익명화 방법을 적용한 임상진료문서 등록 기법 연구

(A New Method of Registering the XML-based Clinical Document Architecture Supporting Pseudonymization in Clinical Document Registry Framework)

김 일 광 † 이 재 영 † 김 일 곤 †† 곽 연 식 †††
(Kim Il Kwang) (Lee Jae Young) (Kim Il Kon) (Kwak Yun Sik)

요 약 진료기관 사이뿐 아니라 국가 경계를 넘어선 환자진료 정보 교류에 대한 요구사항이 세계적으로 증가되고 있으며 이에 대한 연구가 활발하게 진행되고 있다. 본 논문에서는 임상진료문서 등록 저장소에서의 임상진료문서 등록, 조회 방법에 관한 두 가지 기법을 제안한다. 그 첫 번째는, 임상진료문서 관련 부속파일에 대한 참조와 처리를 위해 적하목록(Manifest)을 구성하고 사용을 제안하는 것이다. 두 번째는 한층 강화된 임상진료문서 보안전략을 통해 환자 익명성을 제공할 수 있는 방법이다. 전자는 네트워크 장애와 같은 외부요인에도 임상진료문서 관련 부속자료에 대한 로컬 참조를 가능케 하여 끊김 없는 뷰(view)를 구성할 수 있게 한다. 후자는 환자의 신상정보를 담은 임상진료문서 헤더와 진단과 처치 정보를 담은 임상진료문서 바디가 지리적으로 분산된 하나 이상의 저장소에 분리 저장되기 때문에 어느 하나의 저장소가 공격 당하더라도 공격자는 환자의 단편적인 정보만 획득하게 된다. 이는, 결국 환자의 신상정보와 병력정보를 단절시킴으로써 사생활침해의 소지를 줄이고 개인정보보호 효과를 가져올 수 있게 한다.

키워드 : 익명화, 임상진료문서, 임상진료문서 등록저장소, ebXML 레지스트리, 리파지토리, ebXML 등록저장소

Abstract The goal of this paper is to propose a new way to register CDA documents in CDR (Clinical Document Repository) that is proposed by the author earlier. One of the methods is to use a manifest archiving for seamless references and visualization of CDA related files. Another method is to enhance the CDA security level for supporting pseudonymization of CDA. The former is a useful method to support the bundled registration of CDA related files as a set. And it also can provide a seamless presentation view to end-users, once downloaded, without each HTTP connection. The latter is a new method of CDA registration which can supports a de-identification of a patient. Usually, CDA header can be used for containing patient identification information, and CDA body can be used for diagnosis or treatment data. So, if we detach each other, we can get good advantages for privacy protection. Because even if someone succeeded to get separated CDA body, he/she never knows whose clinical data that is. The other way,even if someone succeeded to get separated CDA header; he/she doesn't know what kind of treatment has been done. This is the way to achieve protecting privacy by disconnecting association of relative information and reducing possibility of leaking private information. In order to achieve this goal, the method we propose is to separate CDA into two parts and to store them in different repositories.

Key words : Pseudonymization, De-Identification, Clinical Document Architecture, CDA, Clinical Document Registry, Clinical Document Repository, CDR, CDA Registry, CDA Repository, xml, ebXML

· 본 연구는 보건복지부 보건의료기술진흥사업의 지원에 의하여 이루어진 것임(A05-0909-A80405-05N1-00000A)
† 정 회 원 : 경북대학교 IHIS연구소 연구원
dinosa@daum.net
intvis@lycos.co.kr
†† 종신회원 : 경북대학교 IHIS 연구소장
ikkim@knu.ac.kr
††† 정 회 원 : 경북대학교 의료정보학과 교수
yskwak@knu.ac.kr
논문접수 : 2007년 2월 27일
심사완료 : 2007년 7월 3일

1. 서 론

최근 의료 정보 공유를 위한 많은 연구와 노력이 세계적으로 진행되고 있으며 진료기관뿐 아니라 국가적 경계를 넘어선 진료정보 교환에 대한 요구사항도 꾸준히 증가하고 있다. 더불어, 이러한 요구사항을 수용하기 위한 표준화 노력들도 진행되고 있다[1-4]. 이 중 HL7 (Health Level 7)은 서로 다른 보건의료분야 소프트웨어 어플리케이션 간 정보가 호환될 수 있도록 하는 메시지 규칙의 집합으로 1987년에 처음으로 개발되었다. HL7 v2.4가 2000년 10월에 ANSI표준으로 인정된 이후 현재 v3.0이 XML기반의 보다 견고한 메시지 구조를 바탕으로 표준화 작업이 진행 중에 있다.

2000년 11월에 HL7은 XML기반의 임상진료문서의 구조적 표현을 위한 목적으로 임상진료문서(CDA: Clinical Document Architecture)를 발표하였고 CDA release 1이 ANSI 표준으로 승인되었다[1]. CDA는 상호 이질적인 병원정보시스템이나 소프트웨어 어플리케이션들이 서로 통신함에 있어서 표준에 기반한 문서교환이 가능하게 해준다. 이러한 CDA의 등장 이후, 이에 기반한 진료문서 교환을 위한 실험적 연구와 다수의 프로젝트가 진행되고 있다[5,6].

그렇지만 여기서 한 가지 고려할 사항은 비록 CDA 문서가 XML에 기반을 두고 임상진료문서를 구조적으로 표현하고 서로 다른 정보시스템간에 문서를 주고 받을 수 있다고는 하지만 환자의 이동회수에 비례해서 시시각각 발생하는 CDA문서를 체계적인 방법으로 영구보관, 관리한다는 것은 결코 쉬운 일이 아니다. 더군다나, 환자중심의 평생전자건강진료기록환경을 생각해 볼 때 서로 다른 진료기관에서 생성, 작성된 동일 환자의 CDA문서를 어떻게 식별하고, 통합, 관리할 것인 가에 대한 문제에는 여러 가지 이견과 이슈가 존재한다.

CEN/ISSS e-Health Standardization Focus Group의 보고서 "Current and future standardization issues in e-Health domain: Achieving Interoperability"[4]에 따르면 미래 의료정보화 기술을 위한 전략적 목표와 이슈로 아래와 같이 5가지를 제시하고 있다.

- 진료 기록 레코드에 대한 접근성 향상(improving access to clinical records)
- 환자의 이동성을 고려한 진료기관 간 데이타 교류(enabling patient mobility and cross border access to healthcare)
- 의료 사고 방지(reducing clinical errors and improving safety)
- 환자와 전문가를 위한 양질의 정보 제공(improving access to quality information on health for patients and professionals)
- 진료 절차의 효율성 증대(improving efficiency of healthcare processes)

본 논문에서는 이러한 세계적 동향에 따라 CDR 프레임워크(Clinical Document Registry Framework)[7,8]에서의 새로운 형태의 임상진료문서 등록, 검색 방법을 제안한다. 제안하는 방법은 임상진료문서의 관련 부속파일에 대한 끊김 없는 참조를 지원할 수 있는 방법과 한층 강화된 보안 전략에 의해 환자의 프라이버시를 보호하고 익명화의 간접적 효과를 유도할 수 있는 방법이다.

본 논문의 구성과 간략한 내용은 다음과 같다.

2장에서는 본 연구와 관련 있는 다른 연구들에 대해 간략하게 살펴보고, 3장에서 그 연구들의 한계점을 피력한다. 4장에서는 그러한 한계점을 해결하기 위한 대안으로 새로운 형태의 임상진료문서 등록방법을 제안, 설명한다. 5장에서는 몇 가지 정성적 특성을 기준으로 본 연구의 중요성과 타당성을 평가한다. 그리고 마지막으로 6장을 통해 본 연구의 내용을 요약하고 결론을 내린다.

2. 관련연구

2.1 Microsoft InfoPath

MS InfoPath는 워드 프로세스와 같은 WYSIWYG 방식의 전통적인 편집기능에 XML기반의 폼 서식 생성과 동적 디자인을 지원하는 하이브리드(Hybrid) 도구이다. 기업 환경에서 사용되는 다양한 종류의 비즈니스 서식을 쉽게 제작할 수 있는 직관적인 인터페이스와 함께 XML에 기반한 구조적 문서의 편집을 용이하게 한다[9].

특히, XML Schema를 입력으로 받아 초기 디자인 뷰를 구성하고 그 위에 사용자가 다양한 UI 컨트롤을 배치시키면 XSLT를 이용해 DOM tree를 동적으로 구성하여 디자인 뷰에 반영해 준다. 또한, HL7 임상진료문서 Schema와 같은 표준적 XML 스키마뿐만 아니라 사용자 정의 스키마에 의한 검증(validation) 기능이 탁월하여 보다 신뢰성 있는 XML문서를 제작할 수 있게 해준다.

이러한 InfoPath를 이용한 Microsoft사의 노력 중 하나는 웹 기반 의료정보 접근에 대한 통합된 접근을 제공함으로써 임상 전문가들이 필요한 진료 정보를 즉시에 제공해 주는 것이다. 즉, 진료 정보가 여러 곳에 중복 저장되는 것을 막고, 연결된 정보 시스템 간에 필요한 데이타는 XML기반으로 단순화시킴으로써 진료 데이타에 대한 손쉬운 접근과 교류가 용이한 시스템을 개발하는 것이 그 목적이다.

의료분야에서의 InfoPath 활용 예 중 하나는 임상진료문서 스키마를 이용한 Progress Note와 Pharmacy Order와 같은 진료 서식의 생성을 들 수 있다. 즉, InfoPath를 이용해 임상진료문서 스키마를 포함한 HL7

SOAP(Subjective, Objective, Assessment, Plan) 포맷을 따르는 Progress Note와 Pharmacy Order를 위한 폼 템플릿(.xsn)파일을 정의한 다음 이것을 네트워크 또는 로컬 시스템을 통해 InfoPath가 읽어 들이게 한다. InfoPath는 읽어 들인 템플릿 파일을 로컬 캐시에 보관하면서 사용자가 정보를 입력할 수 있는 여러 형태의 가시적인 폼을 화면에 구성해 보여준다. 사용자가 폼에 데이타를 입력하면 임상진료문서 스키마에 따른 값의 유효성 여부를 직관적으로 판단할 수 있게 도와주며 최종적으로 완성된 문서는 XML기반의 임상진료문서로 저장된다. 이렇게 저장된 임상진료문서를 Web Service를 이용해 전송하거나 MS SharePoint Service 2.0 form library를 포함한 기타 다른 back-end data source에 저장할 수 있다.

2.2 IBM CDA Builder API

IBM CDA Builder API[10]는 오픈 소스로 진행되고 있는 HL7 애플리케이션프로그래밍 인터페이스인 caAdapter[11]의 복잡성과 난해함 때문에 겪는 개발자들의 어려움을 해소해 준다. 임상진료문서 생성에 필요한 데이타 수집과정을 간소화 하고 API사용을 단순화하기 위해 내부적으로는 HL7 APIs를 포함하면서 개발자가 좀 더 쉽게 사용할 수 있는 형태의 API 집합을 제공하는 것이 특징이다.

핵심 동작 과정은 임상진료문서 생성에 필요한 데이타수집(data access), CDA Builder Beans객체 생성(Population of CDA Builder Objects) 그리고, CDA Builder 호출(Invoking the CDA Builder)의 3가지 과정으로 요약된다. 즉, 임상진료문서에 포함될 데이타를 레거시 데이타베이스 등 외부의 데이타 소스로부터 얻은 후 IBM CDABuilder패키지가 제공하는 몇몇 Beans 객체의 set 메소드에 데이타를 채운다. 그 다음 CDA Builder 객체를 호출하면 CDA Builder 객체는 caAdapter와 같은 HL7 APIs를 이용해 XML파일 형태의 임상진료문서를 생성하여 클라이언트 애플리케이션에 반환해 주는 구조이다. 이 과정에서 표준 의학 용어 맵핑은 TermResolver와 TermManager API가 그 역할을 담당하고 있어 자동화된 용어 맵핑도 가능하게 한다.

2.3 CDR: Clinical Document Registry Framework

CDR 프로젝트는 본 연구의 선행연구로 진행한 ebXML [12] 기반의 임상진료문서 등록 저장소 프레임워크이다 [7,8].

CDR은 임상진료문서의 송수신과 관련 정보의 효과적 관리를 위해 OASIS ebXML의 RIM(Registry Information Model)과 RS(Registry Service)에 기반하여 임상진료문서의 생명주기(Life Cycle)를 관리하고 참여주체(생성자, 소유자, 관리기관 등)들간의 연관관계와 감사정보를 기록할 수 있는 임상진료문서 등록 저장소 시스템이다.

CDR은 그림 1에서와 같이 분산 레지스트리/ 레파지토리 시스템이다. 즉, CDR은 n개 이상의 레지스트리와 레파지토리로 네트워크에 분산 운영된다. CDR Registry는 사용자와 기관정보를 포함한 임상진료문서 등록과 관련된 메타정보를 유지하고 있으며 임상진료문서에 대한 일차적인 검색요청을 받아들이는 일종의 색인서버 역할을 한다. CDR Repository는 물리적인 저장장치에 임상진료문서를 실질적으로 저장, 보관하는 역할을 한다.

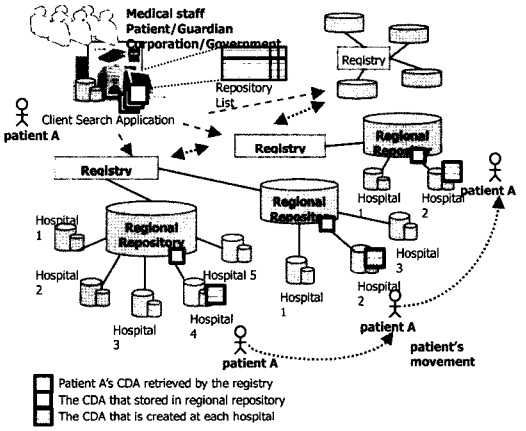

그림 1 Distributed CDR Registry/RepositoryService

CDR에서 임상진료문서는 ebXML의 ExtrinsicObject를 확장하여 관련 정보가 기록된다. ebXML의 ExtrinsicObject는 기본적으로 레지스트리 내부 객체가 아닌 외부에서 제출된 객체 타입을 기술하기 위해 사용되는 객체이다.

임상진료문서의 등록과 검색 등 관련 트랜잭션에 참여하는 액터들 사이의 연관관계는 submitterOf, ownerOf, responsibleFor, employeeOf의 4가지를 정의하고 있다. 이들 각 연관관계는 ebXML의 ClassificationNode 객체로 정의되어 있으며 sourceObject와 targetObject사이의 연관정보는 Association객체에 저장된다. 등록된 임상진료문서에 대해서는 HTTP/SOAP Binding에 의한 질의와 검색기능을 제공하고 프리젠테이션 뷰의 구성은 폴더에 등록된 하나 이상의 XSL파일에 대한 XSLT처리로 동적 뷰를 제공한다.

2.4 IHE-XDS: IHE Cross Enterprise Document Sharing

IHE(Integrating the Healthcare Enterprise)는 미국 RSNA(Radiological Society of North America)와 HIMSS(Healthcare Information and Management Systems Society)에 의해 1998년에 설립된 비영리 단

체로 표준을 개발하는 것이 아니라 기 개발된 표준에 대해서 의료정보 관련 시스템 개발에 필요한 적합한 표준을 선택, 제안하는 노력을 하고 있다. 이러한, IHE의 노력에 따른 결과물은 IHE Technical Framework이라는 이름으로 매년 발행, 갱신되고 있다[13]. 현재 IHE TF는 Radiology, Cardiology, Laboratory와 같은 몇 가지 도메인 영역의 유즈케이스(use-case)를 Integration Profile이라는 이름으로, 그리고 부서간(inter-departmental) 또는 기관간(inter-institutional) 시스템 통합에 대해서는 IT Integration(IT-I)이라는 이름으로 다루고 있다. XDS(Cross Enterprise Document Sharing)는 IT-I가 다루고 있는 RID(Retrieve Information for Display), EUA(Enterprise User Authentication), PIX (Patient Identifier Cross-referencing)와 같은 프로파일 중 하나로 의료기관이 공유하고자 하는 임상진료문서가 있을 경우 그것을 공유하기 위한 스펙을 제시하고 있다.

이와 같은, IHE-XDS에서도 ebXML repository를 임상진료문서 저장을 위한 용도로 사용하고 있으며, ebXML registry는 임상진료문서의 검색과 발견을 위한 메타정보 저장을 위해 사용한다. CDR과 같이 XDS 문서도 ebXML ExtrinsicObject로 관련 정보를 기록한다[14]. 또한, XDS에서는 서로 관련된 문서를 그룹핑하기 위해 폴더(folders)를 사용한다. 예를 들면, 진료기간, 증상, 면역 등의 이름으로 폴더를 만들어 관련 문서를 연관시킬 수 있으며, ebXML의 RegistryPackage객체를 폴더를 만들기 위한 용도로 사용한다. OASIS ebXML RIM의 RegistryPackage 객체는 RegistryObject의 인스턴스들을 논리적으로 그룹핑하기 위해 사용되는 객체이다. 또한, XDS는 특정 임상진료문서와 관련된 다수의 문서 자원들을 하나로 묶어서 등록하기 위한 방법으로 XDS Submission Set을 정의하여 사용한다.

3. 기존 연구의 한계

지금까지 살펴 본 기존 연구들은 모두 임상진료문서 공유 체계 마련이라는 직면한 문제의 해결을 위해 나름대로의 접근 법을 제시하고 있다. 하지만, 이들 연구에서도 아직 간과하고 있거나 접근이 부족한 한계는 다음과 같다.

첫째, 임상진료문서와 해당 부속객체들에 대한 참조 문제이다.

사실, 임상진료문서를 등록할 때 해당 문서와 관련된 다른 정보객체도 참조할 수 있어야 한다. 다시 말해, 임상진료문서의 외양을 결정하는 스타일시트(XSL), 문서에 포함된 다양한 이미지(JPEG, PACS 이미지 등), 사용자 정의 스키마 등 다른 정보객체도 참조할 수 있어야 한다는 뜻이다. 임상진료문서 스펙에서는 이러한 정보들이 링크에 의해 참조되거나 임상진료문서 내부에 바이너리형태로 삽입될 수 있다고 설명한다. 하지만, 이러한 경우 대부분의 링크 정보는 기관 외부에 존재하는 객체를 참조하는 경우가 많기 때문에 필요한 정보객체를 모두 찾기가 쉽지 않다. 따라서, 임상진료문서와 관련된 부속정보도 발견할 수 있으면서 문서에 대한 임의 변경이나 변화가 가해지지 않도록 전자적으로 서명되거나 PDF 포맷과 같이 특별한 형식으로 저장될 수 있는 매커니즘이 별도로 필요하다.

MS InfoPath의 경우 폼 템플릿파일이 이와 같은 문제에 대한 해결책을 어느 정도 보여주었지만 InfoPath 클라이언트끼리만 통신해야 하는 한계가 있어 범용성이 떨어진다. IHE-XDS의 경우는 ebXML의 ebMS와 ebRS를 이용하여 다중문서등록(Multiple Document Submission)을 위한 "Register Document Set" 트랜잭션을 정의하고 있지만 단지 복수 개의 임상진료문서등록절차를 설명할 뿐 부속파일에 대한 처리 문제는 제시하지 못하고 있다.

둘째, 임상진료문서에 기록된 환자 정보의 유출 가능성 문제이다.

이것은 물리적 저장소에 저장된 임상진료문서에 기록되어 있는 환자 정보의 유출 가능성을 말한다. IHE-XDS의 경우, 문서 중심의 EHR 서비스를 위한 프로토콜 측면에서의 트랜잭션을 다루고 있지만 임상진료문서에 대한 접근제어(access control)나 전송관련 보안 메커니즘을 특별히 제시하고 있지는 않다. IHE-TF문서에 그러한 보안적 문제들은 EUA(Enterprise User Authentication)와 ATNA(Audit Trail and Node Authentication)의 결합과 같이 하나 이상의 IHE 프로파일의 결합과 확장을 통해 가능하다고 명시하고 있긴 하지만 영속적 저장소에 저장된 임상진료문서의 유출은 다른 문제이다. 물론, 이 문제도 저장소 자체에 대한 보안 문제로 전가할 수도 있겠다. 하지만, 임상진료문서의 경우 임상진료문서 헤더에는 환자의 신상정보(demographic information)를 비롯하여 제공자 정보(의사, 의료기관)가 기록되어 있고, 임상진료문서 바디에는 해당 환자의 진단과 처치, 투약, 수술 정보 등이 기록되어 있다. 때문에, 임상진료문서가 유출될 경우 임상진료문서 헤더에 포함된 환자 정보의 노출이 사생활침해를 비롯한 여러 가지 사회적 문제를 유발할 수 있다. 더군다나, 아직까지 임상진료문서 표준 스펙에서도 임상진료문서에 대한 전자서명이나 컨텐츠 보호와 관련된 내용이 명확하게 제시되어 있지 않은 상황인 만큼 임상진료문서 자체에 대한 안전한 등록과 영속적 저장, 관리 방법에 대한 조명이 별도로 필요하다고 본다.

4. 적하목록의 구성과 헤더와 바디의 분리에 의한 임상진료문서 등록

앞서 기술한 기존연구의 한계를 극복하기 위한 방법으로 적하목록(Manifest)의 구성과 헤더와 바디의 분리에 의한 임상진료문서 등록을 제안한다. CDR은 기본적으로 ebXML의 구조상 여러 유형의 데이타 타입(HL7 CDA, DICOM, JPEG, PDF 등)을 수용할 수 있으며, 임상진료문서에 관해서는 아래 세 가지 관리 유형을 제시하며 선택적으로 사용할 수 있다. 이 중 첫 번째 방법은 선행연구에서 사용된 가장 범용적인 방법이고, 두 번째와 세 번째 방법이 이번에 새롭게 제안하는 임상진료문서 등록방법이다.

- 단일 임상진료문서 등록, 검색(General Process of CDA Registration & Retrieval)
  임상진료문서에 어떠한 물리적인 조작도 가함이 없이 단순히 일반문서와 같이 취급하여 하나의 문서로 등록, 관리한다.
- 적하목록 구성에 의한 임상진료문서 등록, 검색(CDA Registration & Retrieval Using a Manifest Archiving Structure)
  임상진료문서와 관련된 부속 파일(css, xsl, xsd, jpeg 등)이 있을 경우 적하목록(Manifest) 파일을 구성하여 등록한다.
- 임상진료문서 헤더와 바디 분리에 의한 등록, 검색(CDA Registration & Retrieval By Separation Header & Body)
  좀 더 강화된 보안과 익명화와 프라이버시 보호를 위해 임상진료문서를 헤더와 바디로 분리하여 n개 이상의 서로 다른 저장소에 분산 저장한다.

4.1 단일 임상진료문서 등록, 검색

CDR에서의 가장 일반적인 임상진료문서 등록, 검색 유형으로 임상진료문서에 대해 어떠한 물리적인 조작도 가하지 않는다. 이 유형은 CDR의 기본적인 동작 매커니즘으로 설명될 수 있으며 그림 2로 요약할 수 있다.

순서 1~4까지의 과정은 CDA Consumer가 등록소(Registry)를 통해 기 등록된 임상진료문서에 대한 검색을 수행하는 과정이다. 이미 서두에서 주지한 바와 같이 Registry는 색인 서버 역할을 담당하고 Repository는 실질적인 문서를 물리적인 장치에 저장하는 저장소 역할을 한다. 한가지 유의할 것은, 그림상에 Registry의 반환 값은 따로 표시 되어 있지 않지만 Registry는 질의결과로 해당 문서를 보유하고 있는 저장소 목록(repository list)을 반환한다. 결과적으로 User System은 Registry로부터 받은 저장소 목록을 이용해 해당 문서를 검색하게 되는 것이다.

순서 5~8은 임상진료문서 Provider가 임상진료문서를 등록하는 과정이다. 이때, 물리적인 저장소 공간에 저장된 임상진료문서에 대한 메타데이타는 순서 8을 통해 Registry에 등록된다. 여기서 말하는 메타데이타는 ebXML의 ExtrinsicObject를 상속, 확장한 속성과 문서 등록 시 필요한 다수의 연관정보이다. 이러한 메타데이타는 Registry Object의 구성과 검색을 위해 사용된다. 순서 9~11은 순서 5~8에서 등록했던 임상진료문서를 검색하는 과정이다.

4.2 적하목록 구성에 의한 임상진료문서 등록, 검색

앞서 이야기한 바와 같이 임상진료문서를 등록할 때 해당 문서와 관련된 다른 부속객체도 함께 참조할 수 있어야 한다. 그림 3은 이러한 문제와 관련하여 제안하는 적하목록(Manifest) 파일 콘텐트이다. 참고로, CDR에서는 이러한 콘텐트를 JAVA의 JAR[15]를 이용하여 아카이브한다.

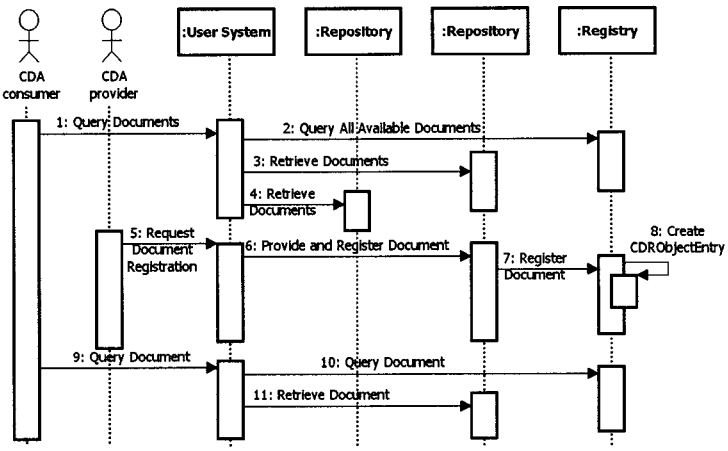

그림 2 CDA document registration & retrieval sequence diagram

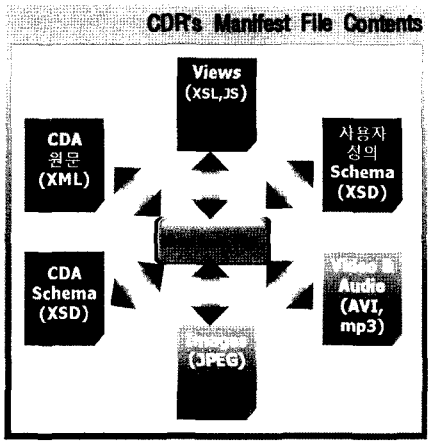

그림 3 CDR의 Manifest 파일 콘텐트

그림을 통해 볼 수 있듯이 임상진료문서 원문과 그와 관련된 부속파일들이 적하목록 파일 안에 함께 포함되어 저장소에 등록된다. 때문에, 일단 클라이언트 시스템에 로드된 적하파일의 임상진료문서는 HTTP요청 단절과 같은 외부적 장애요인과 상관없이 모든 부속객체들을 참조할 수 있다.

4.3 임상진료문서 헤더와 바디 분리에 의한 등록, 검색

앞서 제시한 임상진료문서에 기록된 환자 정보의 유출 가능성 문제에 대한 하나의 해결책으로 다음과 같은 방법을 제안한다. 제안하는 방법은 임상진료문서의 헤더와 바디를 분리하여 암호화하고 하나 이상의 저장소에 분산 저장함으로써 환자 정보와 진단정보의 연관성을 단절하여 개인 정보유출의 가능성을 낮추는 방법이다. 이 경우, 임상진료문서 헤더 즉, 환자의 신상정보를 담고 있는 저장소가 공격 당해 정보가 유출되더라도 자세한 환자의 진단과 처치정보를 알 수 없다. 반대로 임상진료문서 바디 즉, 진단과 처치정보를 담고 있는 저장소가 공격 당해 관련 정보가 유출되더라도 어떤 환자의 진단과 처치정보인지 알 수 없다. 결과적으로 환자의 프라이버시 보호와 환자익명화라는 간접적 효과를 얻을 수 있다.

그림 4는 이와 같은 절차를 Sequence Diagram으로 표현한 것이다. 그림 상의 순서 1~6은 임상진료문서를 헤더와 바디로 분리하여 서로 다른 저장소에 분산 저장하고 있으며, 순서 7~10은 이렇게 분산되어 저장된 헤더와 바디를 찾아와 하나의 임상진료문서로 다시 조합하여 최종적인 결과를 사용자에게 보여준다.

그림 5는 그림 4와 달리 Registry가 아닌 User System이 헤더, 바디 분리 업무를 담당하고 있다. 이는 Registry와 Repository가 객체 등록과 저장이라는 본연의 업무에 충실하게 할 수 있다. 반면, 그림 4에서의 Registry는 임상진료문서 헤더와 바디의 관리를 위해 별도의 컴포넌트를 필요로 하며 많은 요청을 처리해야 하는 부담이 따른다. 하지만, 그림 4와 그림 5 두 유형 모두 임상진료문서 헤더와 바디를 분리함으로써 관련 정보의 연관성을 단절시키고 획득 정보의 가치를 떨어뜨리려는 최종 목적은 동일하다.

그림 6은 이러한 방법이 실제로 어떻게 실현될 수 있는지를 보여주는 구체적인 절차로 CDR에서 제안하는 보안성 강화 전략 과정이다. 여기서 사용된 구성요소들에 대한 역할정의는 아래와 같다.

• Registry

분산된 임상진료문서 헤더와 바디에 대한 저장소로의 색인을 유지하며 SID(Submission ID)를 UserSystem에 발급한다.

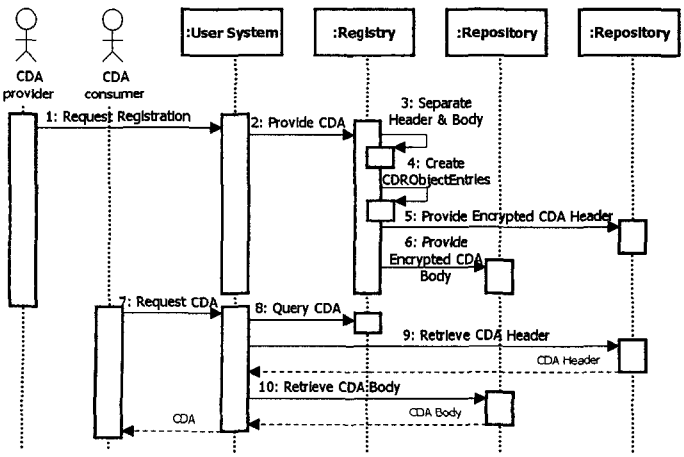

그림 4 CDA헤더와 바디 분리에 의한 등록, 검색 유형1

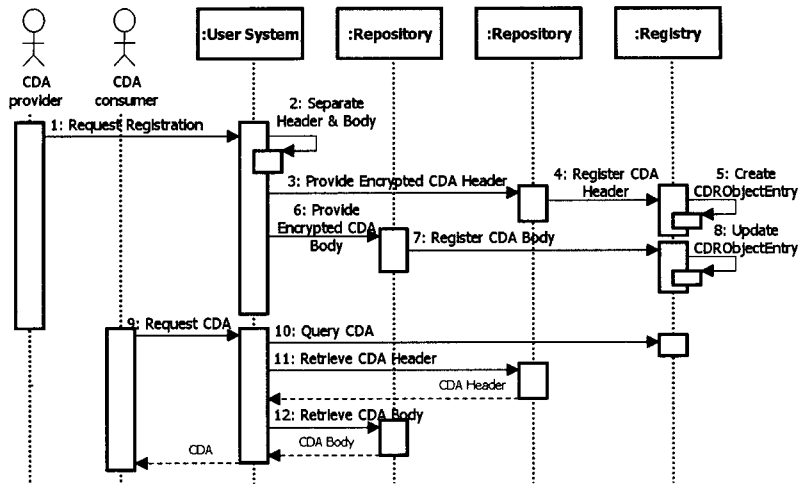

그림 5 CDA헤더와 바디 분리에 의한 등록, 검색 유형2

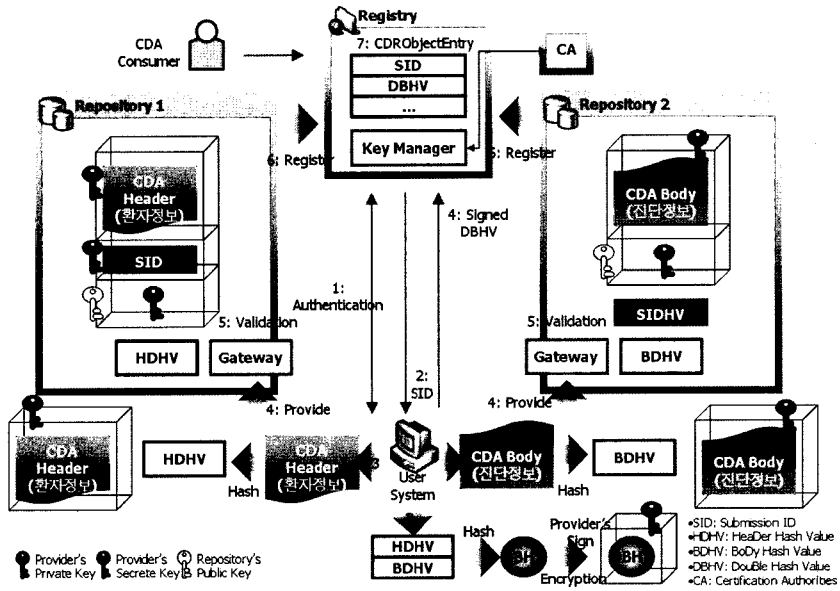

그림 6 CDA헤더와 바디 분리에 의한 보안성 강화 전략

• Repository

분리된 임상진료문서 헤더와 바디를 물리적인 저장장치에 영속적으로 보관한다. 내부의 Gateway를 통해 수신된 자료의 변경 유무 등 적정성을 검증한다.

• UserSystem

CDA Provider가 제공한 임상진료문서를 헤더와 바디로 분리하여 암호화를 수행한 후 저장소에 전송한다.

• KeyManager

Registry 내부에 컴포넌트로 존재하거나 별도 유지되는 키 관리자이다. 외부 CA(Certification Authorities)를 통해 공개키(Public Key), 개인키(Private Key), 비밀키(Secrete Key)등 암호화 키에 대한 검색, 제공 업무를 담당한다.

• Gateway

Repository 내부에 존재하는 구성요소로 UserSystem과 임상진료문서를 주고 받는다.

그림 6에 제시한 보안성 강화 전략의 자세한 알고리즘은 다음 의사코드를 따른다.

```
/* 유저 시스템 측 의사코드 */
granted ← login(userid, password);
if granted = true then {
```

```
  SID ← GetSID_fromRegistry();
  if SID = null then return;
}
// CDA를 헤더와 바디로 분리
separate (CDA);
// 분리된 CDA와 암호화된 정보를 각각의 저장소에 제공
provide(Repository1, EncryptedCDAHeader,
EncryptedSID, EncryptedProvidersSecreteKey, HDHV);
provide(Repository2, EncryptedCDABody, SIDHV,
EncryptedProvidersSecreteKey, BDHV);
// 사용자의 전자서명
provide(Registry, SignedDBHV);

separate (CDA)
  CDAHeader ← getHeader(CDA);
  CDABody ← getBody(CDA);
  HDHV ← hash(CDAHeader);
  BDHV ← hash(CDABody);
  DBHV ← hash(HDHV + BDHV);
  SignedDBHV ← encrypt(CDAProvidersPrivateKey,
  DBHV);
  EncryptedCDAHeader ← encrypt(CDAProviders
  Secrete Key, CDAHeader);
  EncryptedCDABody ← encrypt(CDAProvidersSecrete
  Key, CDABody);
  SIDHV ← hash(SID);
  EncryptedSID ← encrypt(CDAProvidersSecrete Key,
  SID);
  EncryptedCDAProvidersSecreteKey ← encrypt(Repsi-
  torysPublicKey, CDAProvidersSecreteKey);
end separate()
```

1) 사용자인증(Authentication)

먼저, CDA Provider가 User System을 이용 Registry를 통해 사용자 인증을 받는다.

2) SID 발급(Publication of SID)

Registry는 UserSystem이 임상진료문서를 전송할 수 있도록 SID를 발급한다. 이 때, SID는 임의의 UUID이다. SID는 분산된 헤더와 바디의 쌍을 식별하기 위한 고유 식별자로 Registry가 하나의 Registration 트랜잭션을 완료할 수 있는 근거로 사용 된다.

3) CDA 분리(CDA Separation)

임상진료문서 헤더와 바디를 분리하여 암호화한다. HDHV(HeaDer Hash Value)와 BDHV(BoDy Hash Value)는 임상진료문서 헤더와 바디 각각에 대한 해시 값이다. DBHV(DouBle Hash Value)는 HDHV와 BDHV를 하나로 합친 후 그것을 다시 해시하여 얻은 값으로 나중에 헤더와 바디의 결합 및 변경 유무 등의 적정성을 검증하기 위해 사용된다. DBHV는 CDA Provider의 사설키(Private Key)로 전자서명한다. 분리한 헤더와 바디 원문은 CDA Provider의 비밀키(Secrete Key)로 각각 암호화여 포장한다. 그림 상에 표시되지 않았지만 이 비밀키는 저장소의 공개키(Repository's Public Key)로 암호화하여 포장해 둔다. SID의 해시 값 SIDHV을 구한다음 SID 역시 CDA Provider의 비밀키로 암호화하여 포장한다.

4) 저장소에 제공(Provide)

3번 과정에서 준비된 데이타를 임의의 Repository에 제공한다. 그림 상에서는 헤더는 Repository1에 바디는 Repository2에 제공되었다. 이 때, 각 Repository에 제공되는 최종 데이타는 아래와 같다.

- Repository1: 암호화된 헤더원문 | 암호화된 SID | 암호화된제공자의비밀키 | HDHV
- Repository2: 암호화된 바디원문 | SIDHV | 암호화된 제공자의비밀키 | BDHV
- Registry: 서명된 DBHV

여기서 SID를 두 저장소에 서로 다른 값으로 저장하는 이유는 저장소 간 헤더와 바디의 완전한 단절을 위해서다. 즉, 헤더와 바디의 관련성 제거가 목적인 만큼 두 저장소가 같은 SID를 유지하는 것은 바람직하지 않다. 때문에, 해시의 비가역적 특성을 이용하여 서로 다른 값을 저장한다.

```
/* 저장소 측 의사 코드 */
if validate() = true then {
  // Repository 1의 경우
  CDRObjectEntryData.add(SID, HDHV);
  // Repository 2의 경우
  CDRObjectEntryData.add(SIDHV, BDHV);
  register(Registry, CDRObjectEntryData);
}

validate()
  // Repository1의 경우
  CDAProvidersSecreteKey ← decrypt(RepositorysPublicKey,
  EncryptedCDAProvidersSecreteKey);
  SID ← decrypt(CDAProvidersSecreteKey, EncryptedSID);
  CDAHeader ← decrypt(CDAProvidersSecreteKey,
  EncryptedCDAHeader);
  HV ← hash(CDAHeader);
  if HV = HDHV then
    return true;
  else
    return false;
end validate()
```

5) 저장소에서의 검증(Validation)

저장소 Gateway가 UserSystem으로부터 전달받은 데이타를 검증한다. 각 저장소는 자신의 공유키를 이용하여 CDA Provider의 비밀키를 먼저 얻는다. 그 다음 그 비밀키로 헤더와 바디 원문을 차례로 풀고 각각을 다시 해시한다. 다시 해시한 값과 UserSys-

표 1 Evaluations as a document registry & repository

	InfoPath	XDS	CDR
Can support manifest archiving	Yes	N/A	Yes
Support pseudonymization of CDA	No	No	Yes
Document-centric storage/retrieval	Hybrid[1)]	Yes	Yes
Content format agnostic	Yes	Yes	Yes
Can contain multimedia data	Yes	N/A	Yes
Transport level encryption	Yes	Yes	Yes
Implemented	Yes	Yes[2)]	Yes[3)]
Intended for international market	Yes	Yes	Yes
Commercial products available	Yes	No	No

tem으로부터 수신한 값의 비교를 통해 데이타 변경 유무를 검증한다. 상기 의사 코드에서는 Repository1의 예만 보였다.

6) Registry에 등록(Register)

Repository는 validation에 이상이 없는 경우 자신이 수신한 객체(헤더, 바디)와 관련된 메타데이타를 구성하고 Registry에 등록한다.

```
/* Registry 측 의사 코드 */
if (publishedSID = receivedSID) then {
  if hash(ReceivedSID) = receivedSIDHV then {
    if hash(receivedHDHV + receivedBDHV) = DBHV
    then {
      sotre(CDRObjectEntryData);
      commitTrans();
    } else {
      rollBackTrans();
    }
  }
}
```

7) Registry에서의 검증과 트랜잭션 종료(Create CDR-ObjectEntry)

Registry는 각 Repository가 송신하는 SID와 SIDHV를 통해 하나의 Registration 트랜잭션을 완료하고 CDRObjectEntry를 생성한다. UserSystem을 통해 수신한 Signed DBHV는 전자서명된 트랜잭션임을 증명하고 분리된 헤더와 바디의 데이타가 변경되지 않았음을 Registry가 검증할 수 있게 한다.

5. 토론 및 평가

여기서는 지금까지 제안한 내용을 바탕으로 그 실현을 위한 본 연구의 적절성에 대해서 토론, 평가 한다.

표 1은 앞서 소개했던 관련 연구와의 평가 항목이다. 아래 평가 항목들은 본 연구의 주제와 관련성이 있다고 판단된 것을 일부 선정한 것이며 특정 연구나 제품의 절대적 우월성을 평가할 수 없음을 미리 알려둔다.

우선, 평가 대상 3개는 모두 등록, 질의, 검색의 기본 단위로 임상진료문서와 같은 문서(persistent document)를 사용하며 어떠한 형태의 문서 포맷도 수용할 수 있는 문서중심(document-centric) 시스템이다. 단, InfoPath의 경우 편집된 서식에 대한 실시간 DB접근이 가능하고 Share Point 서버와 연동해야만 하기 때문에 Hybrid로 표기했다. 본 연구의 적정성을 강조할 수 있는 항목은 회색으로 표시된 항목이다.

XDS는 임상진료문서 등록에 따르는 부속 파일의 로컬참조를 지원하기 위한 방법을 아직 제시하지 않고 있으며 *InfoPath*는 반드시 *InfoPath*를 클라이언트를 사용해야 하는 단점이 있다. 반면, CDR은 JAR을 Manifest 아카이브 구조로 채택함으로써 웹 환경에서도 부속파일에 대한 로컬참조가 가능하다.

문서 단위 시스템에 있어서 고려해야 할 요소 중의 하나는 해당 문서에 기록된 정보가 영속적(persistent)이어서 노출 시 예기치 못한 문제가 발생할 수 있다는 것이다. 다른 도메인에서의 예를 들면, 국민부채 통계 산출을 위해 대출 계약 문서가 유출 되었을 때 불필요한 계약자 정보까지 함께 노출될 수 있다. 이 경우 계약자의 신상정보를 제외한 계약 내용만 전송할 필요가 있다. 이러한 문서의 익명성 제공 방법은 현재 본 연구에서만 제시하고 있다. 아래는 본 연구의 장점과 단점이다.

• Advantages

- JAR기반 Manifest 아카이빙은 임상진료문서와 관련된 부속자료에 대한 로컬 참조를 가능하게 하여 네트워크 장애와 같은 외부적요인에 대해서도 원활한 Visual을 제공할 수 있다.
- 환자의 신상정보를 담은 임상진료문서 헤더와 진단과 처치 정보를 담은 임상진료문서 바디가 지리적으로 분산된 하나 이상의 저장소에 분리 저장되기 때문에 한층 더 높은 수준의 보안레벨을 제공한다.

1) When used with MS Share point server
2) Work in progress
3) Partially prototype implemented and work in progress

• Disadvantages
- Manifest 파일에 아카이빙된 임상진료문서에 대해서는 별도의 관리 절차가 필요하다.
- 임상진료문서 헤더와 바디의 분리에 따르는 관리적 비용이 많이 들고 관련 트랜잭션의 복잡도가 높아진다.

6. 결론 및 향후 연구방향

지금까지 임상진료문서 공유에 대한 기존연구들을 분석하였고 임상진료문서등록을 위한 적하목록 아카이빙과 임상진료문서 헤더, 바디 분리에 의한 보안성 강화 매커니즘을 새롭게 제안하였다.

적하목록 아카이빙에 의한 임상진료문서 등록은 네트워크 단절과 같은 외부적 요인에 대해서도 부속 자료에 대한 로컬참조를 가능하게 하여 끊김 없는 뷰를 제공해 줄 수 있다. 또한, 임상진료문서 헤더, 바디 분리에 의한 보안성 강화 매커니즘은 환자의 신상정보와 그에 해당하는 진단정보의 관련성을 제거함으로써 제3자가 불법적으로 획득한 정보에 대한 가치를 떨어뜨려 환자 개개인의 익명성까지 제공할 수 있는 특징이 있다.

앞으로는 CDR에서의 적하목록 관리와 CDR에서의 임상진료문서 분배, 복제, 복구 매커니즘에 관한 연구를 계속할 계획이다.

참 고 문 헌

[1] HL7 Clinical Document Architecture Release 2.0., http://xml.coverpages.org/CDA-20040830v3.pdf, http://www.hl7.org/Library/standards_non1.html#CDA

[2] Report from the CEN/ISSS eHealth Standardization Focus Group "Current and future standardization issues in the e-Health domain: Archieving interoperability," Part One: Main text, Draft V4.1, 2004-08-16.

[3] OpenEHR, http://svn.openehr.org/specification/BRANCHES/Release-1.1-candidate/publishing/architecture/overview.pdf, http://www.openehr.org

[4] Introduction to CEN ENV 13606 EHCR Standard, http://www.chime.ucl.ac.uk/work-areas/ehrs/EHCR-SupA/13606-1.htm

[5] Muller ML. Cross-institutional data exchange using the clinical document architecture(CDA), Int J Med Inform;74: pp. 245-256, 2005.

[6] Heimann KU. Discharge and referral data exchange using global standards-the SCIPHOX project in Germany, J Am Med Inform Association; 70: pp. 195-203, 2003.

[7] Kim, IK, Kim, IK, CDR(Clinical Document Repository) framework for electronic health record sharing and medical information network. The 6th CJK Medical Informatics Conference, pp. 65-70: Nagoya, Japan, 2004 November.

[8] Kim, IK, Lee, JY, Kim, IK, Hun, Cho, Kwak, YS, "평생전자건강진료기록을 위한 진료문서 등록저장소 시스템", 대한의료정보학회지 제11권 2호, pp. 199-211, 2005.

[9] Architecture of Microsoft Office InfoPath 2003, http://msdn2.microsoft.com/en-us/library/aa219024(office.11).aspx

[10] IBM CDA Builder API User's Guide, http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/ddqb/HL7_CDA_Builder-UserGuide.pdf

[11] Java SIG HL7 API, http://www.hl7.org/Special/committees/java/index.cfm#WorkProduct

[12] ebXML Specification, Available at: http://www.ebxml.org/specs/index.htm.

[13] IHE IT Infrastructure Integration Profiles, http://www.rsna.org/IHE/tf/ihe_tf_index.shtml

[14] IHE IT Infrastructure Technical Framework, Supplement 2004-2005: Cross-Enterprise Clinical Documents Sharing (XDS), IHE ITI Technical Committee, Trial Implementation Version, August 15, 2004.

[15] Jar File Specification,http://java.sun.com/javase/6/docs/technotes/guides/jar/jar.html

김 일 광

1997년 계명대학교 컴퓨터공학과 졸업 1996년~2001년 이지콤 정보기술 대표 2000년 계명대학교 컴퓨터공학과 석사학위취득. 2007년 경북대학교 컴퓨터과학과 박사학위취득. 2005년~현재 지능형진료지원 및 정보공유시스템개발연구소 근무 관심분야는 이동에이전트시스템, 분산시스템, 그리드 컴퓨팅, 의료정보학

이 재 영

1998년 경북대학교 컴퓨터과학과 졸업 2000년 경북대학교 컴퓨터과학과 석사학위취득. 2005년 경북대학교 컴퓨터과학과 박사과정수료. 2005년~현재 ㈜유케어소프트 개발PM. 관심분야는 패턴인식, 기계학습, 데이타마이닝, 의료정보학

김 일 곤

1980년 서울대학교 수학교육과 졸업. 1988년 서울대학교 전산과학과 석사학위취득 1991년 서울대학교 전산과학과 박사학위취득. 1992년 3월~현재 경북대학교 전기전자컴퓨터학부 교수, 의료정보학과 교수. 2003년 6월~현재 지능형진료지원 및 정보공유시스템개발연구소장. 2005년 1월~현재 한국 ISO/

TC 215 WG2 대표자. 2005년 9월~현재 ISO/TC 215 WG2 Document Registry Framework Project Leader 2005년 12월~현재 EHR 공통핵심기술 연구사업단 제3세부과제책임자. 관심분야는 의료정보학, 에이전트, 서비스 그리드

곽 연 식

1961년 경북대학교 의과대학 졸업. 1972년 미국 Albany의과대학 분자병리 박사학위취득. 1981년~1995년 미국 Case Western Reserve 대학교 병리학 교수 1986년~1995년 미국 Cleveland VA Medical Center CIO. 1994년~1999년 아주대학교 의과대학 임상병리학교실 주임교수 및 병원 전산위원장. 1999년 8월~2002년 8월 경북대학교 의과대학 의료정보학교실 주임교수 및 경북대학교병원 의과학정보센터장. 2002년 9월~현재 경북대학교 의료정보학 초빙교수 2002년 5월~현재 한국HL7 위원장. 2003년 1월~현재 ISO/TC215 의장. 2003년 1월~현재 Deputy Regional Commissioner for Far East Civilian Laboratories, College of American Pathologists Laboratory Accreditation Program. 2003년 10월~2006년 10월 아시아태평양의료정보학회 회장. 2004년 1월~2004년 12월 대한의료정보학회 회장. 2004년~2006년 보건복지부 보건의료기술정책심의위원회 부위원장. 2004년~현재 산업자원부 기술표준원 산업자원표준심의회 위원 2006년 2월~현재 보건복지부 보건의료기술정책심의위원회 위원. 2006년 보건복지부 공공의료기관 정보화 실무위원회 위원장. 2006년~현재 보건복지부 공공의료기관 정보화 운영위원회 위원. 관심분야는 의료정보학, 임상검사실 질 관리