->LoadDS(filename, FILEMAP);
// 환자 ID 정보가 있는 데이터 요소 모듈의 포인터를 얻는다.
KDicomElement * pDE;
pDE = pDS->GetElement(0x0010, 0x0020);
// 데이터 요소로부터 값을 얻는다.
CString PatientID = pDE->GetValue(0);

```

그림 3.2 데이터 집합 디코딩 코드의 예

테스트 영상들에 대한 디코딩 속도를 측정한 결과는 표 3.2와 같다. 속도를 측정할 때 CPU가 캐시를 이용하는 것을 피하기 위하여 매번마다 PC를 재시동하고 일정시간이 지나 CPU의 점유율이 5%미만일 때에 측정을 실시하였다. 파일 매핑 방법은 영상 파일 자체를 읽기 전용의 메모리로 매핑하고 데이터 요소들의 값의 위치만을 읽으므로 메모리를 할당하고 파일에서 읽어 복사하는 기존의 방법보다 매우 빠른 성능을 보였다. 또한 파일의 크기와 소요시간이 비례하는 것을 알 수 있다.

표 3.2 데이터 집합의 디코딩 소요 시간 (단위: ms/dataset)

Dataset Decoding	file map	physical memory
MR	11.6±0.2	14.7±0.1
CT	14.9±0.1	28.1±0.6
CR	18.2±1.3	267.4±12
DX	22.6±3.3	570.9±24.4

3.1.2 인코딩

DICOM 데이터 집합의 인코딩은 영상을 인코딩해야 할 경우와 메시지를 인코딩해야 할 경우 두 가지가 있다. 영상 인코딩은 영상을 획득한 후에 이를 DICOM 형태로 저장해야 하는 모달리티와 기존의 DICOM 영상을 압축해야하는 서버측에서 주로 사용되며 처리 속도나 용량에 크게 제한을 받지 않는다. 메시지 인코딩은 DICOM 메시지를 교환하는 모든 DICOM 응용 프로그램에서 사용된다. 두 인코딩 모두 상대방으로 전송되어야 하므로 정확한 문법이 요구된다. 제안된 툴킷은 다양한 영상 인코딩과 메시지 인코딩에 대하여 DVT를 통하여 엄격하게 검증하였다. 영상과 메시지를 인코딩하는 예는 각각 그림 3.3, 3.4와 같다.

```

// 데이터 집합 모듈을 생성한다.
KDicomDS * pDS = new KDicomDS;
// 모듈에 CT 영상 IOD의 데이터 요소 모듈들을 추가한다.
pDS->NewDS(CT_IMAGE_STORAGE);
// 환자 ID 정보가 있는 데이터 요소 모듈의 포인터를 얻는다.
KDicomElement * pDE;
pDE = pDS->GetElement(0x0010, 0x0020);
// 데이터 요소 모듈에 값을 추가한다.
pDE->AddValueLO("10000");
...
// 필요한 모든 데이터 요소 모듈에 값을 추가한다.
...
// Implicit Little Transfer로 인코딩하여 저장한다.
pDS->SaveDS(filename, IMPLICIT_LITTLE, TRUE);

```

그림 3.3 영상 인코딩 코드의 예


```

// 데이터 집합 모듈을 생성한다
KDicomDS * pDS = new KDicomDS;
// 모듈에 데이터 요소 들을 추가할 수 있도록 한다.
pDS->NewDS(UNKNOWN_CLASS);
// 데이터 요소 모듈을 추가한다.
KDicomElement * pDE;
pDE = pDS->AddElement(0x0000, 0x0002);
// 데이터 요소 모듈에 값을 추가한다.
pDE->AddValueUI("1.2.840.10008.1.1");
...
// 원하는 모듈들을 추가한다.
...
// Implicit Little Transfer로 인코딩하여 저장한다.
pDS->SaveDS(filename, IMPLICIT_LITTLE, FALSE);

```

그림 3.4 메시지 인코딩 코드의 예

3.2 영상 모듈

영상의 올바른 출력 여부를 평가하기 위하여 DICOMScope와 상용 뷰어들을 사용하였다. DICOMScope는 영상 밸리데이션 도구로 널리 이용되나 데이터 집합의 문법이 정확하지 않을 경우와 동영상일 경우, 압축 영상일 경우 등 부적절한 DICOM 데이터 집합에 대하여 영상 출력을 거부한다. 그러나 임상에서 사용되는 상용 뷰어들은 대부분의 영상들을 출력하였다. 그러므로 DICOMScope와 추가적으로 2002년 IHE 인증을 획득한 Infinite사의 PiView 및 Marotech사의 M-View를 사용하였다. [12][13]

테스트 영상들을 제안된 툴킷으로 출력하고 그 결과를 뷰어들의 출력물과 비교하여 동일함을 검증하였다. 디코딩된 테스트 셋들을 파일맵 및 물리적 메모리를 이용하여 출력하는데 소요된 시간은 표 3.3과 같다. 또한 디코딩과 영상출력에 소요된 시간을 합산, 테스트 셋들을 모두 출력하는데 소요된 시간, 사용된 메모리를

상용 툴킷과 비교한 결과는 표 3.4와 같다. 파일맵만을 사용할 때와 물리적 메모리만을 사용할 때를 비교할 때 소요된 시간은 거의 차이가 없었으나 파일맵만을 사용할 때에 메모리가 거의 사용되지 않는 장점을 보였다. 그러나 동일한 영상을 반복하여 사용하는 경우에는 물리적 메모리를 사용할 때가 더 유리할 것이다. 영상을 출력하는 코드는 그림 3.5와 같다.

표 3.3 영상 출력 시간 (단위 : ms/image)

Dataset Decoding	filemap		physical memory	
Image Loading	filemap	physical memory	filemap	physical memory
MR	16.7±0.3	13.0±0.5	3.6±0.1	2.1±0.2
CT	32.5±0.5	28.3±0.3	7.5±0.2	5.5±0.1
CR	266.6±2	252.4±1.3	56±1.9	43.1±1.8
DX	490.8±27.1	456.2±3.3	64.5±0.6	52.5±1.3

표 3.4 디코딩 및 영상 출력의 종합

측정 항목 디코딩/영상출력	영상별 소요시간 (단위 : ms/image)				테스트 셋 전체 출력 소요 시간 (단위:sec)	메모리 사용 (단위:MB)
	MR	CT	CR	DX		
파일맵/파일맵	28.3	47.4	284.8	513.4	11.8	3
파일맵/메모리	24.6	43.3	270.6	478.8	10.8	124
메모리/파일맵	18.2	35.6	323.4	635.4	12.3	248
메모리/메모리	16.7	33.6	310.5	623.4	11.9	370
L사의 상용툴킷	31.5	47.3	353.6	692.8	14.4	256

```
// 데이터 집합 모듈을 생성한다.  
KDicomDS * pDS = new KDicomDS;  
// 데이터 집합 파일을 파일맵을 사용하여 디코딩한다.  
pDS->LoadDS(filename, FILEMAP);  
// 영상 모듈을 생성한다.  
KDicomImage * pImage = new KDicomImage;  
// 데이터 집합 모듈로부터 영상을 구성한다.  
pImage->LoadImage(this, pDS, FILEMAP);  
// 영상의 위치와 크기를 지정한다.  
pImage->MoveImage(0,0,512,512);
```

그림 3.5 영상 출력 코드의 예

3.2.1 영상 출력 관련 기능

(1) 문자 오버레이(overlay)

툴킷에 의해 출력된 영상 및 문자 오버레이는 그림 3.6과 같다. 문자 오버레이는 영상과 함께 반드시 표시되어야 하는 항목으로서 일반적으로 병원에서 요구되는 항목들을 툴킷에서 자동으로 설정하여 출력해 주도록 하였다. 개발자는 이 기능을 사용하지 않고 나름대로의 오버레이를 설정할 수 있도록 하였다.

좌측 상단에는 환자 ID, 생년월일, 성별, 나이, 환자 이름, 영상 번호, 모달리티 관련 내용들을 표시한다. 우측 상단에는 기관명, 제조사명, 시리즈(series) 번호, 촬영 날짜, 시간을 표시한다. 우측하단에는 마우스 포인터의 위치 및 영상의 값, 윈도우 중심 및 폭, 영상의 행와 열이 표시된다.

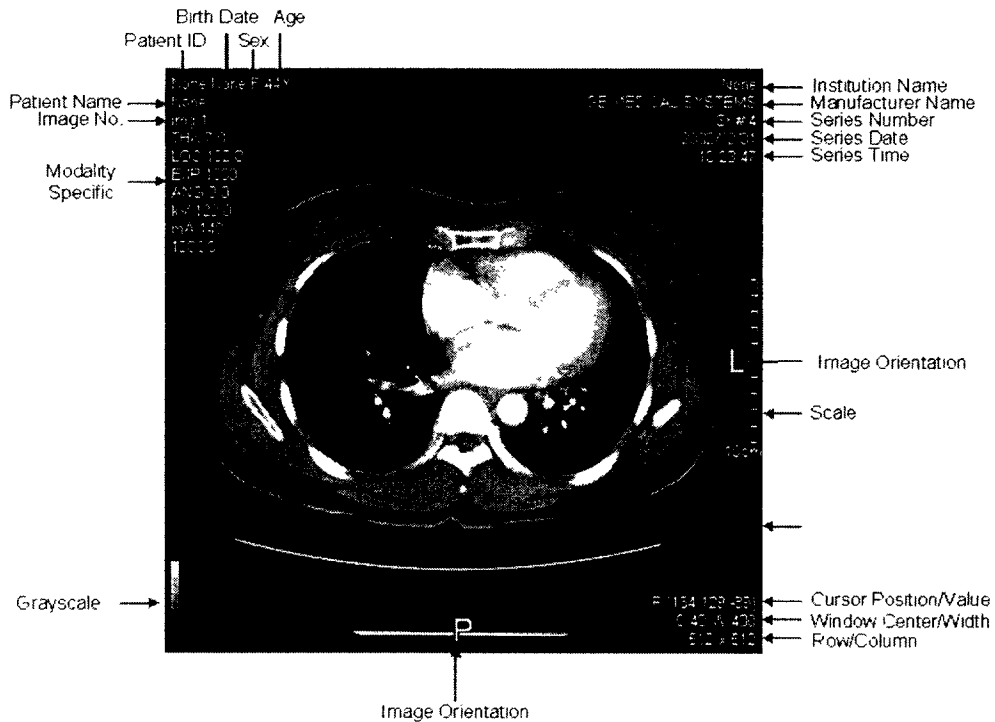


그림 3.6 오버레이

(2) 영상 방위

CT, MR 등과 같은 단면 영상은 3차원적인 구조를 가지고 있으며 DICOM 데이터 집합의 image orientation (0020, 0037) 데이터 요소는 그림 3.7과 같은 좌표계에서 각 축들과 영상의 첫 번째 행, 열의 cosine 값을 가지고 있다. Cosine 값들로부터 영상과 각 축들과의 각도를 알 수 있으므로 영상의 위치를 파악할 수 있다. 영상의 우측 및 하단에 영상의 위치를 약어로 표시하는 방법은 CT 및 MR 관련 장비에서 오랫동안 관습적으로 사용되어진 방법이다. 영상을 회전하거나 대칭할 때 영상의 위치 표시는 틀킷에 의하여 자동으로 변환된다.[1]

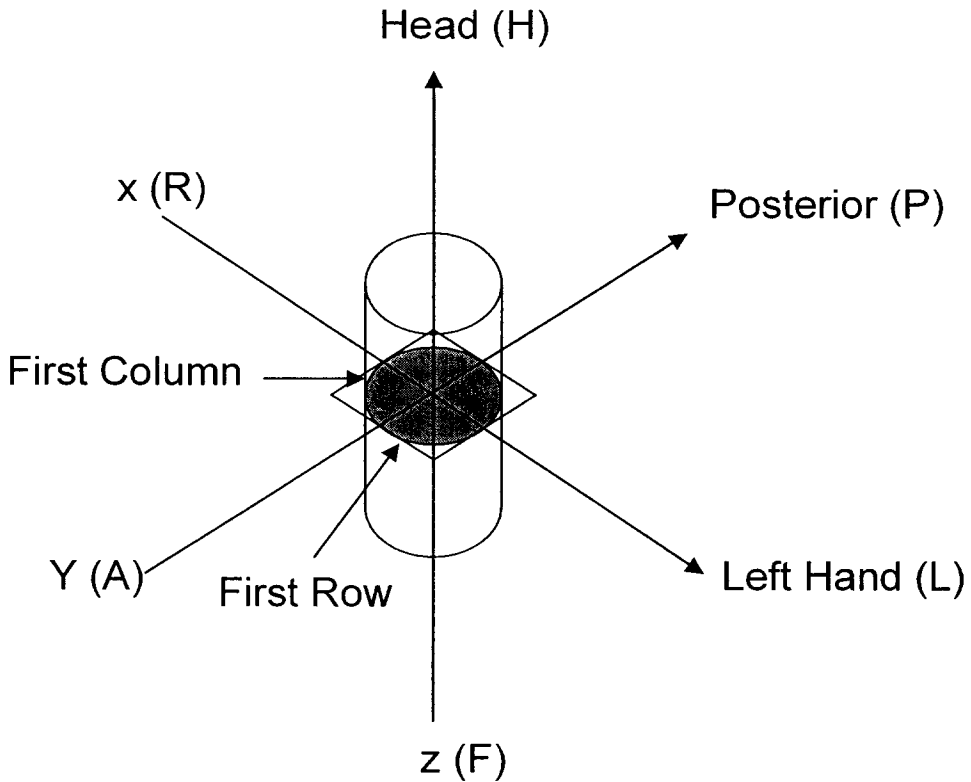


그림 3.7 영상 방위

(3) 자

CT, MR 등과 같은 장비들은 pixel spacing (0028,0030) 데이터 요소에 픽셀간

의 실제 거리를 기재하며 이것을 이용하여 자를 표시할 수 있다. 제안된 툴킷에서는 영상의 우측에 10개의 눈금을 가지는 자를 표시하였다. 영상에 Pixel Spacing 데이터 요소가 있을 때에는 10cm를 나타내며 그 값이 없을 때에는 100 픽셀을 표시하도록 하였다.

(4) 보간

영상의 크기가 매우 작거나 고배율로 확대 되었을 경우 그림 3.8과 같이 각진 픽셀 현상이 나타난다. 제안된 툴킷에서는 윈도우즈 API (application programming interface)에서 제공되는 SetStretchBltMode 함수에 HALFTONE 인자를 사용하여 보간을 구현하였다. 그림 3.9는 보간된 영상이다.

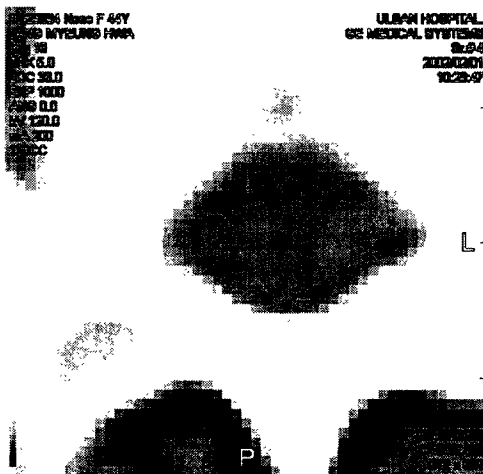


그림 3.8 원영상

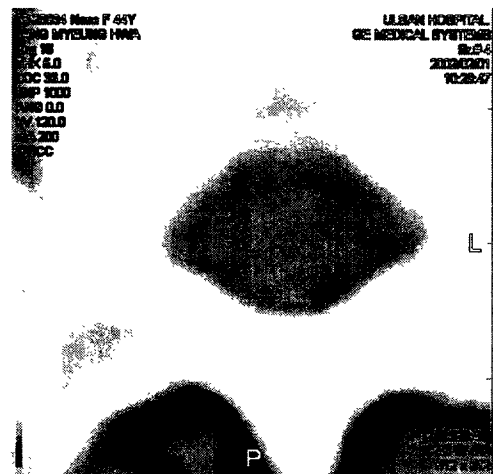


그림 3.9 보간이 적용된 영상

(3) 확대

마우스 왼쪽 버튼을 누른 상태에서 좌우로 이동하여 영상을 확대 축소할 수 있다. 신속한 영상 출력을 위하여 마우스 버튼을 클릭할 때 생성된 장치 의존 비트맵 (Device Dependent Bitmap, DDB)를 사용하므로 실시간 확대 축소 중에는 보간 기능을 사용할 수 없다. 그러나 마우스 버튼을 놓아 확대 축소 작업을 종료

하면 영상을 보간하여 출력한다. 그림 3.10은 확대 기능으로 확대된 영상이다.

(4) 이동

마우스 왼쪽 버튼을 누른 상태에서 좌우로 이동하여 영상의 위치를 움직일 수 있다. DDB를 사용하므로 실시간으로 움직일 때에는 확대 기능과 마찬가지로 보간을 사용할 수 없다. 그림 3.11은 이동 기능으로 이동한 영상이다.



그림 3.10 확대

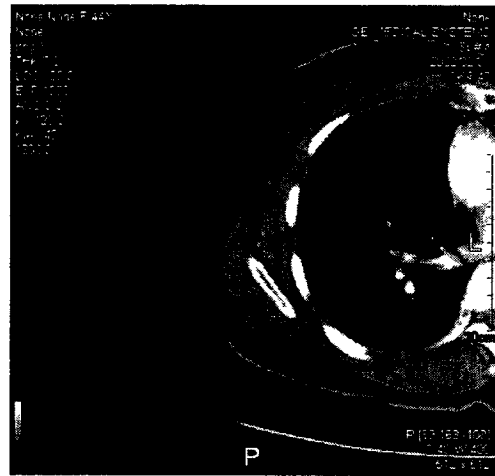


그림 3.11 이동

(5) 윈도우 중심, 폭 조정

대부분의 상용 뷰어에서 윈도우를 조정하는데 사용되는 방법을 채택하였다. 즉 마우스의 오른쪽 버튼을 누른 상태에서 좌우 및 상하로 이동하여 영상의 윈도우 중심 및 폭을 실시간으로 조정한다.

윈도우를 조정하는 작업은 영상의 크기만큼 계산량이 발생하여 시간이 소모되므로 컴퓨터의 속도와 영상의 크기에 매우 의존적이다. 일반적으로 CPU가 Pentium 2 이상인 PC에서 영상의 크기가 512x512인 영상을 실시간으로 처리할 수 있었다. 그 이상의 크기를 가지는 영상은 실시간으로 조정하려할 때 속도가 떨어져 깜박임이 발생하였다. 이를 해결하기 위하여 현재 출력된 부분만을 컴퓨터에

따라 특정한 크기로 리샘플링하여 새로운 영상을 만들고 윈도우를 조정하게 하는 방법을 사용하였다. 사용자는 리샘플링 크기가 512x512일때 영상의 저품질화를 느낄수 있었으나 충분한 속도를 얻을 수 있으며 리샘플링 윈도우의 크기가 1024x1024 이상일 때에는 저품질화를 느끼지 못하였으나 CPU가 최소한 Pentium 4 성능이 요구되었다. 툴킷은 또한 그림 3.12과 같이 히스토그램(histogram)을 보며 윈도우를 조정할 수 있는 기능도 제공한다.

(6) 부분 확대

특정 부분을 확대하여 관찰하는 기능이다. 그림 3.13과 같이 마우스 왼쪽 버튼을 누르면 사각형의 확대 상자가 나타나며 마우스를 움직이며 관찰할 수 있다. Ctrl 키를 누른 상태에서 마우스를 움직이면 배율을 조정할 수 있으며 Shift 키를 누른 상태에서 마우스를 움직이면 확대 상자의 크기를 조정할 수 있다.

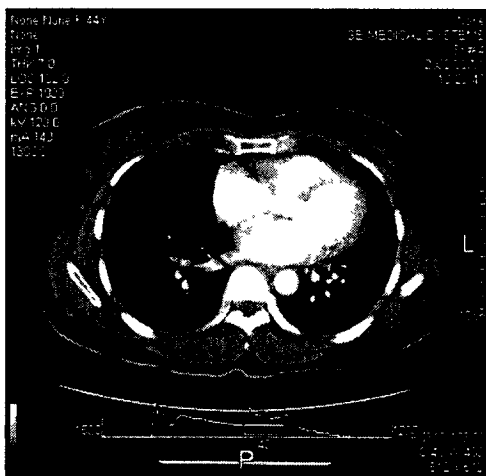


그림 3.12 윈도우 조정

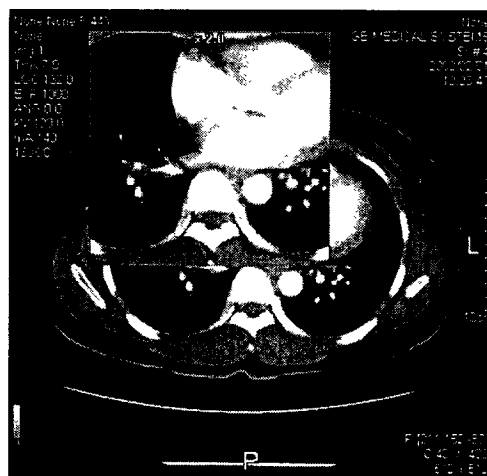


그림 3.13 부분 확대

(7) 반전

현재 보이는 영상을 반전시킨다. 현재 보이는 영상은 8비트 비트맵 영상이므로 보이는 것만 반전시켜 출력할 뿐 원본 데이터는 바뀌지 않는다. 그림 3.14는 반전

기능에 의해 반전된 영상을 보여준다.

(8) 동영상 출력

DICOM 데이터 집합이 동영상을 포함하고 있을 때에는 동영상 관련 기능들을 제공한다. 영상 탐색 기능을 이용하여 첫 번째, 마지막, 다음, 이전 영상으로 이동할 수 있고 재생, 멈춤 기능을 이용하여 동영상을 재생할 수 있으며 이때 반복 기능을 사용하거나 요요 기능을 사용할 수 있다. DICOM 데이터 집합에서 제공되는 프레임 시간을 참조하여 동영상 출력 프레임 수를 결정하며 이 값을 임의로 조정할 수도 있다.

(9) 커브 데이터 출력

DICOM에서는 심전도와 같은 1차원 신호를 커브(curve) 데이터 형태로 저장할 수 있다. 동영상을 포함하는 데이터 집합들은 커브 데이터를 가질 수 있으며 이때 커브 데이터를 올바르게 출력하여야 한다. 제안된 툴킷은 동영상과 함께 커브데이터를 출력하며 현재 출력되는 동영상과 커브의 위치가 일치되도록 표시 막대를 출력하였다.

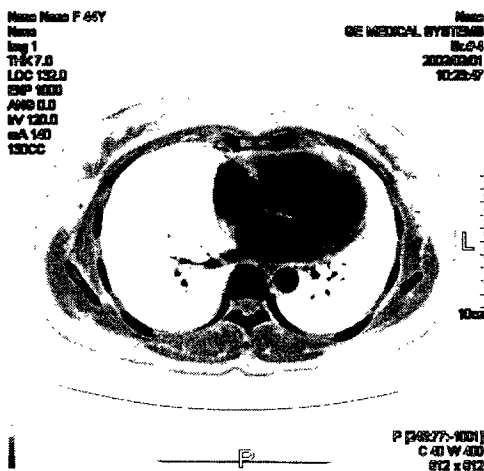


그림 3.14 반전



그림 3.15 커브 데이터의 출력

3.2.2 영상 처리 기능

진단에 필요한 흐림 처리, 선명 처리, 외곽선 검출, 잡음 제거, 회전, 대칭 기능을 그림 3.16~3.21과 같이 각각 구현하였다. 영상 처리 기능은 기능을 적용한 후에 데이터 집합을 저장할 때에 갱신된 영상이 저장되므로 주의하여야 한다. 영상 처리 후에는 항상 되돌리기 기능을 이용하여 이전 상태로 되돌릴 수 있다.[14][15]

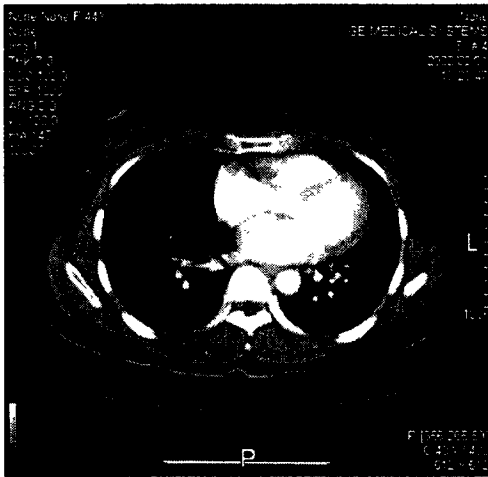


그림 3.16 흐림 처리

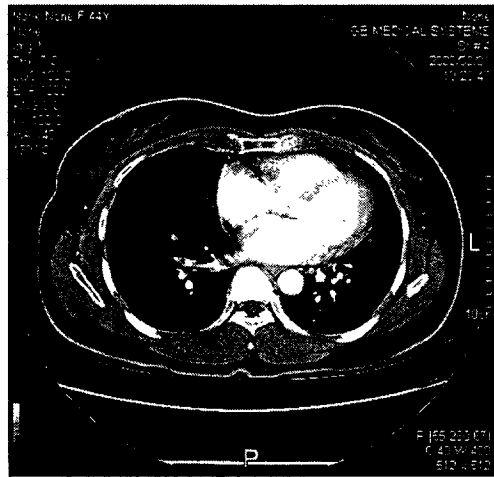


그림 3.17 선명 처리

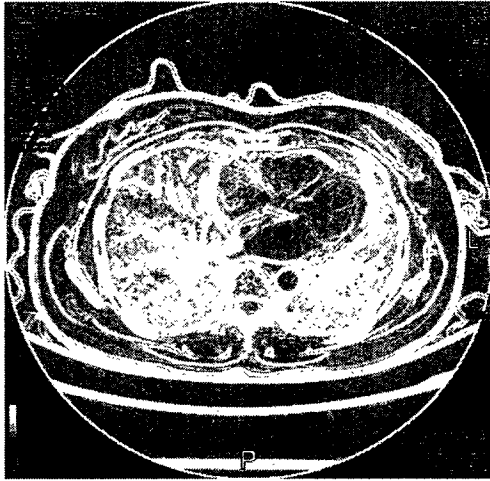


그림 3.18 외곽선 검출

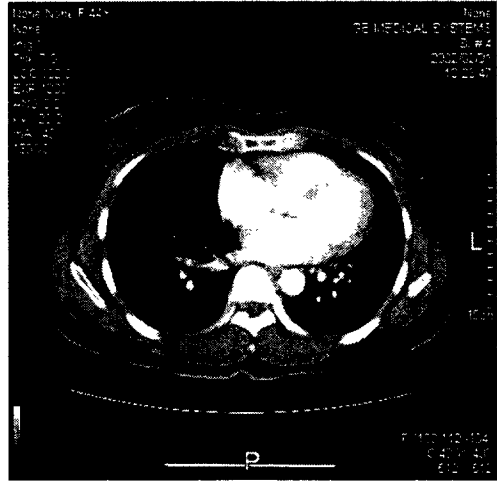


그림 3.19 잡음 제거

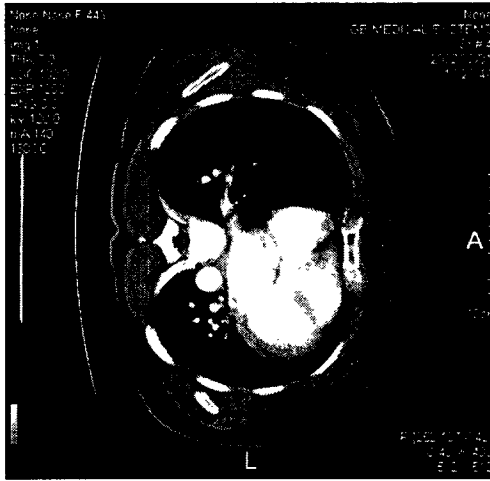


그림 3.20 회전

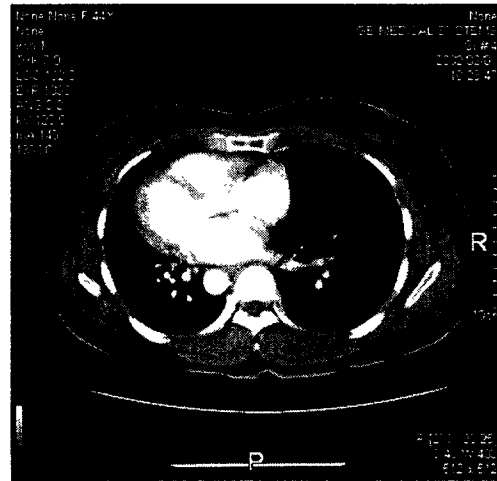


그림 3.21 대칭

3.2.3 주석

제안된 툴킷에서는 그림 3.22와 같이 주석 관련 기능을 구현하였다. 주석은 영상 모듈과 별도로 관리되며 데이터 집합이 저장될 때에도 별도의 파일로 저장된다. 구현된 주석 객체들은 선 객체로서 직선, 화살표, 영역 객체로서 사각형, 원,

자유 도형, 텍스트 객체로서 문자, 측정 객체로 각도 측정, 거리 측정이 있다. 모든 객체는 마우스를 이용하여 간단히 그릴 수 있으며 그린 후에도 제어 점을 잡고 위치나 형태를 변경시킬 수 있다. 영역 객체에서 마우스 오른쪽 버튼을 클릭하면 해당 영역의 폭, 높이, 면적, 평균, 표준편차, 최대값, 최소값이 출력된다. 또한 주석 객체를 영상 데이터에 각인하는 기능도 제공한다.



그림 3.22 주석 객체

3.3 네트워크 모듈

3.3.1 DIMSE

DICOM 표준에서는 11가지의 DIMSE를 정의하고 있다. 제안된 툴킷에서는 임상에서 반드시 필요한 항목들을 표 3.4와 같이 구현하였다. 구현되지 않은 항목들은 주로 DICOM 프린터를 개발할 때 사용되는 항목들로서 제안된 툴킷에서는 포함하지 않았다.

표 3.5 구현된 DIMSE

이름	형태	SCU 구현 여부	SCP 구현 여부
C-STORE	DIMSE-C	O	O
C-GET	DIMSE-C	O	O
C-MOVE	DIMSE-C	O	O
C-FIND	DIMSE-C	O	O
C-ECHO	DIMSE-C	O	O
N-EVENT-REPORT	DIMSE-N	X	X
N-GET	DIMSE-N	O	X
N-SET	DIMSE-N	O	X
N-ACTION	DIMSE-N	O	X
N-CREATE	DIMSE-N	O	X
N-DELETE	DIMSE-N	O	X

DIMSE는 SCU와 SCP 두 가지 역할을 할 수가 있다. 두 역할 모두 새로운 연결이 이루어질 때마다 새로운 쓰레드에서 작업이 시작되므로 개발자의 쓰레드 영역에서 임의로 접근할 수 없다.

SCU의 역할을 하는 과정은 먼저 새로운 소켓 클래스를 요청하여 소켓 ID를 부여 받고 소켓 클래스에 하나 이상의 DIMSE 작업을 추가한 후에 소켓 클래스의 작업 시작을 지시하면 추가된 작업이 순차적으로 이루어지게 된다. 이후의 과정은

소켓 클래스의 메시지 처리 루틴을 사용하여 정보를 획득할 수 있다.

SCP의 역할을 할 때에는 새로운 접속이 발생하게 되면 새 소켓 클래스가 생성되며 이 쓰레드의 메시지 처리 루틴을 이용하여 정보를 획득할 수 있다.

3.3.2 DICOM 서비스 클래스

툴킷에서 구현된 DICOM 서비스 클래스들은 표 3.5와 같다. 각각의 서비스 클래스의 특징 및 사용법은 다음과 같다.

표 3.5 구현된 DICOM 서비스 클래스

서비스 클래스	구현 여부
Verification Service Class	O
Storage Service Class	O
Query/Retrieve Service Class	O
Study Content Notification Service Class	X
Patient Management Service Class	X
Study Management Service Class	X
Result Management Service Class	X
Print Management Service Class	O
Media Storage Service Class	X
Storage Commitment Service Class	X
Basic Worklist Management Service	O
Queue Management Service Class	X

(1) Verification Service Class

원격 DICOM AE와의 통신 여부를 확인하기 위하여 빈번하게 사용되는 기능이다. C-ECHO DIMSE를 사용한다. 그림 3.23은 Verification Service를 사용하는 예제 코드이다.

```

// 소켓 관리 모듈을 생성한다.
ServiceClass * pService = new ServiceClass;
// 소켓 쓰레드 모듈을 생성하여 ID를 부여받는다.
int nMessageID = 1, nSocketID;
nSocketID = pService->NewSocket(ConnectInfo);
// C-ECHO REQUEST DIMSE를 추가한다.
pService->AddCEchoRequest(nSocketID, nMessageID);
// 작업 시작을 통보한다.
pService->StartSCU(nSocketID);

```

그림 3.23 Verification Service Class 사용 예

(2) Storage Service Class

원격 DICOM 장비에 데이터 집합을 저장하고자할 때 사용되는 기능이다. C-STORE DIMSE를 사용한다. 그림 3.24는 Storage Service Class를 사용하는 예제 코드이다.

```

// 소켓 관리 모듈을 생성한다.
ServiceClass * pService = new ServiceClass;
// 소켓 쓰레드 모듈을 생성하여 ID를 부여받는다.
int nMessageID = 1, nSocketID;
nSocketID = pService->NewSocket(ConnectStructure);
// C-STORE REQUEST DIMSE들을 추가한다.
pService->AddCStoreRequest(nSocketID, nMessageID, FileName_1);
pService->AddCStoreRequest(nSocketID, nMessageID+1, FileName_2);
...
// 작업 시작을 통보한다.
pService->StartSCU(nSocketID);

```

그림 3.24 Storage Service Class의 사용 예

(2) Query/Retrieve Service Class

원격 DICOM 장비에 있는 영상을 조회하고 로컬 장비로 영상을 가져올 때 사용되는 기능이다. C-FIND, C-MOVE, C-GET DIMSE가 사용된다. 그림 3.25의 예제 코드와 같이 C-FIND DIMSE를 이용하여 환자, 검사, 시리즈, 영상과 관련된 정보를 조회한다.

```
// 소켓 관리 모듈을 생성한다.
ServiceClass * pService = new ServiceClass;
// 소켓 쓰레드 모듈을 생성하여 ID를 부여받는다.
int nMessageID = 1, nSocketID;
nSocketID = pService->NewSocket(ConnectStructure);
// C-FIND REQUEST DIMSE를 추가한다.
pService->AddCFindRequest(nSocketID, nMessageID, FindStructure);
// 작업 시작을 통보한다.
pService->StartSCU(nSocketID);
```

그림 3.25 Query/Retrieve Service Class 사용 예 (C-FIND)

C-FIND DIMSE를 사용하여 획득한 UID 정보를 이용하여 C-MOVE DIMSE를 이용하면 원격 DICOM 장비에 있는 영상을 로컬 DICOM 장비나 또다른 원격 DICOM 장비로 이동시킬 수 있다. 이때 영상을 받는 DICOM 장비는 C-STORE DIMSE를 지원해야 한다. 그림 3.26은 C-MOVE DIMSE를 사용한 예제 코드이다.


```

// 소켓 관리 모듈을 생성한다.
ServiceClass * pService = new ServiceClass;
// 소켓 쓰레드 모듈을 생성하여 ID를 부여받는다.
int nMessageID = 1, nSocketID;
nSocketID = pService->NewSocket(ConnectStructure);
// C-MOVE REQUEST DIMSE들을 추가한다.
pService->AddCMoveRequest(nSocketID, nMessageID, MoveStructure1);
pService->AddCMoveRequest(nSocketID, nMessageID+1, MoveStructure2);
...
// 작업 시작을 통보한다.
pService->StartSCU(nSocketID);

```

그림 3.26 Query/Retrive Service Class 사용 예 (C-MOVE)

C-FIND DIMSE를 사용하여 획득한 UID 정보를 이용하여 C-GET DIMSE를 이용하면 원격 DICOM 장비에 있는 영상을 로컬 DICOM 장비로 가져올 수 있다. 그림 3.27은 C-GET DIMSE를 사용한 예제 코드이다.

```

// 소켓 관리 모듈을 생성한다.
ServiceClass * pService = new ServiceClass;
// 소켓 쓰레드 모듈을 생성하여 ID를 부여받는다.
int nMessageID = 1, nSocketID;
nSocketID = pService->NewSocket(ConnectStructure);
// C-GET REQUEST DIMSE들을 추가한다.
pService->AddCGetRequest(nSocketID, nMessageID, GetStructure1);
pService->AddCGetRequest(nSocketID, nMessageID+1, GetStructure2);
...
// 작업 시작을 통보한다.
pService->StartSCU(nSocketID);

```

그림 3.27 Query/Retrive Service Class 사용 예 (C-GET)

(3) Basic Print Management Service Class

DICOM 프린터로 영상을 전송하여 프린트할 때 사용되는 기능이다. N-GET, N-SET, N-CREATE, N-ACTION, N-DELETE DIMSE가 사용된다. DICOM 프린터를 사용하는 과정은 상기한 DIMSE들을 반복적으로 사용해야 하는 복잡성이 있으므로 제안된 툴킷에서는 이러한 기능들을 캡슐화한 프린트 모듈을 구현하였다. 그림 3.28은 영상을 프린트하는 예제 코드이다.

```

// 소켓 관리 모듈을 생성한다.
ServiceClass * pService = new ServiceClass;
// 소켓 쓰레드 모듈을 생성하여 ID를 부여받는다.
int nMessageID = 1, nSocketID;
nSocketID = pService->NewSocket(ConnectStructure);
// 프린트 모듈을 생성한다.
KDicomPrint * pPrint = new KDicomPrint;
// 프린트할 영상들을 추가한다.
KD_FILM_SESSION * pFS;
KD_FILM_BOX * pFB;
KD_IMAGE_BOX * pIB;
// Basic Film Session을 추가한다.
pFS = pPrint->CreateFilmSession(1, szDestination, szMedium,
    szLabel, szPriority);
for(i=0;i<nNumPage;i++){
    // Basic Film Box을 추가한다.
    pFB = pPrint->CreateFilmBox(pFS, m_nLayOutV, m_nLayOutH,
        szOrientation, szSize, szMagnification,
        szBorder, szEmptyImage, atoi(szMin),
        atoi(szMax), szTrim);
    // Image Box를 추가한다.
    for(j=0;j<m_nLayOutV * m_nLayOutH;j++){
        pIB = >m_pPrint->GetImageBox(pFB, j);
        // 영상을 추가한다.
        pFrame->m_pPrint->SetImageBox(j+1, pIB, pImage, NULL);
    }
}
// 프린트를 시작한다.
pService->StartPrint(nSocketID, pPrint);

```

그림 3.28 Basic Print Management Service Class 사용 예

(4) Basic Worklist Management Service Class

워크리스트 서버에 있는 검사 정보를 조회하는데 사용되는 기능이다. C-FIND DIMSE가 사용된다. 그림 3.29는 예제 코드이다.

```
// 소켓 관리 모듈을 생성한다.  
ServiceClass * pService = new ServiceClass;  
// 소켓 쓰레드 모듈을 생성하여 ID를 부여받는다.  
int nMessageID = 1, nSocketID;  
nSocketID = pService->NewSocket(ConnectStructure);  
// C-FIND REQUEST DIMSE를 추가한다.  
pService->AddCFindRequest(nSocketID, nMessageID, FindStructure);  
// 작업 시작을 통보한다.  
pService->StartSCU(nSocketID);
```

그림 3.29 Basic Worklist Management Service Class 사용 예

3.4. 테스트 PACS 구현

제안된 툴킷을 이용하여 테스트 PACS를 구현하였다. 테스트 PACS의 개발 목적은 툴킷을 이용한 PACS의 개발 여부를 판단하기 위함이다. PACS가 구현된 후에 각종 밸리테이션 도구 및 상용 PACS 제품과 연동하여 작동 유무를 검사하였다. 개발된 구성 요소들은 모달리티 또는 PACS 관련 제품을 개발하거나 설치할 때에 유용하게 사용될 수 있다. 테스트 PACS는 워크리스트 서버, 영상 획득 서버, 데이터베이스 서버, 영상 조회용 뷰어로 구성된다. 시스템을 단순화하기 위하여 그림 3.30과 같이 영상 획득 서버와 데이터베이스 서버가 같은 시스템에 존재하도록 하였고 단기 저장 서버 및 장기 저장 서버가 별도로 존재하지 않도록 하였다.[16][17]

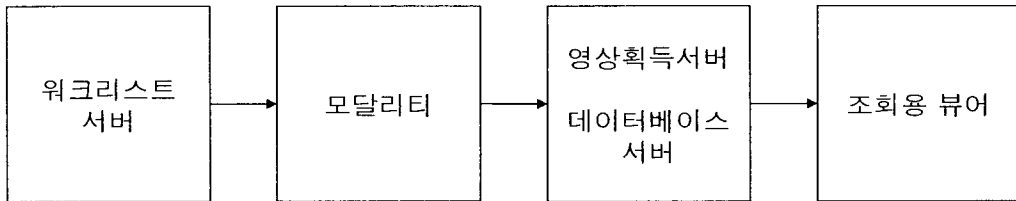


그림 3.30 조회용 뷰어

구현된 구성요소들의 DICOM 표준 준수 여부를 평가하기 위하여 밸리테이션 도구로서 DVT, DCMTK를 사용하였다. 또한 임상 환경에서 연동 여부를 확인하기 위하여 Infinit사과 Marotech사의 PACS 구성요소들을 사용하였다. 각각의 구성요소를 살펴보면 다음과 같다.

3.4.1 워크리스트 서버

워크리스트 서버는 처방전달시스템과 연동되어 검사와 관련된 정보를 획득하여야 하나 테스트 PACS 환경에서는 처방전달시스템이 존재하지 않으므로 사용자가 검사정보를 직접 입력하는 방식을 사용하였다. 모달리티와 검사의 종류는 사용

자가 직접 추가, 삭제하는 방식이므로 어떤 모달리티에도 사용할 수 있다. 다중 쓰레드를 지원하여 두 곳 이상의 노드에서 동시에 접속하여도 지연 과정 없이 작업을 처리할 수 있다. 또한 모달리티 모니터 기능을 제공하여 현재 접속중인 모달리티의 네트워크 정보와 정보 조회수 등을 실시간으로 출력하여 준다. 그림 3.31은 구현된 워크리스트 서버를 보여주고 있다.

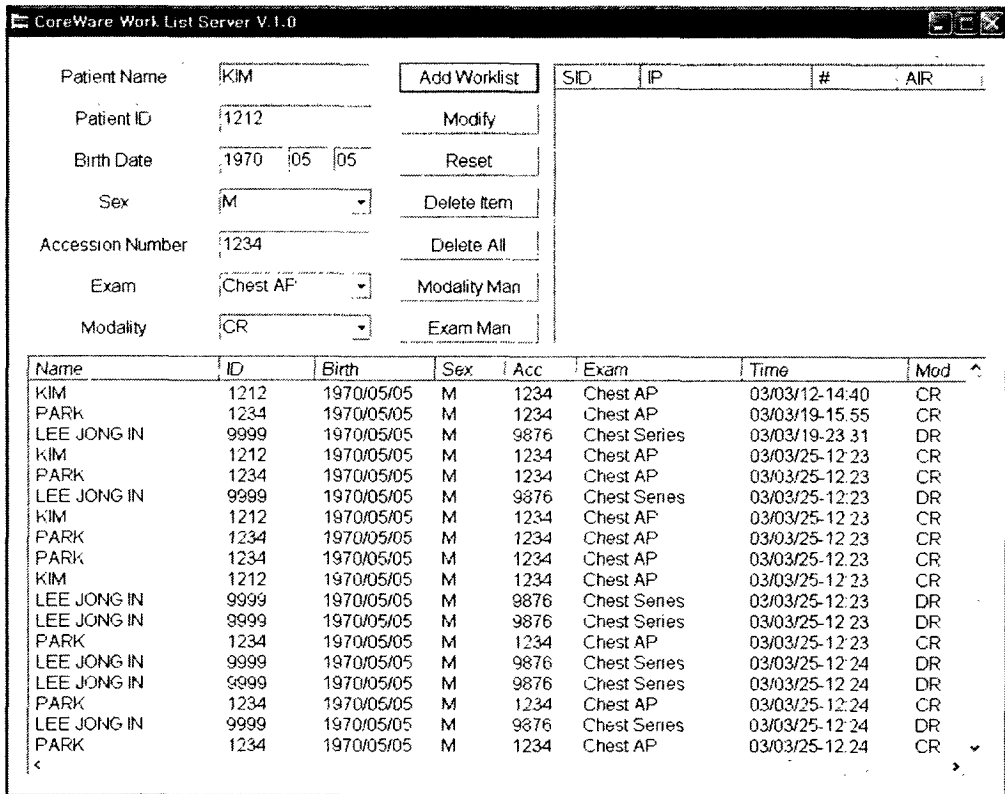


그림 3.31 워크리스트 서버

3.4.2 영상 획득 서버

영상 획득 서버는 모달리티로부터 DICOM 데이터를 집합을 획득하여 환자, 검사, 영상에 관련된 정보를 데이터베이스로 전송하고 영상은 지정된 위치에 저장하였

다. 다중 쓰레드를 지원하여 두 곳 이상의 노드에서 동시에 접속하여도 지연 과정 없이 작업을 처리할 수 있다. 또한 모달리티 모니터 기능을 제공하여 현재 접속중인 모달리티의 네트워크 정보와 정보 조회수 등을 실시간으로 출력하여 준다. 그림 3.32는 구현된 영상 획득 서버를 보여주고 있다.[18][19]

SID	IP	#	AIR
1	127.0.0.1	10	ON
2	127.0.0.1	1	ON
3	127.0.0.1	1	OFF
4	127.0.0.1	1	OFF
5	127.0.0.1	1	OFF
6	127.0.0.1	1	OFF
7	127.0.0.1	1	OFF
8	127.0.0.1	10	OFF
9	127.0.0.1	1	OFF

그림 3.32 영상 획득 서버

3.4.3 데이터베이스 서버

DICOM에서 권장하는 데이터베이스 모델을 기준으로 그림 3.33과 같이 데이터베이스를 설계하였다. 설계된 데이터베이스는 테스트용으로 사용하기 적합하도록 크기가 작고 간단한 마이크로소프트 액세스를 사용하였다. 다중 쓰레드를 지원하여 두 곳 이상의 노드에서 동시에 접속하여도 지연 과정 없이 작업을 처리할 수 있다. 그림 3.34는 구현된 데이터베이스 서버이다.

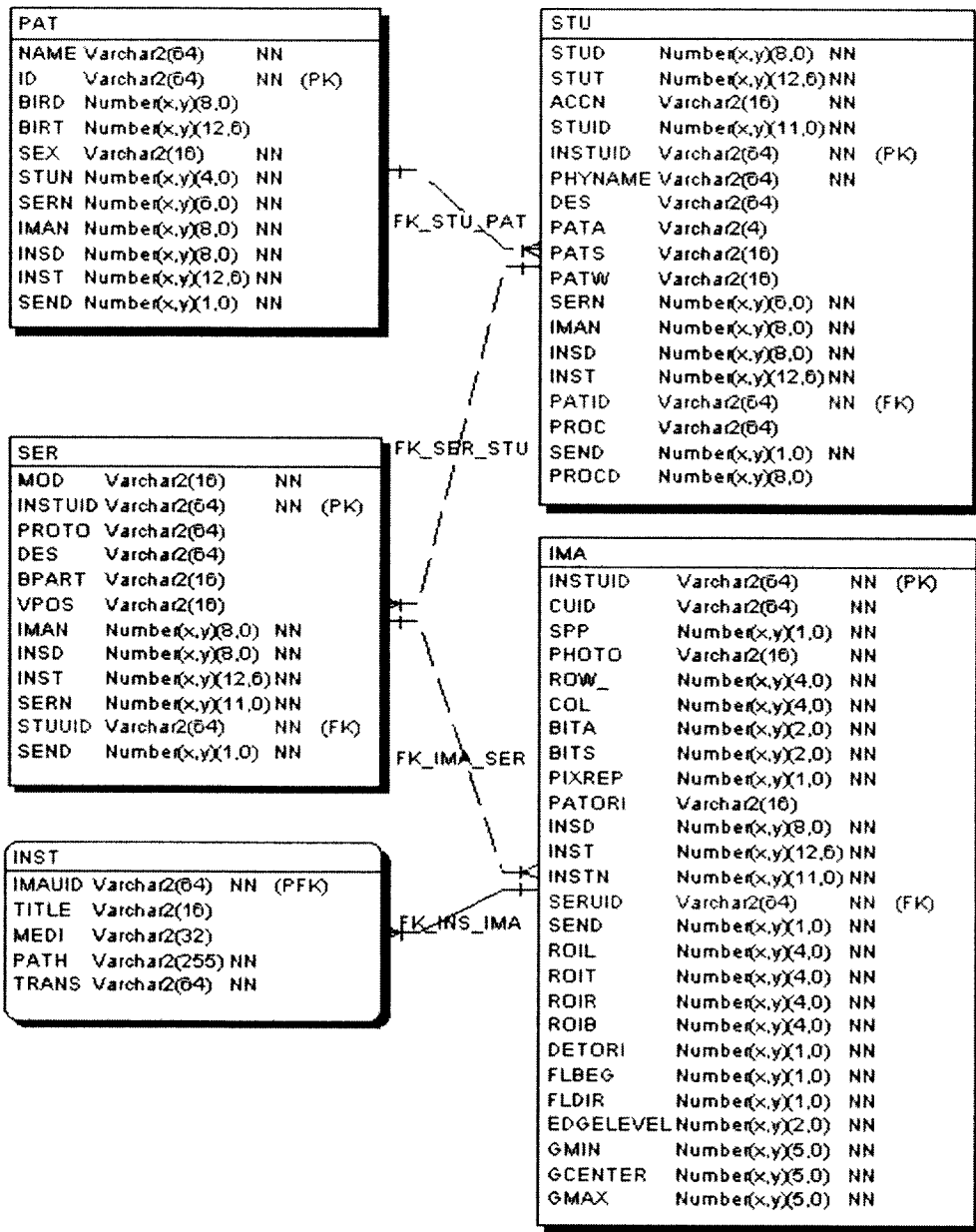


그림 3.33 데이터베이스의 ERD

Date/Time	Name	ID	MD	ACC	S#	I#
1996/10/14 - 14 50 59	AVE+ ABDOMEN	0000002019	CT		1	10
1999/01/30 - 16 52 51	SMPTE	SMPTE	OT	NONE	1	1
2000/06/21 - 00 00 00	Modality LUT Test	MLUT_nn	OT		1	1
2001/08/03 - 09 36 07	PARK SAM SUN FM	155401	MR	28547	1	1
2001/08/03 - 15 57 43	KIM JAE SU M66	129143	CT		2	20
2001/08/04 - 10 59 37	KIM EUNG HWAN M73	145366	CT		5	117
2001/12/07 - 18 41 00	CHOI JUNG SOON 64	1949074	CR	NONE	1	1
2002/01/16 - 16 06 05	SHIN DONG K.YU	123456	DR		1	1
2002/02/01 - 10 14 56	JANG MYEUNG HWA	98-23634	CT	NONE	1	1
2003/03/31 - 15 39 34	Test Man	12345678	DX	NONE	2	2
2003/05/09 - 15 31 40	KIM YEONG SIK	01091195	DX	030509.	2	2
2003/05/09 - 05 59 48	SHIN CHOON SEOK	01094310	DX	030509.	1	1
2003/09/03 - 14 41 46	TEST	TEST	CR	0000	1	1
2003/09/27 - 15 49 04	KIM	700525	RF	5023	1	9
2003/09/27 - 15 44 14	Park Soon Man	545865	RF	587232	1	10
2003/09/27 - 11 19 47	KIM	700525	RF	5023	1	2
2003/10/04 - 17 43 58	KIM	700525	RF	5023	1	6
2003/10/15 - 16 45 23	KIM DONG SUN	KKKK	DX	5987	1	8
2003/10/16 - 16 38 24	KIM DONG SUN	KKKK	DX	5987	1	4

그림 3.34 데이터베이스 서버

3.4.4 뷰어

뷰어는 임상외가 진단을 위하여 영상을 관찰하는데 사용되는 도구로서 제안된 툴킷의 대부분의 기능이 포함된 구성요소이다. 뷰어의 구성과 기능은 다음과 같다.[20][21]

(1) 영상 검색 메뉴

제안된 뷰어는 윈도우 탐색기, 로컬 데이터베이스, 원격 서버, DICOMDIR의 네 가지 메뉴를 이용하여 영상을 검색할 수 있다.

① 윈도우 탐색기 메뉴

그림 35와 같이 마이크로소프트 윈도우즈의 탐색기와 동일한 인터페이스를 제

공하여 시스템에 있는 DICOM 파일을 검색한다. 영상을 선택하였을 때 우측 하단에 미리보기 및 영상에 대한 간단한 설명을 보여준다. 파일이나 폴더를 선택한 후 파일 열기를 지시하면 파일 또는 폴더 내부에 있는 모든 DICOM 파일들이 열린다. “IMPORT” 버튼을 사용하여 파일을 로컬 데이터베이스로 불러올 수 있다.

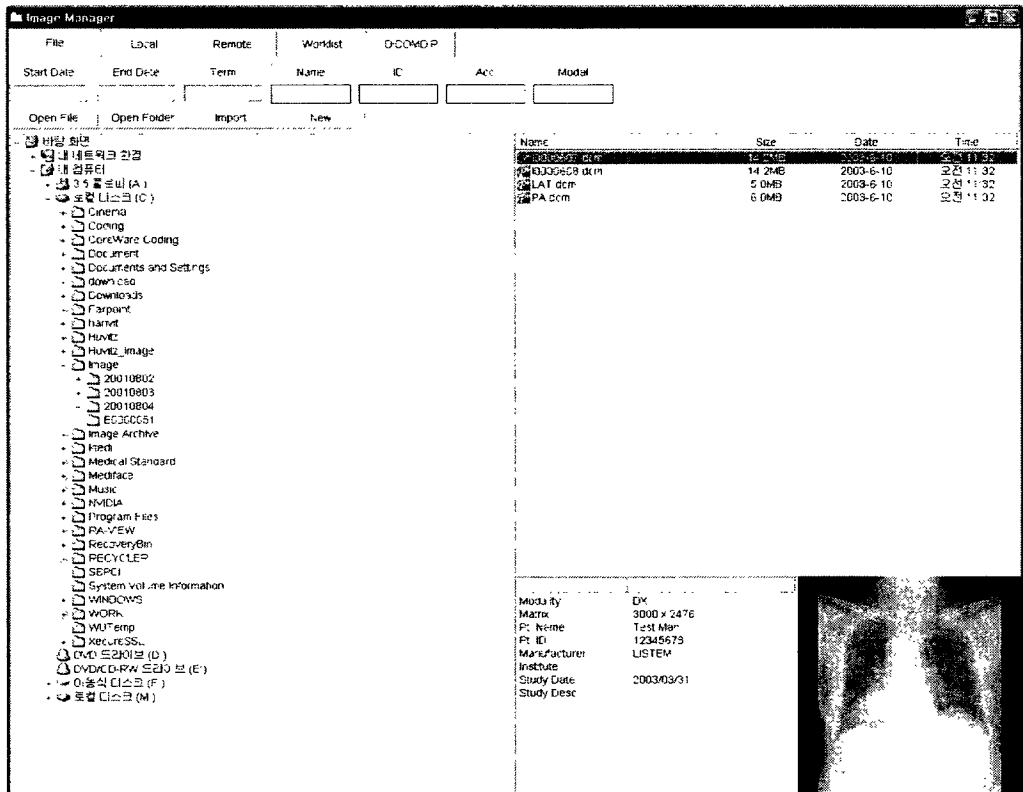


그림 3.35 윈도우 탐색기 스타일의 영상 검색

② 로컬 데이터베이스 메뉴

로컬 데이터베이스를 관리하는 메뉴이다. 원격 DICOM 장비로부터 전송된 데이터 집합들은 모두 로컬 데이터베이스에 저장된다. 영상은 검사 단위로 저장되고 관리된다. 검사의 첫 번째 시리즈의 첫 번째 영상을 작은 영상으로 출력하여 주는 그림 3.36과 같은 앨범 뷰 스타일 또는 텍스트 목록으로 표시해 주는 리스트 뷰

스타일로 검색될 수 있다. 환자 이름, ID, 촬영 날짜, 모달리티등 다양한 항목들을 이용하여 검색이 가능하다. 하나 이상의 검사를 선택하고 열기를 지시하면 영상이 출력된다. “SEND” 버튼을 이용하여 영상을 미리 지정된 DICOM 장비로 전송한다. DICOMDIR 버튼을 누르면 선택된 영상을 DICOMDIR 형태로 저장한다.

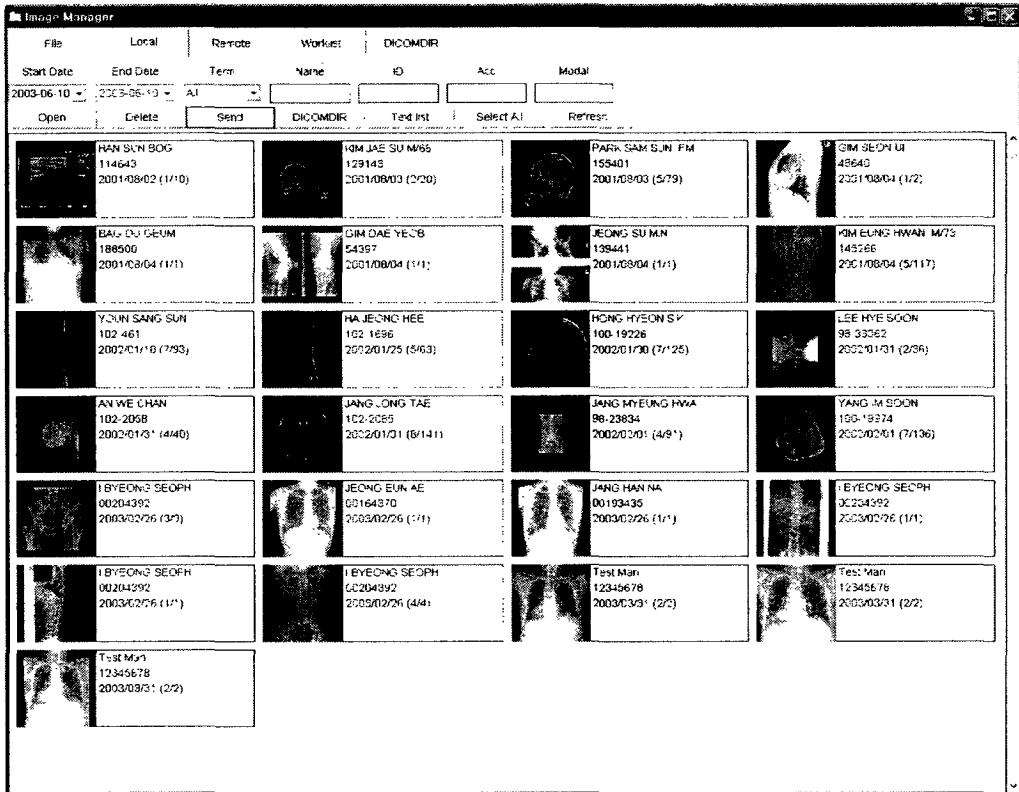


그림 3.36 로컬 데이터베이스 메뉴

③ 원격 서버 메뉴

원격 DICOM 장비에 있는 영상을 조회할 때 사용한다. 촬영 날짜, 환자 이름, ID, 모달리티를 이용하여 검색이 가능하다. 영상을 조회한 후에 “OPEN” 버튼을 사용하면 영상을 전송받아 로컬 데이터베이스에 저장하고 출력한다. 그림 3.37은

구현된 메뉴의 모습이다.

Study Date	Patent Name	Patent ID	Sex	Description	Accession #	Study C	Modality
20030509	YUON SEON YEONG	01035813	M		0305090520	2824	
20030509	KANG DAE YEONG	00311435	F		0305090522	2825	
20030509	HWI JOONG HEE	01071326	F		0305090518	2823	
20030509	HWANG HWA	01095012	M		0305090506	2822	
20030509	HWI CH NAM	00136611	F		0305090503	2821	
20030509	LEE HYUNG WON	01028019	F		0305090501	2820	
20030509	DE SAN A YI KA	01092360	M		0305090497	2819	
20030509	CHO KYEONG BOON	01032657	F		0305090495	2818	
20030509	HWI KIM TAE	01092621	M		0305090490	2817	
20030509	MOON HYEON SOOK	01092657	F		0305090486	2816	
20030509	HWON HYEON MI	01092674	F		0305090473	2815	
20030509	HWI YEONG SH	01091195	M		0305090463	2814	
20030509	HWI JOON HYEONG	01042425	M		0305090460	2813	
20030509	SUN YE SOON	01095613	F		0305090453	2812	
20030509	CHUN HWA YUL	00198152	F		0305090450	2811	
20030509	CHUNG HYE LYEONG	01084354	F		0305090451	2810	
20030509	PARK SOON OHK	00115644	F		0305090435	2809	
20030509	CHANG JONG SHK	01066113	M		0305090397	2808	
20030509	CHO SOON OHK	00186127	F		0305090384	2807	
20030509	TAE JONG SK	01093434	M		0305090367	2806	
20030509	CHANG EUN SEOK	01094755	F		0305090366	2805	
20030509	CHANG MI JA	01098028	F		0305090337	2804	
20030509	HWI DONG BEOK	00184672	M		0305090336	2803	
20030509	PARK JONG JA	01095046	F		0305090333	2802	
20030509	HWI YONG WOOD	01095041	M		0305090332	2801	
20030509	HWI JONG JA	01063075	F		0305090326	2800	
20030509	CHANG SEONG CHEO	01095027	M		0305090325	2799	
20030509	HWI HYEONG JA	01095020	F		0305090321	2798	
20030509	BANG HYEON SOO	01084245	M		0305090320	2797	
20030509	HWI KYOO SOOK	00152231	F		0305090317	2796	
20030509	CHO SEONG CHEUL	01047370	M		0305090309	2795	
20030509	HWI BYEONG DO	00126654	M		0305090302	2794	
20030509	HWI BYEOL SHK	01095646	M		0305090301	2793	
20030509	LEE TAE HG	00307252	M		0305090300	2792	
20030509	NAM HYEONG WUL	01092378	M		0305090299	2791	
20030509	HWI KWON TAE	00214556	M		0305090289	2790	
20030509	CHO DOO SEON	00161414	M		0305090295	2789	
20030509	CHO SEONG CHUN	01087057	M		0305090273	2788	
20030509	HWI KI SEON	01002734	M		0305090272	2787	
20030509	HWI YONG SOO	01084680	M		0305090261	2786	
20030509	HWI WOLAE	01083815	F		0305090259	2785	
20030509	HWI MAN BUN	01095049	M		0305090256	2784	
20030509	MO PYEONG LUK	00220678	M		0305090257	2783	
20030509	CHUNG JAE YIL	01092866	M		0305090240	2782	
20030509	HWI DO EUN	00306467	M		0305090231	2781	
20030509	PARK KIM AN	01093129	F		0305090226	2780	
20030509	PARK WON SOCK	00100363	F		0305090230	2779	

그림 3.37 원격 서버의 영상 검색

④ DICOMDIR 메뉴

DICOM Part 10에 정의된 DICOMDIR 형태로 기록된 CD-ROM등의 미디어를 읽을 때 사용한다. 미디어를 삽입하고 "QUERY" 버튼을 사용하면 미디어에 수록된 영상들을 검사 단위로 모두 출력한다. 원하는 검사를 선택한 후에 "OPEN" 버튼을 사용하면 영상을 출력한다. 그림 3.38은 DICOMDIR 미디어를 읽은 모습이다.

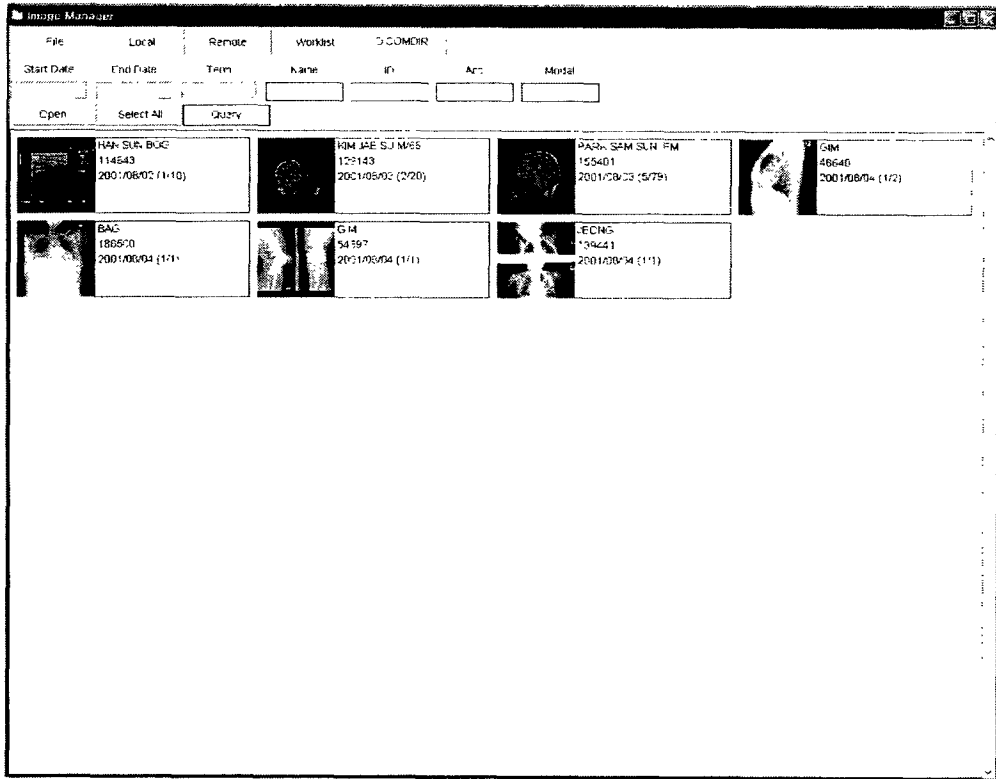


그림 3.38 DICOMDIR로 기록된 미디어 검색

(2) 영상 출력 창

영상 출력창은 그림 3.39와 같다. 영상 출력창은 멀티모니터를 지원하여 하나의 영상 출력창을 여러 모니터에 걸쳐 사용할 수도 있고 각 모니터마다 별도의 영상 출력창을 사용할 수도 있다.

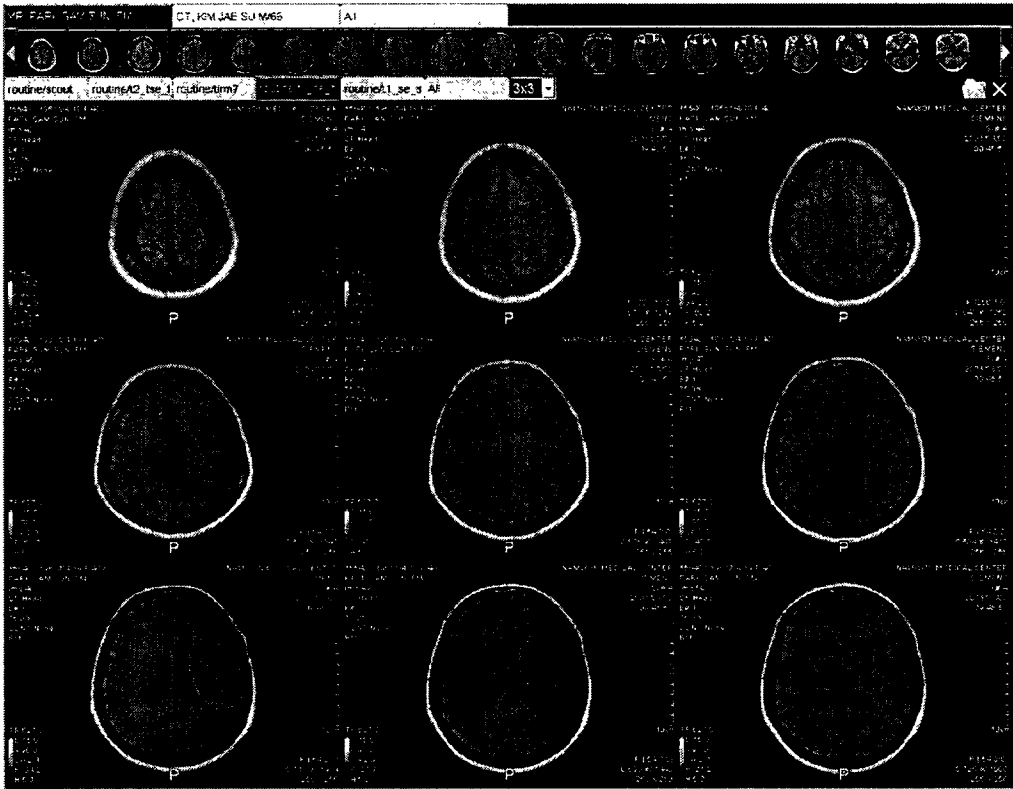


그림 3.39 영상 출력 창

① 화면 구성

화면 구성은 맨 위로부터 검사 탭, 축소 창, 시리즈 탭, 영상 창으로 되어있다. 검사 탭은 열려진 검사를 선택하는 기능을 하며 여러 검사가 열려 있을 때 편리하게 선택할 수 있도록 하였다. 축소 창은 현재 시리즈의 영상들을 모두 보여준다. 시리즈 탭은 열려진 시리즈에 대하여 검사 탭과 동일한 역할을 한다. 영상 창은 영상을 출력하는 창으로서 다양한 레이아웃을 선택할 수 있다.

(2) 도구 모음 바

마우스를 모니터의 상단으로 움직이게 되면 그림 3.40과 같은 도구모음 바가 자동으로 출력된다. 도구 모음 바에는 영상 관찰에 도움이 되는 각종 기능들의 버

튼으로 구성되어 있다.



그림 3.40 도구 모음 바

각 버튼들의 기능은 다음과 같다.

① 마우스 왼쪽 버튼 기능 선택

마우스를 왼쪽 버튼을 누르고 움직일 때의 기능을 선택한다. 포인터, 확대, 이동, 부분 확대, 윈도우 조정의 기능을 선택할 수 있다.

② 기하학적 영상 변환

영상을 시계 방향 및 반시계 방향 회전, 상하 대칭 및 좌우 대칭, 반전 할 수 있는 기능을 제공한다. 변환된 영상은 되돌리기 기능을 이용하여 원래 상태로 돌아갈 수 있다.

③ 영상 배율

마우스를 이용한 확대 기능 이외에 툴킷에서 자동으로 제공되는 배율 기능으로서 일대일 기능, 맞춤 기능, 실제 크기 기능이 있다. 일대일 기능은 모니터의 픽셀과 영상의 픽셀을 일대일 대응시켜 보간에 의한 영상 왜곡을 완전히 제거한 형태의 영상이다. 맞춤 기능은 현재 영상 출력창에 영상을 가득 차게 맞추는 기능으로서 확대 기능 등을 사용하여 영상의 크기 및 위치가 변경되었을 때 초기 상태로 돌리려면 이 기능을 사용한다. 실제 크기 기능은 모니터의 픽셀간 거리를 측정하고 입력하여 이 자료를 토대로 출력될 영상을 실제 크기로 출력하는 기능이다.

④ 영상 필터

흐림 처리, 선명 처리, 외곽선 처리 등의 필터를 사용하여 영상 처리가 가능하다. 그림 3.41은 영상 필터 대화상자이다. 전체 영상에 대해서는 실시간으로 영상 필터를 적용할 수 없지만 영상 필터 대화상자에는 축소 영상이 출력되어 영상 필터가 적용된 영상의 모습을 실시간으로 보여준다.

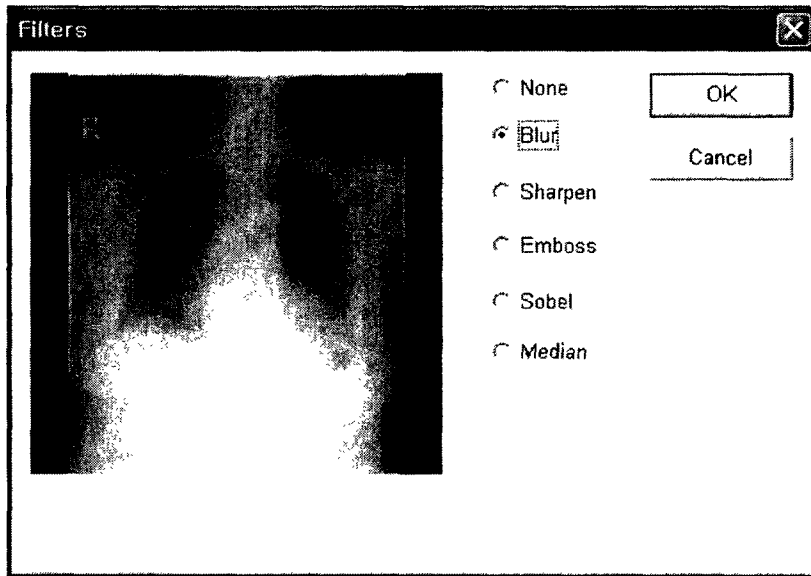


그림 3.41 영상 필터 대화상자

④ 주석

선, 화살표, 사각형, 원, 자유곡선, 문자 주석을 영상에 추가할 수 있다. 또한 각도 및 거리 측정 도구를 사용할 수 있다.

⑤ 프린트

윈도우즈 프린터 및 DICOM 프린터를 사용할 수 있다. 그림 3.42는 프린트 대화상자로서 영상 출력과 관련된 다양한 기능을 설정할 수 있다. 프린터 대화상자에서는 선택된 이미지 또는 현재 시리즈를 선택하여 출력할 수 있다. 사용자가 영상의 레이아웃을 직접 선택할 수 있으며 미리보기 기능을 지원하여 출력될 모양

을 예측할 수 있다. 또한 DICOM 프린터의 작동 유무를 프린트 이전에 확인해볼 수 있다. 프린터의 세부 사항 설정은 윈도우즈 프린터의 경우 윈도우즈의 프린터 설정 창이 나타나며, DICOM 프린터의 경우 DICOM 프린터 설정창이 출력된다.

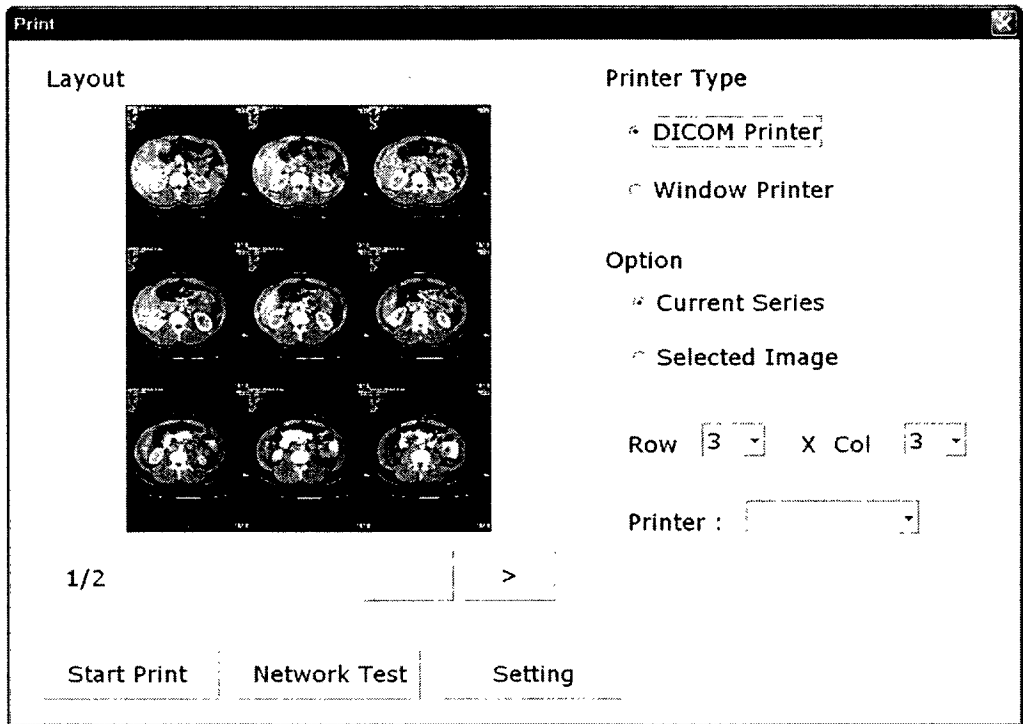


그림 3.42 프린트 대화상자

제 4 장 고찰

본 장에서는 제안된 툴킷의 성능을 고찰하여 보고 툴킷을 개발함에 있어 문제가 되었던 요소들을 되돌아본다. 또한 툴킷을 사용하여 제품화되었거나 현재 개발 중인 제품을 소개하고 툴킷의 앞으로의 개선 방향에 대하여 논의한다.

4.1 툴킷의 성능 고찰

임상의는 올바른 진단을 위하여 가능한 모든 방사선 촬영을 지시하고 그 결과 영상을 상호 비교하여 진단에 참조한다. 제안된 툴킷은 환자 1명에 대하여 CT, CR, MR 촬영을 실시했다고 가정된 테스트 영상들에 대하여 범용 컴퓨터에서 CT 영상 50장, MR 영상 100장, CR 영상 5장을 출력하는데 각각 2.3초, 1.4초, 1.4초가 소요되어 모든 작업이 5초 이내 가능하였으며 PC의 물리적 메모리도 거의 소비하지 않았다. 또한 기존의 영상 툴킷과 차별성을 두어 진단에 필요한 대부분의 영상 처리를 지원하였으며 DICOM 표준에 의한 영상의 송수신 및 프린트를 위하여 고수준의 모듈화를 구현하여 기존 툴킷에 비교하여 매우 편리하게 개발할 수 있도록 하였다.

4.2 툴킷 개발의 장애 요소

테스트 대상이 된 영상들은 대부분 DICOM 표준을 충실히 따랐으며 툴킷과 잘 연동되었으나 일부 영상들은 그렇지 못하였다. 이러한 영상들은 벨리태이션 도구들에서도 오류를 나타내었다. 그러나 대부분의 상용 툴킷들은 이러한 오류에 적절히 대처하는 융통성을 보였으며 이로 인하여 제안된 툴킷도 몇몇 부적절한 영상들에 대하여 출력을 거부할 수가 없어 예외 처리 규정을 따로 두어 정상적으로 처리해야만 했다. 이러한 이유로 표준을 구현하는 것 못지않게 표준을 따르지 않는 데이터 집합에 대하여 예외 처리 규정을 만드는 작업도 매우 큰 비중을 차지하였다. 이러한 영상은 PACS 설치 과정에서 모달리티 제조업체와 PACS 업체간

에 항상 분쟁을 만들어내었다. 이러한 문제점들을 공식적으로 해결해줄 수 있는 기구의 출현이 매우 요구된다고 사료된다.

4.3 응용 사례

(1) CT

틀킷은 1999년 최초 개발을 시작하여 2001년 후반기에 L사에서 개발한 CT 시제품의 콘솔에 최초로 사용되었다. (그림 4.1) CT 콘솔에서는 DICOM 데이터 집합의 인코딩/디코딩, 영상 출력, 영상 처리 및 주석, Storage Service Class 기능들을 구현하였다. 그러나 데이터 집합 및 영상 출력 속도가 느리고 DIMSE가 단일 쓰레드에서 동작하는데 기인된 문제점들이 나타나며 틀킷의 개선이 요구되었다.

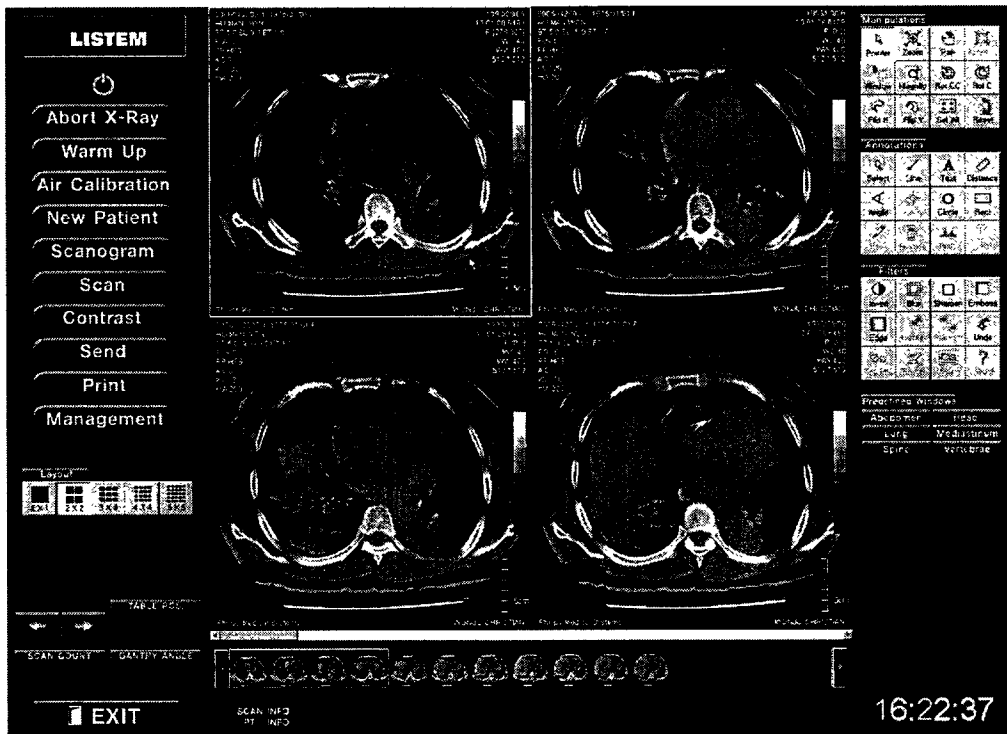


그림 4.1 L사의 CT 콘솔

(2) PACS 뷰어

2002년 전반기에는 국내의 PACS 회사의 뷰어 제품(그림 4.2)에 채택되어 2003년 후반기까지 국내 9개 종합병원에 설치되어 사용되고 있다. 이 기간을 통하여 툴킷은 임상에서 발생하는 매우 다양한 영상을 경험하게 되며 많은 문제점들이 제기되고 수정되었다. 또한 임상에서 많은 요구 사항들이 주문되었고 이를 툴킷에 반영하였다. 뷰어의 특성상 다량의 영상을 처리해야 하므로 매우 효율적인 코드가 요구되어 파일 매핑 기법을 사용하여 메모리를 절약하였고 여러 시행착오를 거쳐 알고리즘을 개선하여 속도를 증가시켰다.

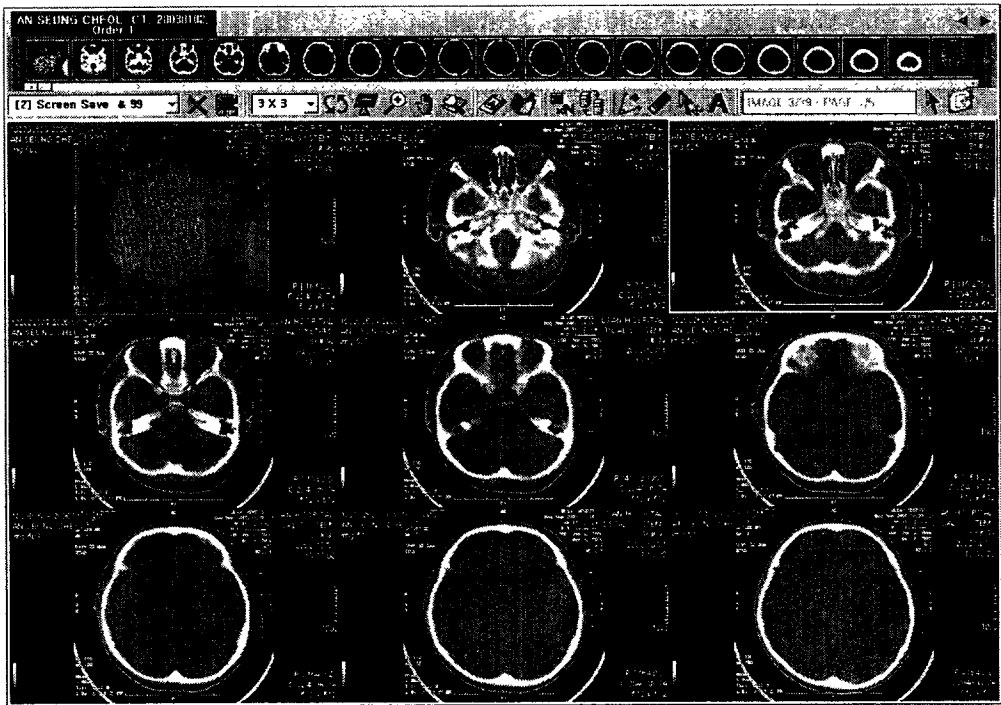


그림 4.2 P사의 PACS 뷰어

(3) DR

2003년 전반기에는 L사의 모달리티 제품에 채택되어 2003년 후반기까지 총 6개 종합병원에서 사용되고 있다. 이 제품은 모달리티로서는 최초로 임상에서 사용된 제품이었다. 이 기간을 통하여 Basic Worklist Management Service Class, Print Management Service Class들에 대한 기능이 추가되었다. 임상에 설치된 국내외의 다양한 PACS들의 구성 요소들과 연동되며 많은 기능들이 수정되거나 추가되었다. 그림 4.3은 투시 촬영용 콘솔 화면이다.

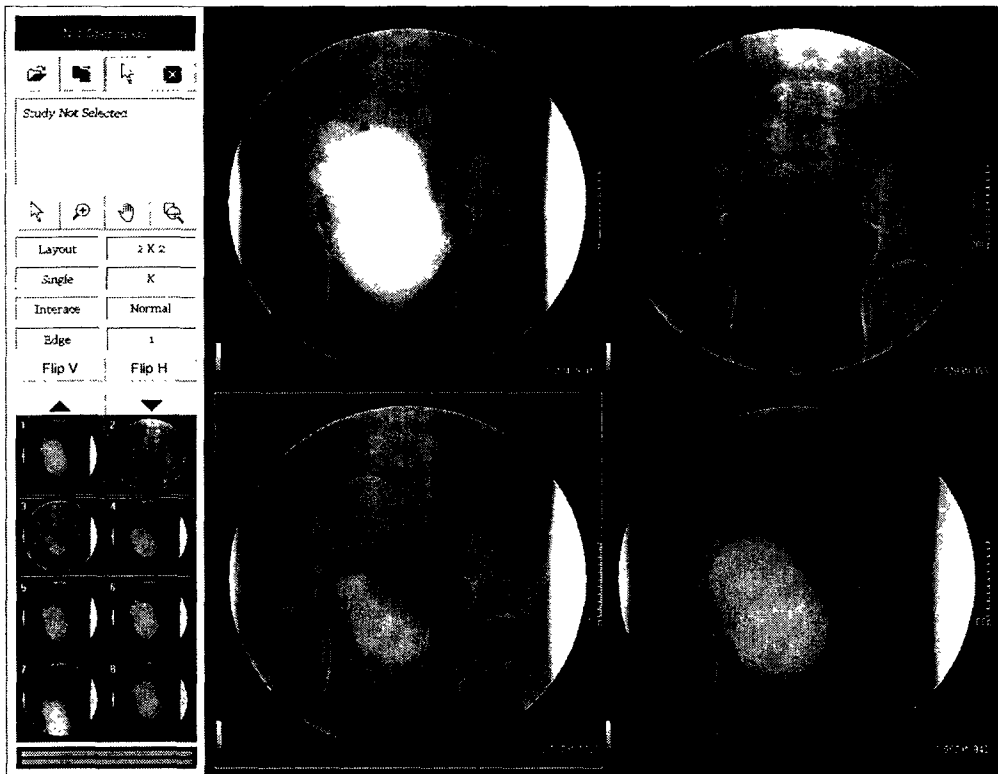


그림 4.3 L사의 디지털 투시 콘솔

(4) 안과용 영상 관리 시스템

또한 같은 시기에 그림 4.4와 같이 H사의 검안기용 영상 관리 시스템의 개발에 사용되었다. 이 기간을 통하여 툴킷의 컬러 영상과 관련된 기능들이 수정 보완되었다.

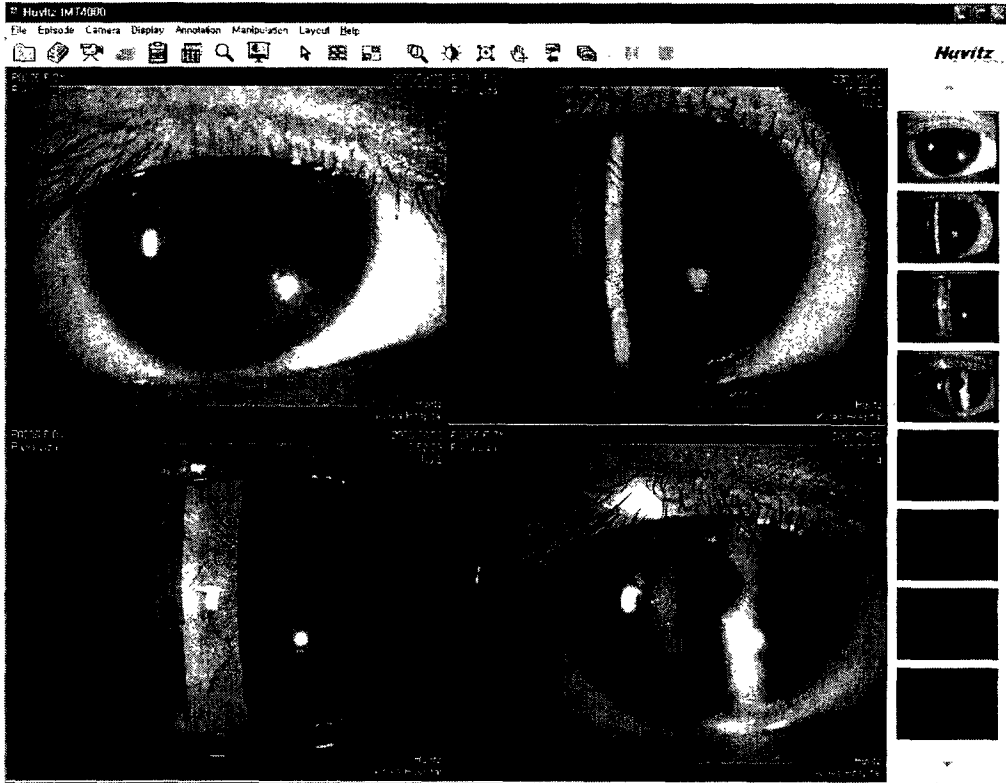


그림 4.4 H사의 안과용 영상 관리 시스템

4.4 개선 방향

DICOM은 매년 새로운 판이 발표되고 있으며 이중 임상에 적용되는 부분도 점차 확대되고 있다. 제안된 툴킷은 임상에서 주로 쓰이는 기능을 위주로 구현하였으며 DICOM 표준의 구현되지 않은 부분과 새롭게 발표되는 새 문서와 수정본을 구현해야 한다. 또한 사용하기 편리하도록 개선해야 한다. 제안된 툴킷에서는 기

존 툴킷의 단점인 복잡성을 줄이기 위하여 객체 지향적인 구조를 가지고 대부분의 기능을 모듈화 시켰으나 DICOM의 방대함과 네트워크를 사용해야 하는 문제 때문하여 여전히 복잡함을 보이고 있다. 그러므로 새로운 방법에 의한 단순화가 요구된다.

제 5 장 결론

병원 환경은 처방 전달 시스템 등의 다양한 정보 체계가 존재하며 또한 수많은 의료 장비가 존재한다. 최근까지 의료 장비에서 발생하는 정보들을 통합하려는 많은 표준들이 개발되었으며 그 중 가장 널리 사용되는 것이 DICOM이다. DICOM은 그 이름이 뜻하는 대로 디지털 영상 장비간의 통신이라는 원래 목표를 초과 달성하여 현재에는 일반 장비까지 그 영역을 확대하고 있다.

DICOM 표준을 준수하기 위해서는 DICOM 표준을 구현해야 하며 이를 위하여 DICOM 툴킷이 반드시 필요하다. 현재 업계에서 사용되고 있는 툴킷은 DICOM 표준의 구현에만 중점을 두고 있어 툴킷의 사용자가 이를 이용하여 응용 프로그램을 개발할 경우 영상 관련 부분의 구현에 별도의 시간과 노력을 투자해야 하는 부담이 있다. 또한 유닉스를 주 기반으로 개발된 후에 윈도우즈용으로 이식되어 윈도우즈를 주로 하는 병원 환경에서 최상의 성능을 발휘할 수 없으며 속도가 느리거나 메모리 소모가 심하여 대량의 영상을 처리해야 하는 병원 환경에 적합하지 않고 툴킷의 구조가 매우 복잡하다.

본 논문에서는 기존 툴킷의 단점을 보완한 다음과 같은 특징을 갖는 DICOM 툴킷을 구현하였다. 첫 번째, DICOM 관련 기능과 영상 관련 기능을 모두 구현하였으며 추가로 의료 영상 관련 기능들을 구현하였다. 두 번째, 범용 컴퓨터를 주로 사용하는 병원 환경에 적합하도록 윈도우즈 운영체제를 기반으로 하였다. 세 번째, 빠른 영상 출력 속도를 보장하며 물리적 메모리를 거의 사용하지 않도록 설계하였다. 네 번째, 의료 영상 저장 시스템 등의 환경에서 필요한 대부분의 의료 영상 관련 기능이 포함되어 있으며 복잡성을 줄이기 위해 객체 지향적인 구조로 설계하였다. 구현된 툴킷은 수백장으로 구성된 테스트 영상을 수초 이내에 출력하며 메모리를 거의 소모하지 않아 임상 환경에 적합하였다. 구현된 툴킷은 임상에서 사용되기에 충분한 성능을 보였으며 실제로 제품화되어 성공적으로 사용되었다. 또한 제안된 툴킷으로 PACS의 구성요소인 워크리스트 서버, 영상 획득 서버, 데이터베이스 서버, 뷰어를 각각 구현하여 성능을 평가하였다.

DICOM 표준은 새로운 장비나 기술이 출현할 때마다 계속 확장되고 있으며 이를 따르는 업체 및 장비도 계속 증가하고 있다. 그러므로 DICOM에 대한 지속적인 연구와 효율적인 툴킷의 개발이 매우 필요하다고 사료된다.

참 고 문 헌

1. NEMA Standard Publication PS 3.1-14, Digital Imaging and Communications in Medicine (DICOM), National Electrical Manufacturers Association, 2101 L Street, N. W., Washington, D. C. 20037, 1992-2001
2. H. K. Huang, PACS, Basic Principles and Applications, A John Wiley & Sons, 1999
3. <http://www.dclunie.com/dicom-status/status.html>, December 1, 2003
4. <http://wuerlim.wustl.edu/DICOM>, December 1, 2003
5. <http://dicom.offis.de>, December 1, 2003
6. <http://www.merge.com>, December 1, 2003
7. <http://www.leadtools.com>, December 1, 2003
8. Adrian Moise, M. Stella Atkins, "New Trends in Radiology Workstation Design", SPIE Conference on PACS and Integrated Medical Information Systems, pp. 174-181, San Diego, 2002
9. K. Kirk Shung, Michael B. Smith, Benjamin M. W. Tsui. Principles of Medical Imaging, Academic Press, pp. 51-54, 1992
10. M.M. Sung, H.J. Kim, "A Study of DICOM Validation in YUMC Mini PACS", Journal of Korean Society of PACS, Vol. 6, No. 2, pp. 67-70, 2000
11. Jorg Riesmeier, Marco Eichelberg, Peter Jensch, "An Approach to DICOM image display handling the full flexibility of the standard's specification", SPIE Conference on Image Display, pp. 363-369, San Diego, 1999
12. <http://www.infinitt.com>, December 1, 2003
13. <http://www.marotech.co.kr>, December 1, 2003
14. Randy Crane, Simplified Approach to Image Processing, A Prentice Hall, 1997
15. Russ, John C, "The Image Processing Handbook 2nd ed", CRC Press, 1995

16. SS Furue, N Bertozzo, M Yamaguti, MA Gutierrez, "A Flexible Storage Architecture for Large PACS", Computers in Cardiology, 29, pp.405-408, 2002
17. David A. Clunie, "*Designing and Implementing a PACS-Aware DICOM Image Object for Digital X-ray, Mammography and Intra-oral Applications*", SPIE Conference on PACS Design and Evaluation, pp. 83-89, San Diego, 1999
18. J. Bernarding, A. Thiel, I.Decker, T. Tolxdorff, "Implementation of a dynamic platform-independent DICOM-server", Computer Methods and Programs in Biomedicine 65, pp. 71-78, 2001
19. H. K. Huang, F. Cao, "Fault-Tolerant PACS Server Design", SPIE Conference on PACS and Integrated Medical Information Systems, pp. 83-92, San Diego, 2001
20. Steven C. Horii, "*DICOM Image Viewers: a survey*", SPIE Conference on PACS and Integrated Medical Information Systems, pp. 251-259, San Diego, 2003
21. K. S. Song, M. J. Shin, J. H. Lee, Y. H. Ahn, "*Economic Impact of Off-line PC Viewer for Private Folder Management*", SPIE Conference on PACS Design and Evaluation, pp. 414-419, San Diego, 1999

ABSTRACT

A Study For The Development of DICOM Toolkit Software

Kim, Dong Sun
Dept. of Biomedical Engineering
The Graduate School
Yonsei University

Digital Imaging and Communications in Medicine (DICOM) is the standard among the digital medical imaging systems. DICOM toolkit is the software to implement DICOM. It is essential to develop any digital medical imaging systems. Commonly accepted toolkits have following obstacles. 1) Since they only support functions for DICOM, developers have to implement the image processing and display related functions by themselves. Some toolkits provide those functions, but they are not suitable for medical images. 2) Since toolkits are designed and implemented at UNIX, when it is used hospitals which are using the Windows for their PACS, the ported code can not play the best performances. 3) The processing time of the conventional toolkits are slow and the memory requirement is too high. So it is not adequate for the hospital environment that handles mass image data. 4) Since toolkit structure is too complicated, which are mainly due to the complexity of DICOM, we need to develop highly efficient structure for DICOM.

In this paper, the developed toolkit has solved the problems in the conventional DICOM toolkits and has following features. 1) It provides not

only DICOM and image related functions but also supports medical image specialized functions. 2) It is designed to do best performances at Microsoft Windows which is currently becoming major operating system in hospital computer environment. 3) It guarantees high image processing and display speed and uses less physical memory. 4) It is designed as object oriented structure. for the simplification of the structure of DICOM complexity and supports most of functions needed at PACS environment.

The developed toolkit took 12 seconds to display 120 images which are collected from different modalities and manufacturers. And the developed DICOM used less physical memory space than others. And the developed DICOM has been successfully applied for the following image modalities such as PACS viewing system, Digital Fluoroscopy, Console of Computed Tomography and the Digital Slit Lamp system.

Key Works : DICOM, Toolkit, PACS