

검사실 정보 교류를 위한 임상문서표준규격의 적용

송준현^o 김일곤, 이성현, 도형호, 예정훈
 지능형 진료지원 및 정보 공유 시스템 개발 센터

jhsong0135@dreamwiz.com, ikkim@knu.ac.kr, xenex@dreamwiz.com, hhdo@knu.ac.kr, 4ngel@naver.com

Adoption of CDA(Clinical Document Architecture) for reporting laboratory results

Joon Hyun Song^o, Il Kon Kim, Sung Hyun Lee, Hyoungdo Do, Jung Hoon Yeah
 Intelligent Health Information Sharing System Development Center

요 약

HITSP(Healthcare Information Technology Standards Panel)은 헬스 케어 관련 산업의 상호 운용성을 위해 일반적으로 수용되고 유용한 표준들을 선별하여 표준 세트를 제공하는 것을 목적으로 한다. HITSP에서는 평생전자건강진료정보(EHR, Electronic Health Record)의 활성화를 위해 첫 번째 해결해야 할 영역으로 검사실 결과 정보 교류를 정하였다. 이에 본 논문에서는 검사실 결과 정보 교류를 위한 방법으로 HITSP에서 제시하는 HL7 버전 2.x 메시지와 CDA 방법 중 인증(authentication) 처리가 가능하고 영속성(persistence)이 있는 CDA 방법을 선택하였다. 또한 CDA를 작성하고 처리하는 방법을 제시하고, 더 나아가 평생전자건강진료정보(EHR)를 위해 CDA를 적용하여 검사실 결과 정보를 교류하여 보았다. 이에 병원과 EHR 시스템의 상호 운용성이 높아져 진료 과정의 효율성을 높일 수 있었고 환자와 의료진에게 양질의 검사 결과 정보를 제공할 수 있었다.

1. 서 론

평생전자건강진료정보(EHR, Electronic Health Record)는 디지털 형태의 분산된 개인 건강 기록으로 평생 지속적인 건강기록의 의미를 가진다. 평생전자건강진료정보(EHR)는 일반적으로 네트워크를 통해서 접근되며 흩어져 있거나 지속성이 있는 건강진료 정보들로 구성된다 [1].

평생전자건강진료정보(EHR)를 위해선 검사실 결과 정보의 교류가 상당히 중요하다. HITSP(Health Information Technology Standards Panel)에서도 EHR을 위한 검사실 결과 정보 교류의 중요성을 인식하고 첫 번째 해결해야 할 영역으로 검사실 결과 정보의 교류를 정하였다 [2].

HITSP은 헬스 케어 관련 산업에서 상호 운용성을 지원하기 위해 유용하고 검증된 표준들을 선별하여 표준 세트를 제공하는 것을 목적으로 한다. HITSP의 상호 운용성을 위한 규격(Interoperability Specification)에서 검사실 정보교류를 위한 기술적인 방법으로 HL7(Health Level 7) 버전 2.x 메시지를 이용하는 방법과 CDA 문서를 통해 하는 방법 두 가지를 제시하고 있다 [2, 3].

CDA(Clinical Document Architecture)는 교환을 목적으로 임상문서의 구조와 의미를 정의하는 HL7에서 제시하는 XML 기반 임상문서표준규격이다. HL7 RIM(Reference Information Model) 모델과 HL7 버전 3 데이터 타입에 기반을 두고 만들어졌다. 2000년 11월에 CDA 릴리즈 1이 ANSI 표준으로 승인되었고, 2005년 5월에 CDA 릴리즈 2가 또한 표준으로 승인받았다. 릴리즈 2는 릴리즈 1에서보다 CDA 각 구성요소들을 명확히 정의하여 활용도를 높였다 [4, 5].

CDA는 사람이 읽을 수 있는 영역(human-readable)을

포함하며 개념의 명확한 표현을 위해 SNOMED CT(the Systematized Nomenclature of Medicine, Clinical Terms)[6], LOINC(Logical Observation Identifiers Names and Codes)[7, 8]를 이용할 수 있다.

본 논문에서는 검사실 결과 정보 교류를 위해 CDA 문서를 생성 및 처리하는 방법을 제시하고 평생전자건강진료정보(EHR)를 위해 CDA를 적용한 검사실 결과 정보 교류 방법을 제시한다.

2. 본 론

CDA의 이해를 위해 먼저 CDA 구조를 소개하고 검사실에서 사용할 CDA 문서 템플릿, CDA 문서 처리 방법을 제시한다. 다음으로 CDA 문서 템플릿과 처리방법을 이용하여 검사실 결과 정보 교류를 위해 적용하는 방법을 제시한다.

2.1 CDA 구조 소개

CDA는 구조적으로 헤더(header)와 바디(body)로 두 부분으로 나뉜다. 헤더는 구조가 엄격히 정의되며 CDA 문서 자체를 위한 내용(문서번호, 문서버전, 작성자 등)과 참여자(participation 즉, CDA 문서와 관련 있는 사람이나 조직을 나타내는 부분) 등의 정보를 표현한다. 바디는 구조가 일반적이고 유연하며 표현하고자 하는 임상내용(처방정보, 투약정보, 검사정보 등)을 표현한다 [4, 5]. CDA는 의미적으로 레벨 1, 2, 3로 구분할 수 있다. 레벨 1은 CDA 헤더부분이고 레벨 2, 3은 CDA 바디 부분이다. 레벨 2는 사람이 읽을 수 있는 부분(narrative block)을 나타내며 레벨 3은 CDA 바디 중 기계적인 처리가 가능한 부분(entry)이다. 레벨 3은 기계적인 처리를

위한 부분이기 때문에 레벨 2의 narrative block과 거의 동일한 의미를 가진다 레벨 2는 반드시 있어야 하지만 레벨 3는 선택적이다.

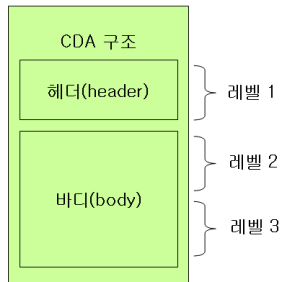


그림 1 CDA 구조

CDA를 이용하여 각종 임상문서 서식을 표현하는데 외국의 사례로는 IHE XDS-LAB의 검사실용 CDA 문서 템플릿[9], 미국의 Care Record Summary를 위한 CDA 문서 템플릿[10], 캐나다의 e-MS(Electronic Medical Summary)를 위한 CDA 템플릿[11] 등이 있다. 세계 각국에서 CDA를 이용하여 기존 임상문서의 형식을 표준화해서 사용하려는 노력이 진행 중이다

2.2 검사실에서 사용할 CDA 문서 템플릿

CDA 문서의 헤더 부분은 필수 요소와 선택 요소가 있다. CDA 문서 템플릿의 헤더 부분으로는 필수 요소는 모두 포함하고 선택 요소 중 일부만을 포함하는 것으로 한다.

CDA 바디 부분은 구조가 유연하기 때문에 구조를 재정의하여 사용할 수 있다. CDA 바디의 컴포넌트 요소(component)가 다시 컴포넌트 요소를 재귀적으로 포함할 수 있는 구조이다. 이에 각종 임상 정보들을 집합적으로 하나의 문서로 만들 수 있다. 예를 들어, 검사 시 발생하는 결과들을 하나의 CDA 문서에 표현할 때 컴포넌트들을 나열하여 표현할 수도 있고 한 컴포넌트 아래 여러 하위 컴포넌트를 두어 표현할 수도 있다

IHE XDS-LAB Supplement[1]에서는 검사실 CDA 템플릿을 제시하고 있는데 바디 부분을 상위레벨과 하위레벨로 분리하여 작성하도록 한다. 상위레벨에는 문서 분류 즉 문서가 나타내고자 하는 주제를 표현하고 하위레벨에서는 세부 내용을 표현하게 한다. 일반적으로 볼 수 있는 CDA 템플릿과는 다르게 하나의 계층을 더 두어 표현하게 한다[10].

일반적인 CDA 문서 템플릿의 바디 부분은 다음과 같다

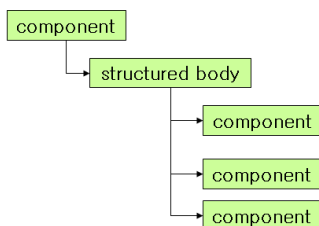


그림 2 CDA 바디 부분

표현하고자 하는 내용을 컴포넌트 요소들의 나열로 표현한다.

IHE XDS-LAB Supplement의 CDA 문서 템플릿에서는

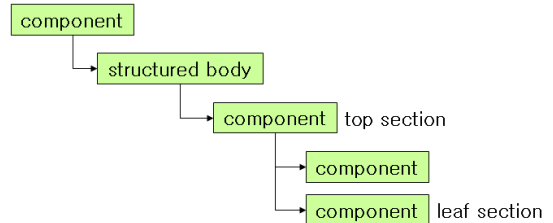


그림 3 IHE XDS-LAB의 CDA 문서 바디 부분

상위레벨(top section)과 하위레벨(leaf section)로 구분하여 상위레벨은 문서의 대분류를 적고 하위레벨에서는 세부 내용을 표현하게 한다. 본 논문에서는 검사실 CDA 문서 템플릿의 바디 부분으로 IHE XDS-LAB Supplement의 CDA 템플릿을 사용한다.

2.3 CDA 문서 처리 방법

검사실 정보 교류를 위해선 기존 병원 정보 시스템의 검사실 데이터를 이용하여 CDA 문서를 생성 및 처리하는 방법이 필요하다. 이에 CDA 문서를 생성(building), 해석(parsing), 검증(validation)하는 방법을 제시한다.

2.3.1 CDA 문서 생성(building) 및 해석(parsing) 방법

CDA 문서는 XML 문서이기 때문에 그 요소와 속성들을 정확히 파악하여 에디터를 이용하여 수동으로 작성할 수 있다. 하지만 검사실에서 발생하는 결과들을 자동으로 생성하기 위해서는 CDA 문서를 생성하는 방법이 필요하다.

CDA 문서를 자동으로 생성하기 위해서 본 논문에서는 자바의 JAXP(Java API for XML Processing)기술을 이용한다. JAXP는 XML 문서를 처리하기 위한 자바 API(Application Programming Interface)이다[13]. CDA 문서에 필요한 요소(element)와 속성(attribute)들을 JAXP를 이용하여 하나씩 추가하면서 CDA 문서를 작성한다. CDA 문서 작성 시 HL7에서 제공하는 CDA XML 스키마에 부합하도록 한다.

CDA 문서를 생성하는 방법에는 JAXP 이외에도 JAXB(Java Architecture for XML Binding)[14], javaSIG API(HL7 Java Special Interested Group API)[15] 등의 여러 가지 방법이 있다. JAXB는 CDA XML 스키마를 자바 클래스로 맵핑하여 XML 문서 생성 시 일반 자바 프로그래머들이 자바 클래스를 다루듯이 CDA 문서를 쉽게 생성할 수 있게 한다. 또 다른 방법으로 javaSIG API를 이용할 수 있다. javaSIG API는 HL7 java SIG(Special Interested Group)에서 내놓은 HL7 V3 메시지 즉, RIM(Reference Information Model)[16]를 처리할 수 있는 오픈 소스이다. CDA도 HL7 V3 메시

지처럼 MDF(Message Development Framework)[17]를 통해 나온 RIM 모델을 활용하고 있다. 이에 RIM 모델을 처리할 수 있는 javaSIG API를 사용하면 CDA 문서를 레벨 1, 2 부분뿐만 아니라 레벨 3 부분까지 쉽게 작성 가능하다. javaSIG API를 이용하면 레벨 3 부분까지 쉽게 작성해낼 수 있는 장점이 있으나 API의 불안정성으로 인해 본 논문에서는 CDA 문서 생성 시 JAXP 기술을 이용하였다.

아래는 JAXP 기술로 CDA 문서를 작성하는 코드를 보여주는 그림이다.

```
// ClinicalDocument
Element clinicalDocument = (Element) document.createElement("ClinicalDocument");
clinicalDocument.appendChild(clinicalDocument);

clinicalDocument.setAttributeNode(document.createAttribute("xmlns"));
clinicalDocument.setAttribute("xmlns", "urn:hl7-org:v3");
clinicalDocument.setAttributeNode(document.createAttribute("xmlns:voc"));
clinicalDocument.setAttribute("xmlns:voc", "urn:hl7-org:v3/voc");
clinicalDocument.setAttributeNode(document.createAttribute("xmlns:xsi"));
clinicalDocument.setAttribute("xmlns:xsi", "http://www.w3.org/2001/XMLSchema");
clinicalDocument.setAttributeNode(document.createAttribute("xsi:schemaLocation"));
clinicalDocument.setAttribute("xsi:schemaLocation", "urn:hl7-org:v3 CDA

// ClinicalDocument/typeId
Element typeId = (Element) document.createElement("typeId");
typeId.setAttributeNode(document.createAttribute("root"));
typeId.setAttribute("root", "2.16.840.1.113883.1.3");
typeId.setAttributeNode(document.createAttribute("extension"));
typeId.setAttribute("extension", "POCD_HD000040");
clinicalDocument.appendChild(typeId);

// ClinicalDocument/templateId
Element templateId = (Element) document.createElement("templateId");
templateId.setAttributeNode(document.createAttribute("root"));
templateId.setAttribute("root", "2.16.840.1.113883.10000.10000");
clinicalDocument.appendChild(templateId);
```

그림 4 CDA 문서 작성 코드

아래 그림은 작성된 CDA 문서의 헤더와 바디 부분이다. 헤더는 문서에 대한 기본 정보, 환자 정보 등을 포함하고 바디는 검사 결과 정보를 포함한다.

```
<ClinicalDocument xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3 CDA.xsd"
<!--
CDA Header
-----
-->
<typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040"/>
<templateId root="2.16.840.1.113883.10000.10000"/>
<id extension="xxxx" root="2.16.840.1.113883.10000.10000"/>
<code code="11502-2" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" display
Laboratory Report Total />
<!-- 병원 임상병리 검사실 -->
<effectiveTime value="20070205"/>
<confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
<languageCode code="en-KR"/>
<versionNumber value="1"/>
<recordTarget>
  <patientRole>
    <id extension="0347106" root="2.16.840.1.113883.10000.10000"/>
    <patient>
      <name>
        <given>+</given>
        <family>김</family>
      </name>
      <administrativeGenderCode code="F" codeSystem="2.16.840.1.113883.10000.10000"/>
      <birthTime value="19690304"/>
    </patient>
  </patientRole>
  <recordTarget>
    <author>
      <time value="20070205"/>
      <assignedAuthor>
        <id extension="IMD03" root="2.16.840.1.113883.10000.10000"/>
        <assignedPerson>
          <name>
            <given>+</given>
            <family>DI</family>
          </name>
        </assignedPerson>
      </assignedAuthor>
    </author>
    <custodian>
      <assignedCustodian>
        <representedCustodianOrganization>
```

그림 5 CDA 문서의 헤더 부분

```
<component>
  <structuredBody>
    <!--
    혈액학
    -----
    -->
    <component>
      <section>
        <code code="18768-2" codeSystem="2.16.840.1.113883.6.1" displayName="CELL
COUNT-DIFFERENTIAL STUDIES"/>
        <component>
          <section>
            <code code="24317-0" codeSystem="2.16.840.1.113883.6.1" displayName="PLATELETS PANEL"/>
            <text>
              <table>
                <thead align="center">
                  <tr>
                    <th align="left" styleCode="Bold">검체명 : CBC(Cor
Count) />
                    <th align="left" styleCode="Bold">검체명 : CBC(Cor
                </thead>
                <tbody align="center">
                  <tr>
                    <td align="left">H-1</td>
                    <td align="left">Hematocrit(39-52%)</td>
                    <td align="left">44.2</td>
                    <td align="left">44.2</td>
                  </tr>
                  <tr>
                    <td align="left">H-2</td>
                    <td align="left">Hemoglobin (13-17 g%)</td>
                    <td align="left">14.4</td>
                    <td align="left">14.4</td>
                  </tr>
                  <tr>
                    <td align="left">H-3</td>
                    <td align="left">R B C Count (4.2-6.3 * 1012/L)</td>
                    <td align="left">465</td>
                    <td align="left">465</td>
                  </tr>
                  <tr>
                    <td align="left">H-4</td>
                    <td align="left">W B C Count(4-10 * 109/L)</td>
                    <td align="left">900</td>
                    <td align="left">900</td>
                  </tr>
                </tbody>
              </table>
            </text>
          </section>
        </component>
      </section>
    </component>
  </structuredBody>
</component>
```

그림 6 CDA 문서의 바디 부분

CDA 문서를 해석하는 방법은 생성 방법의 역순이다. CDA 문서를 전송받게 되면 CDA 문서를 해석하여 내부(메모리)에 유지하고 있는 CDA 구조에 넣게 되고 필요한 용도에 맞게 재사용할 수 있다.

2.3.2 검증(validation) 방법

CDA 문서는 첫 번째 단계로 CDA XML 스키마를 이용하여 CDA 구조에 맞는지 검증할 수 있다. 두 번째 단계로 검사실용으로 재정의된 CDA 문서 템플릿에 맞는지 스키마트론(schematron)을 이용하여 검증할 수 있다 [18, 19].

스키마트론은 XML 문서를 검증하기 위한 규칙이다. 스키마트론은 DTD, XML 스키마 등의 스키마 언어에 비해 쉽고 유연하게 스키마를 만들어 낼 수 있게 한다. 스키마트론은 XPATH, XSLT 기술을 이용하여 작성한다. 또한 스키마트론은 CDA XML 문서를 검증하기 위해 XSLT(eXtensible Stylesheet Language Transformations)를 사용한다. [20, 21].

아래 그림은 스키마트론(IHE_XDS_LAB.sch)을 이용하여 CDA XML 문서(LAB_CDA.xml)를 검증하는 방법을 보여준다.

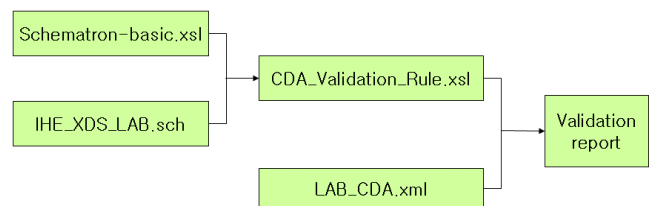


그림 7 스키마트론을 이용한 CDA 문서 검증

첫 번째 단계로 스키마트론 도구

(Schematron-basic.xsl)와 스키마트론(IHE_XDS_LAB.sch)을 XSLT를 이용하여 CDA문서를 검증할 수 있는 규칙(CDA_Validation_Rule.xml)을 생성해낸다. 두 번째 단계로 CDA 문서(LAB_CDA.xml)가 생성된 규칙(CDA_Validation_Rule.xml)에 부합하는지를 검증한다.

2.4 검사실 정보 교류를 위한 CDA 적용

평생전자건강진료정보(EHR)를 위해 검사실 정보 교류를 보여주는 대표적인 시나리오를 작성하였다 다음으로 CDA 문서 템플릿과 CDA 문서 처리 방법을 이용하여 적용하여 보았다.

2.4.1 검사실 정보 교류를 위한 대표 시나리오

환자 김**는 평소 피로함을 많이 느껴 주로 가는 내과에 방문하여 박** 의사를 만났다. 의사 박**는 그녀의 진료 기록과 신체검사 결과를 살펴보고**구에 있는 대학병원의 최** 의사에게 진료 받을 것을 제안하다 박**의 병원의 간호사는 환자 김**에게 진료의뢰서와 진료기록, 이학적 검사 결과를 환자에게 전해준다

박** 의사는 최** 의사가 검사결과들을 요청할 수 있기 때문에, 정규검사를 통해 취득한 혈액과 소변에 대한 검사를 실시하고 그 검사 결과에 대한 차트에 대한 복사본을 환자에게 건네준다.

환자 김**는 수요일 오전에 **구에 있는 대학병원 최** 의사를 만난다. 최** 의사는 환자가 가지고 온 정보들을 살펴보고 진료를 시작한다 최**는 약물을 처방하려고 하지만 해당 약물을 처방하기 전에는 검사실 검사결과에 따른 주의사항들을 파악해야 하기 때문에 임상병리 검사를 할 것을 이야기한다 하지만 김**는 이미 자신의 주치의 박**가 월요일 오전에 검사를 실시했고 그 결과를 가지고 왔다고 이야기 한다 하지만 환자의 실수로 검사 결과지는 알아볼 수 없을 만큼 훼손되어 버렸다 의사 최**는 검사를 시행하려 하지만 검사결과가 나오는데 하루 이상 기다려야 하고 다시 **구로 오기 힘든 환자는 한사코 예전 결과를 참고할 것을 요구한다

<현재 시점에서의 검사정보 활용 예사>

최**는 박**의 병원으로 전화를 한다 하지만 박**의 병원에는 환자가 너무 많이 몰려 바로 검사결과를 보내주기가 어렵다고 이야기한다 최**는 환자 김**에게 정보 요청을 위한 문서에 서명해줄 것을 요청한다 그 후 위임문서와 검사결과지의 복사본 요청에 대한 문서를 박**의 병원으로 팩스를 보낸다. 한 시간이나 지난 뒤에야 필요한 결과가 팩스로 전송된다

<EHR 환경에서의 검사정보 활용 예사>

(EHR 시스템이 있다고 가정한다)

최**는 자신의 업무용 컴퓨터를 이용해 브라우저를 실행하고, EHR 시스템에 로그인 한다 연결되어 있는 리더기에 자신의 스마트카드를 이용해 입력을 하여 공인된 의

료인임을 인증 받는다. 그 후 김**의 이름과 생년월일을 입력하면 몇 가지 신상정보를 볼 수 있다

해당 시스템은 환자 김**가 자신의 주치의 박**와 연관되어 있는 의료인들이 박** 병원 방문기록에 대한 요약정보를 볼 수 있도록 허락했다는 것을 알려준다 (최**는 월요일 방문기록을 선택하고 요약정보에 검사결과에 대한 링크가 있다는 것을 파악한다)

해당 링크를 선택하여 검사정보를 살펴본다 EHR 시스템은 의사 최**의 기록시스템에 검사결과 정보를 업로드 하고, 김**는 추가 검사 없이 과거 검사기록을 바탕으로 한 처방을 받고 병원을 나선다

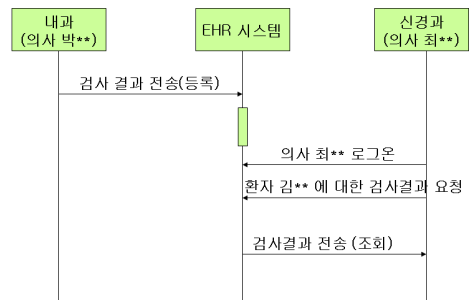


그림 8 교류를 위한 대표 시나리오

검사실 정보 교류가 EHR 시스템을 통하여 이루어지면 시나리오에서와 같이 신뢰할 수 있는 검사실 결과 정보를 제공할 수 있어 환자와 의료진에게 신속하고 정확한 서비스가 가능하다.

2.4.2 평생전자건강진료정보(EHR)를 위한 CDA 적용

검사실 결과 정보를 CDA 문서를 작성하여 EHR 시스템에 전송(등록)할 수 있다. EHR 시스템에 등록된 검사 결과 정보를 CDA 문서로 전송(조회)받을 수 있다. 아래 그림은 검사 결과 CDA 문서를 EHR 시스템에 전송하는 것과 EHR 시스템으로부터 검사 결과를 CDA 문서로 조회하는 것을 보여준다.

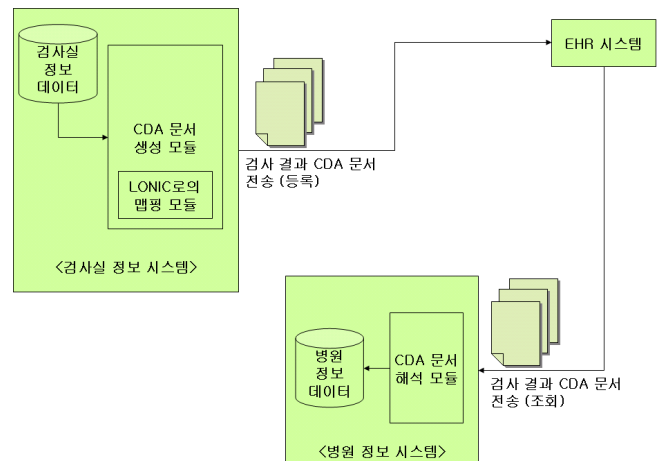


그림 9 EHR 시스템의 CDA 적용

2.5 CDA 문서 작성을 위한 LOINC

검사실 결과 정보의 교류를 위해선 표준 코드의 사용이 반드시 필요하다. 표준 코드에는 LOINC, SNOMED CT 등이 있다. IHE XDS-LAB Supplement에서 LOINC를 사용할 것을 권하기 때문에 본 논문에서도 CDA 문서 작성 시 LOINC를 사용한다. LOINC는 상당히 방대하기 때문에 IHE XDS-LAB에서는 사용빈도와 중요도가 높은 코드들을 분류해 놓았다[7, 12].

병원 정보 시스템에서도 각종 서식지(퇴원 요약지, 수술 기록지 등)에 코드를 사용하는데 편의를 위해 병원 내에서만 통용 가능한 자신만의 코드들을 사용한다. 이러한 코드들은 검사실 정보 교류를 목적으로 하는 CDA 문서에는 적당하지 않기 때문에 병원 정보 시스템의 코드들을 LOINC로의 맵핑 후에 사용하여야 한다[22].

맵핑 작업은 임상검사실 전문가, 간호사, 병원 정보 시스템 담당자의 도움을 받아서 진행하였다. LOINC를 배포하는 Regenstreif Institute는 LOINC로의 맵핑을 쉽게 도와주는 RELMA(Regenstreif LOINC Mapping Assistant) 프로그램도 함께 배포한다. RELMA를 사용하기 위해서는 기존 병원 정보 시스템에서 사용하는 코드를 RELMA가 이해하는 형태(LMOF, Local Master Observation File)로 바꾸어 주어야 한다. 그러면 RELMA는 기존 코드가 어떤 LOINC에 해당되는지의 맵핑 정보를 맵핑 테이블에 기록한다. 이 맵핑 테이블을 이용하여 CDA 문서 작성에 LOINC를 사용할 수 있다.

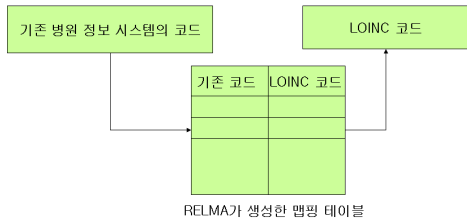


그림 10 LOINC 맵핑 테이블

3. 결 과

본 논문에서는 검사실 정보 교류를 위해 CDA와 HL7 메시지 중에서 CDA를 선택하였다. HL7 메시지는 일시적(transient)이며 트리거 기반(trigger-based)이며 영속적(persistent)이지 못한 반면, CDA 문서는 영속적이며 완전(wholeness)하며 임상주의 인증(clinician authentication)을 담을 수 있고 가독성이 좋기(human-readable) 때문에 CDA를 검사실 정보 교류를 위해 선택하였다.

아래 표는 CDA 문서 처리 방법으로 일반적인 XML 처리 기술(JAXP, JAXB 등)과 javaSIG API를 이용하는 방법의 장, 단점을 보여준다. javaSIG API를 이용하면 CDA 문서의 레벨 1, 2 부분 뿐만 아니라 레벨 3 부분까지 쉽게 작성할 수 있는 장점이 있지만 javaSIG API의 불안정성으로 인해 CDA 문서 처리 시 실제 사용하진 않았다.

표 1 CDA 문서 처리 방법 비교

	일반적인 XML처리 기술(JAXP, JAXB 등)	javaSIG API
장점	<ul style="list-style-type: none"> 세부적인 XML 문서 처리 가능함 	<ul style="list-style-type: none"> CDA 문서의 레벨 3 부분 작성을 쉽게 할 수 있음
단점	<ul style="list-style-type: none"> CDA 문서의 레벨 3 부분 작성 시 상당한 어려움 있음 CDA 문서 생성 시 상당한 시간이 소요됨 	<ul style="list-style-type: none"> javaSIG API의 불안정성으로 인해 실사용이 어려움

다음으로 실제 2차급 병원에 검사실 정보 교류를 위해 적용해 보았다. 기존 2차급 병원의 검사실 정보를 담은 데이터베이스를 분석하여 CDA 문서를 만들어 내었다. 그리고 가상의 EHR 시스템으로 전송(등록)하고 EHR 시스템을 조회하여 검사실 결과 정보를 얻어올 수 있었다. 다음은 테스트에 사용한 검사실 결과 정보를 담은 CDA 문서이다. 문서 전송 후 XSLT 기술을 이용하여 확인한 모습이다.

아래 그림은 기본 혈액검사인 전혈구 검사(CBC, Complete Blood Count) 결과이다.

그림 11 혈액검사를 나타내는 CDA 문서

아래 그림은 일반화학검사 중에서 간기능 검사(Liver Function Test), 신기능 검사(Renal Function Test), 포도당 측정검사(Glucose), 전해질 검사(Electrolyte)의 결과이다.

그림 12 일반화학검사를 나타내는 CDA 문서

검사실 결과 정보 교류를 위해 CDA 문서를 사용하였고 이로 인해 EHR 시스템의 상호 운용성을 높였다 각 병원에서 검사실 결과 정보를 EHR 시스템으로 전송 시 CDA 문서 규격을 사용하기 때문이다 또한 EHR 시스템을 증개자로 하여 병원간 검사실 결과 정보를 교류할 수 있다. 이에 병원간의 상호 운용성도 높일 수 있었다

4. 결론 및 향후 과제

평생전자건강진료정보(EHR)를 위해 검사실 정보는 매우 중요한 비중을 차지한다 이에 HITSP에서도 해결해야 할 첫 번째 영역으로 검사실 결과 정보를 우선적으로 정했다. 검사실 결과 정보 교류를 위해 HITSP에서 제시한 방법이 CDA 문서와 HL7 버전 2.x 메시지이다. 본 논문에서는 HL7 버전 2.x 메시지보다 신뢰성이 높은 CDA 문서를 이용하여 검사실 결과 정보 교류에 적용하는 방법을 제시하였다. 또한 CDA 문서 처리하는 방법을 생성 해석, 검증의 영역으로 나누어 각 방법론을 제시하였다. 검사실 결과 정보 교류의 목적인 평생전자건강진료정보(EHR)에 활용하기 위해 대표적인 시나리오를 제시하고 CDA를 적용하여 보았다. 이로 인해 병원과 EHR 시스템 상호 운용성을 높여 양질의 검사 결과 정보를 제공할 수 있었다. 그리고 CDA 문서의 호환성을 위해 LOINC 사용 시 기존 병원 정보 시스템에서 사용하고 있는 코드를 맵핑하여 사용하는 방법도 제시하였다

검사실 결과 정보 교류를 위한 CDA 문서 작성 시 기계적인 처리를 위한 레벨 3까지 명확히 작성하진 않았다. 레벨 3의 명확한 명세는 아직 분명하게 제시되지 않고 있고, 계속 전문위원회(HL7 Template SIG)에서 토의 중이므로, 레벨 3 부분에 대한 연구를 충분히 수행한 후 에 검사실 정보 교류에 적용하여야 한다

5. 참고문헌

1. 광연식, 평생전자건강진료정보: 규격정의, 종류 및 표준적용, 대한의료정보학회지 제11권, 제1호, 25페이지, 2005
2. HITSP, Healthcare Information Technology Standards Panel, <http://www.hitsp.org>
3. HL7, Health Level 7, <http://www.hl7.org>
4. CDA, Clinical Document Architecture, http://www.hl7.org/Library/standards_mem1.cfm#CDA
5. Robert H. Dolin, Liora Alschuler, Sandy Boyer, Calvin Beebe, Fred M. Behlen, Paul V. Biron, Amnon Shabo, HL7 Clinical Document Architecture, Release 2, Journal of the American Medical Informatics Association, Volume 13, Number 1, Jan/Feb 2006
6. SNOMED CT, the Systematized Nomenclature of Medicine Clinical Terms, <http://www.snomed.org>
7. LOINC, Logical Observation Identifiers Names and

Codes, <http://www.loinc.org>

8. Stanley M. Huff, Roberto A. Rocha, Clement J. McDonal, Georges J. E. De Moor, Tom Fiers, W. Dean Bidgood, Jr. Arden W. Forrey, William G. Francis, Wayne R. Tracy, Dennis Leavelle, Frank Stalling, Brian Griffin, Pat Maloney Diane Leland, Linda Charles, Kathy Hutchins, John Baenziger, Development of the Logical Object Identifier Names and Codes(LOINC) Vocabulary, Journal of the American Medical Informatics Association, Volume 5, Number 3, May/June 1998
9. IHE TF, IHE Laboratory Technical Framework, http://www.ihe.net/Technical_Framework/index.cfm#laboratory
10. Implementation Guide for CDA Release 2 - Level1 and 2 - Care Record Summary (US r e a l m) , <http://www.hl7.org/documentcenter/public/standards/informative/crs.zip>
11. e-MS(Electronic Medical Summary) Clinical Document Architecture Implementation Guide, <http://www.e-ms.ca/documents/CDA%20Implementation%20Guide.pdf>
12. IHE Laboratory Technical Framework Supplement 2006-2007, Sharing Laboratory R e p o r t s (X D * - L A B) , http://www.ihe.net/Technical_Framework/upload/IHE_LAB_TF_Sharing_Lab_Report_XD_Lab_TI_2006_09_14.pdf
13. JAXP, Java API for XML Processing, <http://java.sun.com/webservices/jaxp>
14. JAXB, Java Architecture for XML Binding, <http://jaxb.dev.java.net>
15. javaSIG API, HL7 JAVA SIG에서 배포한 오픈소스, <http://aurora.regenstrief.org/javasig>
16. HL7 RIM, Reference Information Model, http://www.hl7.org/Library/data-model/RIM/modelpage_mem.htm
17. HL7 MDF, Message Development Framework, <http://www.hl7.org/Library/mdf99/mdf99.pdf>
18. XML 스키마, XML Schema, <http://www.w3.org/XML/Schema>
19. 스키마트론, schematron, <http://www.schematron.com>
20. XPATH, XML Path Language, <http://www.w3.org/TR/xpath>
21. XSLT, eXtensible Stylesheet Language Transformations, <http://www.w3.org/TR/xslt>
22. Agha N. Khan, Stanley P. Griffith, Catherine Moore, Dorothy Russell, Arnulfo C. Rosario, Jr., Jeanne Bertolli, Journal of the American Medical Informatics Association, Volume 13, Number 3, May/June 2006