

# 의료 정보 검사코드 표준화를 위한 LOINC 자동 매핑 프레임워크

안후영<sup>†</sup>, 박영호<sup>\*\*</sup>

## 요 약

전자의무기록(Electronic Medical Record, EMR)은 모든 검사 과정이 텍스트 기반의 데이터 형태로 저장되는 의료 분야의 의무기록 시스템을 의미한다. 그러나 국내의 전자의무기록 시스템은 각 의료기관마다 고유한 의료정보검사코드 형태를 이용하여 기록하는 방식으로 정보를 저장하기 때문에 병원 간의 의료 검사 기록 형태들의 공유, 해석, 분석에 많은 문제점들을 가진다. 위의 문제들을 해결하기 위하여 표준화되어있지 않은 병원들의 검사코드들을 LOINC (Logical Observation Identifiers Names and Code)로 표준화하려는 연구들이 많다. 현재까지의 연구들은 로컬 의료정보검사코드를 수동으로 LOINC 로 변환하는 방법이 연구되었다. 또한 대용량 의학 정보들을 다루기에 적절하지 않은 파일 기반에서 코드들을 관리하는 연구들이 이루어져왔다. 기존의 문제점을 해결하기 위하여 본 논문에서는 의료 용어 표준화 알고리즘을 제안하고, 구현하여 해결하였다. 또한, 대표적인 상용시스템이 가졌던 문제점인 검색어를 의사가 직접 생성해야 했던 부분을 LOINC 의 여섯 가지 자동 속성 추출 및 검색어 자동 생성 기능을 구현하여 해결하였다. 또한, 기존의 시스템들이 고려하지 않았던 대용량 데이터의 매핑 부분을 파일 시스템 기반이 아닌 데이터베이스 기반 검색 프레임워크를 구축하였다.

## An Automatic LOINC Mapping Framework for Standardization of Laboratory Codes in Medical Informatics

Hoo Young Ahn<sup>†</sup>, Young Ho Park<sup>\*\*</sup>

## ABSTRACT

An electronic medical record (EMR) is the medical system that all the test are recorded as text data. However, domestic EMR systems have various forms of medical records. There are a lot of related works to standardize the laboratory codes as a LOINC (Logical Observation Identifiers Names and Code). However the existing researches resolve the problem manually. The manual process does not work when the size of data is enormous. The paper proposes a novel automatic LOINC mapping algorithm which uses indexing techniques and semantic similarity analysis of medical information. They use file system which is not proper to enormous medical data. We designed and implemented mapping algorithm for standardization laboratory codes in medical informatics compared with the existing researches that are only proposed algorithms. The automatic creation of searching words is being possible. Moreover, the paper implemented medical searching framework based on database system that is considered large size of medical data.

**Key words:** EMR(Electronic Medical Record, 전자 의료 기록), LOINC (Logical Observation Identifiers Names and Code), Laboratory Code(검사코드), Mapping System(매핑 시스템)

※ 교신저자(Corresponding Author): 박영호, 주소: 서울시 용산구 효창원길 52 숙명여자대학교 명신신관 508호 (140-742), 전화: 02)2077-7297, FAX: 02)710-9704, E-mail: yhpark@sookmyung.ac.kr

접수일: 2009년 1월 28일, 완료일: 2009년 5월 29일  
<sup>†</sup> 준회원, 숙명여자대학교 멀티미디어학과 박사과정 (E-mail: hyahn85@sookmyung.ac.kr)  
<sup>\*\*</sup> 정회원, 숙명여자대학교 멀티미디어학과 교수

## 1. 서 론

전자의무기록은 모든 진료과정이 텍스트 기반의 데이터 형태로 저장되는 의무기록 시스템을 의미한다. 이러한 의무기록의 데이터화는 환자의 진료, 교육, 연구, 의료 기관 등의 측면에서 정보의 접근성을 용이하게 하고, 의료정보의 통합적 관리와 전달이 가능하다는 점에서 유용하다[1-3]. 의료 정보의 공유는 의료정보검사코드의 표준화라는 문제의 해결이 포함되어야 한다. HIS(Hospital Information System)에서 대표적인 진단 검사 코드의 표준화 코드인 LOINC가 있다. LOINC는 HL7(Health Level Seven)을 통한 전자적 의료정보의 표준코드로서, 여러 나라에서 다양한 언어로 제공되고 있다.

RELMA는 LOINC를 관리하는 Regenstrief Institute사에서 개발한 것으로, LOINC 검색을 통해, 사용자의 매핑을 도와주는 프로그램이다. 그러나, RELMA 시스템은 국외 의료정보코드들을 대상으로 개발되었다. 또한 병원들에서 관리하고 있는 기존의 로컬코드들을 검색어로 직접 사용할 수 없으며, 로컬코드의 LOINC 검색을 위하여 의료 지식을 적용하여 LOINC에서 검색될 수 있는 형태로의 검색어를 수동으로 생성한 후 검색을 수행할 수 있는 단점이 있다.

본 논문에서는 의료정보검사코드의 표준화를 위한 LOINC 자동 매핑 시스템의 구조를 제안하고, 의료정보코드 매핑 프레임워크를 개발하였다.

본 논문의 공헌은 다음과 같다.

- 다양한 의료 정보 검사코드를 LOINC로 자동 매핑하는 효과적인 데이터베이스 프레임워크를 제안하였다.
- LOINC의 6가지 대표 속성들을 자동으로 추출하는 알고리즘을 제안하였다.
- 데이터베이스 인덱스를 활용하여 대용량 데이터를 효과적으로 처리할 수 있는 윈도우 응용 프로그램을 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 사용하는 용어들을 정의하고, 3장에서는 로컬 코드들의 LOINC 매핑을 위한 기존의 연구들을 소개한다. 4장에서는 본 논문에서 제안하는 LOINC 매핑 시스템에 대하여 설명하고, 5장에서는 제안하는 시스템의 주요 알고리즘들로 대응코드 생성 자동화 알

고리즘 및 검색 알고리즘과 검색 알고리즘을 소개하고, 알고리즘의 성능을 분석한다. 6장에서는 실험을 통하여 구현한 시스템의 효율성을 증명한다. 마지막으로 7장에서는 결론 및 기대효과를 설명한다.

## 2. 용 어

**용어 1. 로컬코드(Local Code)**는 각 의료 기관에서 의료정보검사 결과를 표기하는 검사 결과 표기법을 의미한다. 본 논문에서는 표준화 되지 않은 국내 병원들의 의료정보검사코드를 로컬코드라 한다.

**용어 2. LOINC(Logical Observation Identifiers Names and Code)**[4]는 진단검사 결과 표현을 위해 사용하는 표준용어 체계이다. Health Level Seven (HL7)을 통한 전자적 의료 정보의 교류에 있어 검사 정보의 표준코드로 자리매김하고 있다[5-8]. 의료 검사 결과를 LOINC 6가지 주요 속성(검사명, 특성, 시간, 검체, 단위, 방법)으로 표기한다.

**용어 3. RELMA (Regenstrief LOINC Mapping Assistant Program)**는 Regenstrief Institute[5]사에서 개발한 대표적인 LOINC 매핑 도구이다. 로컬 코드의 LOINC 검색을 통해, 사용자의 LOINC 변환을 도와주는 프로그램이다.

**용어 4. Analyte/Component(검사명)**[2]은 일반적인 의료 검사의 명칭을 의미하며 연속 검사인 경우 몇 번째인지를 표기하고, 부하 검사인 경우 부하 약물의 용량을 포함한다. 예를 들어 혈당 측정의 경우는 검사명이 glucose가 된다.

**용어 5. Kind of Property (특성)**[2]은 같은 검사명이더라도 검사 결과의 단위나 예민도가 다른 경우 구분하기 위한 항목이다. 예를 들면 호산구(혈액 백혈구 세포의 일종으로 기생충 질환과 알레르기 질환 등에 걸렸을 경우에 늘어남)수 검사에서 이것이 절대치인지 전체 백혈구의 백분율인지를 구분하기 위한 항목이다. 예를 들어, 측정 대상의 질량(mass)을 측정했는 가 부피 (volume)를 측정 했는가에 따라 특성 항목이 달라진다.

**용어 6. Time Aspect (시간)**[2]은 임의 시점에 채취한 검체에 대한 결과인지, 시간을 두고 채취 것인지를 표현하는 것이다.

**용어 7. System (검체)**[2]는 검사의 대상 물질을 표기하는 항목으로서, 예를 들어 소변(urine)이나 혈

청(serum) 등을 의미한다.

**용어 8. Scale (단위)**[2] 는 결과 단위의 종류를 정량, 서수, 명칭 등으로 표기하는 항목이다. 예를 들어 정성적으로 측정 결과를 가지는 경우는 단위 항목이 qualitative를 가지며, 정량적으로 측정 결과를 가지는 경우의 단위 항목은 quantitative를 가진다.

**용어 9. Method (방법)**[2]은 같은 검체를 사용하는 동일한 검사항목이더라도 검사 방법에 따라 유의하게 참고 범위나 예민도가 달라지는 경우를 명시하는 항목이다.

### 3. 관련 연구

본 장에서는 의료정보검사코드의 표준 용어 체계인 LOINC와 대표적인 LOINC 매핑 시스템인 RELMA에 대하여 설명하고, LOINC 표준화를 위한 기존의 연구들을 소개하고, 장점 및 단점을 분석하며, 본 논문과의 차이점을 설명한다.

#### 3.1 LOINC

LOINC (Logical Observation Identifiers Names and Code)는 진단검사 결과 표현을 위해 Regenstrief Institute에서 개발된 표준 용어 체계로 약 41,000여 개의 코드를 포함하고 있다. LOINC 는 의료 검사 결과를 LOINC 6가지 주요 속성으로 표기한다.

LOINC 코드의 6가지 속성은 모두 독립적이므로, 하나라도 다른 값을 가지면, 전혀 다른 의료정보검사 코드가 된다. 이미 여러 나라의 언어로 LOINC 가 제공되고 있고, 그 사용은 국제적이다[9-12].

#### 3.2 RELMA

RELMA 는 로컬코드의 LOINC 검색을 통해, 사용자의 LOINC 변환을 도와주는 프로그램이다. 그러나, LOINC 와 로컬코드 간 개념차이로 인해, LOINC 에 적합한 용어로 별도의 변환이 필요하다는 문제점을 가진다. 특히, 로컬코드를 LOINC 와 매핑하기 위해 로컬코드의 분석을 통해 LOINC 와의 대응 여부를 확인하고 로컬코드가 LOINC 와 대응할 수 없으면, 로컬코드의 변형을 수작업 통해 대응코드를 만든 후 RELMA 에서 코드 매핑 해야 한다[13-16].

#### 3.3 의미 기반 검사코드 표준화 연구

본 논문에서는 연구 [8]를 ‘검사 결과 자동 매핑 방법 연구’라고 부르도록 한다. 위의 연구에서는 의료정보검사코드의 LOINC 로의 자동 변환을 위해서 검사 정보 시스템으로부터 생성되는 데이터들을 논리적 규칙에 따라 용어들의 관계, 유의어 분석, LOINC 매핑의 빈도수 등을 분석하여 관련된 로컬코드와 LOINC 를 자동적으로 매핑 시키는 방법을 사용한다.

#### 3.4 의미 기반 검사코드 표준화 연구

위의 연구는 로컬코드의 표준화를 위하여 서로 다른 병원들의 로컬코드들에 확장된 정의(Extensional definition)들을 사용하여 유사한 확장 정의를 가지는 로컬코드들의 경우는 유사한 LOINC 를 갖는 특성을 활용하여 로컬코드들을 LOINC 에 매핑하는 방법을 제안하고 있다. 그러나 위의 논문은 로컬코드들 자체로 표준화된 의료정보검사코드를 생성하지 못하고, 변환을 위한 수동 작업의 비중이 매우 높은 단점을 가진다.

#### 3.5 로컬 코드 LOINC 자동 변환 연구

위의 연구는 국내의 로컬코드 표준화를 위한 연구로서, 로컬코드와 LOINC 의 의미론적 방법을 토대로 하여 병원들의 로컬 검사 기록 결과를 LOINC 로 전산화를 통한 자동 변환하는 방법에 관한 연구이다 [17-20]. 위의 논문에서는 RELMA 변환 시스템의 수동 변환률을 줄이기 위한 변환 방법을 제안하며 다양한 형태와 종류의 로컬코드들을 LOINC 로 자동 변환[21,22]시키는 다양한 방법을 고안하였으며, 이를 위해 로컬코드로부터 LOINC 의 여섯 가지 속성을 추출하는 방법들을 정의하였다.

## 4. LOINC 매핑 프레임 워크

본 장에서는 LOINC 매핑 시스템에 대하여 설명한다.

#### 4.1 LOINC 매핑 규칙

그림 1은 본 논문에서 제안하는 로컬 코드의 LOINC 매핑 규칙이다.

$$f(X): X \rightarrow Y \quad (1)$$

$$g(Y): Y \rightarrow Z \quad (2)$$

*X : Local Codes of Any Hospitals*  
*Y : Corresponding Codes*  
*Z : Associated in LOINC*

그림 1. LOINC 매핑 규칙

그림 2는 로컬 병원들의 검사정보코드들이 그림 3의 LOINC 매핑 시스템 흐름도를 거쳐 LOINC 로 변환된 표준화된 의료 정보 검사 데이터베이스를 표현한다.

그림 3은 본 논문의 LOINC 매핑 프레임워크 전체 시스템 흐름도이다.

서로 다른 병원들의 다양한 종류와 형태의 로컬코드들은 각 로컬코드 별로 대응코드 변환 과정을 거친 후 LOINC 로 변환되며, 최종적으로는 각 병원 별로 매핑 된 LOINC 테이블들을 일치하는 LOINC 번호로 데이터베이스의 조인(Join) 연산을 거치면, 각 병원들의 로컬코드가 LOINC 번호로 매핑되어 통합된 하나의 Local-LOINC 매핑 데이터베이스가 생성되는 것이다.

#### 4.2 데이터베이스 스키마 및 인덱스 구성

본 연구에서는 데이터베이스 서버로 Microsoft

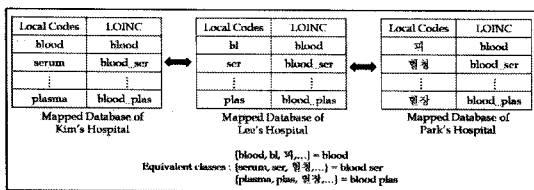


그림 2. 로컬코드의 LOINC 매핑 결과 데이터베이스

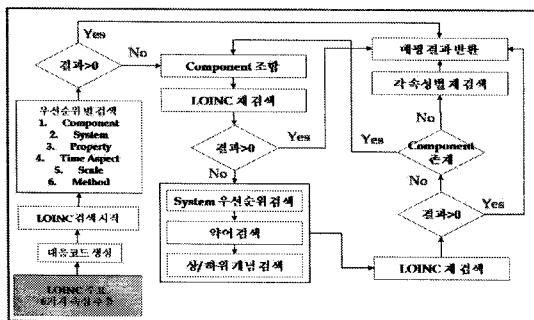


그림 3. LOINC 매핑 시스템 흐름도

SQL Server2000을 사용하였다. 매핑을 위한 로컬코드 데이터로써 진단검사의학과 영역의 로컬코드 1,141개와 핵 의학과 체외검사영역(Radioimmunoassay, RIA)의 로컬코드 98개에서 묶음 코드와 정도 관리 검사용 코드 89개를 제외한 1,150개의 로컬코드를 대상으로 하여 데이터베이스를 구성하였다.

#### 5. 주요 알고리즘

본 장에서는 제안하는 시스템의 주요 알고리즘인 LOINC 의 6가지 속성 추출 알고리즘을 설명한다.

##### 5.1 데이터베이스 스키마 및 인덱스 구성

알고리즘 1은 로컬코드로부터 LOINC 의 Analyte/Component 속성을 추출하는 알고리즘이다. 알고리즘의 입력 값은 로컬코드로의 AnalyteDes 필드이다. AnalyteDes 필드는 아날라이트에 대한 자세한 설명을 적어 놓은 로컬코드의 필드이다. 로컬코드의 아날라이트 속성 값과 LOINC 의 Analyte/Component 속성 값은 변형 없이 그대로 추출하며(라인 4), AnalyteDes 값이 존재하지 않으면, Null을 반환한다(라인 6).

##### Algorithm 1. Extract Analyte/Component from a Local code

Input : LocalAnal = an AnalyteCodeDes (로컬 코드 의 아날라이트 설명)  
 Output : analyte = an Analyte/Component (Analyte/Component of a LOINC )

```

1: Begin
2: {
3:   if exist the AnalyteCodeDes then
4:     return 'AnalyteCodeDes;'
5:   else
6:     return null;
7: }
8: End
    
```

알고리즘 2는 표 12를 기반으로, 로컬코드의 unit 필드로부터 LOINC 의 Property 를 추출하는 알고리즘이다.

**Algorithm 2. Extract Property from a Local Code**

Input : LocalUnit = unit of a local code (로컬코드의 단위 속성)  
 Output : LOINC \_property = A property of LOINC (LOINC 의 property 속성)

```

1: Begin
2: {
3: if not exist the local unit then return null;
4: else compare the local unit with LOINC property
  in table 2
5: {
6: if(LocalUnit=="E.u./dl|"mAU/ml|"KUA/L"
 |"EU/ml|"mU/ml|"uIU/ml"|"mAU/ml|" U/L"
 |" U/ mL") return "ACnc/CCnc"
7: else if(LocalUnit=="mm/hr") return "ARat"
8: else if(LocalUnit=="U") return "A/C"
9: else if(LocalUnit=="mM") return "CAct"
10: else if(LocalUnit=="ng/ml/ hr") return "CRat"
11: else if(LocalUnit=="10E3/ul") return "Fr"
12: else if(LocalUnit=="%") return "Fr/Prctl"
13: else if(LocalUnit=="fl") return "Len"
14: else if(LocalUnit=="mg/24hr|mg/day|pg|ug/24hr")
  return "Mass"
15: elseif(LocalUnit=="gm/dl|gm/L|ng/dl|ng/ml|pg/
  ml|ug/dl|ug/L|ug/ml") return "MCnc"
16: else if(LocalUnit=="mg/mmol") return "MCrto"
17: else if(LocalUnit=="ug/min|mg/24hr|gxylose/5hr")
  return "MRat"
18: else if(LocalUnit=="copies/ml /ul/10E6/ul")
  return "NCnc"
19: else if(LocalUnit=="/HPF/LPF")
  return "Num ber"
20: else if(LocalUnit=="mmhg") return "pres"
21: else if(LocalUnit=="ratio") return "Rto"
22: else if(LocalUnit=="gm/dl|gm/L|ng/dl|ng/ml
  |pg/ml|ug/dl|ug/L|ug/ml") return "MCnc"
23: else if(LocalUnit=="mEq/L|mmol/L|nmol/L|pmol/
  L|umol/L|umol/L|uEq/L|mosm/k")
  return "SCnc"
24: else if(LocalUnit=="nM/mM.cre")
  return "ScrRto"
25: else if(LocalUnit=="umol/h/gHb") return "SRat"
26: else if(LocalUnit=="vol%") return "VFr"
27: else return null;
28: }
29: }
30: End
    
```

알고리즘 3의 입력 값은 bottleName(용기명)과 System 이고, 출력 값은 LOINC 의 System 속성 값

이다. 로컬코드로부터 LOINC 의 System을 추출하기 위해서는 로컬코드의 용기명과 검체명을 입력 받아, 이들을 소문자로 변환한 후(라인 6), 소문자로 변환된 System이 "blood" 속성값을 갖는지 확인한다(라인 7).

**Algorithm 3. Extract System from a Local code**

Input : LocalbottleName = a bottle name of local code (용기명, bottle 코드)

Localsystem = an system of local code(로컬코드의 검체명)

Output : system = an system of LOINC (LOINC 의 검체)

```

1: Begin
2: {
3: if not exist the bottleName and system of local
  code then return null;
4: else
5: {
6: change the system and bottleName to lower
  case
7: if(system has "blood")//검체가 '피'를 포함하고
  있는 경우
8: {
9: if(system has "arterial","heparin coated
  syringe")
10: return "arterial blood";//동맥혈
11: else if(system has "umbilical") return "cord
  blood";// 태반혈
12: else return bottleName;
13: }
14: else
15: {
16: return search System_Trans table;
17: }
18: }
19: }
20: End
    
```

알고리즘 4는 로컬코드로부터 2장의 정의 6인 Time Aspect 속성을 추출하는 알고리즘이다. 입력 변수로 받은 세 개의 변수인 용기명, 검사명, 검체를 확인하여 위의 변수들에 시간 표현 여부를 세 번의 비교를 통해 판단한다. 이는 각 단계에서 최대 3번(min, h, hour, day) 비교하여 시간 표현 값을 추출하게 되므로 O(1)\*3의 상수 시간 복잡도를 얻을 수 있으며, 모두 비교했을 때, 시간 표현이 없는 경우의 O(1)을 추가하여도 결과 시간 복잡도는 O(1)의 상수

시간 복잡도를 보장하게 된다.

**Algorithm 4. Extract Time Aspect from a Local Code**

Input : LocalbottleName = a bottle name of local code (검사에 사용된 용기명)  
 Localanlyte/component = an analyte/component of local code(검사명)  
 Localspecimen = an specimen of local code(로컬코드의 검체)  
 Output : TimeAspect = A time aspect of the LOINC (LOINC의 시간 속성)

```

1: Begin
2: {
3:   if exist the bottleName of local code then
4:   {
5:     find substring(min, h, hour, day)
6:     if exist an expression above
7:       return 'min' or 'h' or 'hr' or 'day'
8:   }
9:   else if exist the anlyte/component of local code then
10:  {
11:    find substring(min, h, hour, day)
12:    if exist an expression above
13:      return 'min' or 'h' or 'hr' or 'day'
14:  }
15:  else if exist the specimen of local code then
16:  {
17:    find substring(min, h, hour, day)
18:    if exist an expression above
19:      return 'min' or 'h' or 'hr' or 'day'
20:  }
21:  else
22:    return Null;
23: }
24: End
    
```

알고리즘 5은 로컬코드로부터 2장의 정의 8인 Scale 속성을 추출하는 알고리즘이다. Scale 속성을 추출하기 위해서는 로컬코드의 Property 값이 추출되었는지를 확인하고(라인 3), 알고리즘 2를 거친 후 추출된 Property 값을 확인하여, 값이 존재하면, 'Qn'을 반환한다(라인 3). 만약 알고리즘 2을 거친 후, Property 값이 추출되지 않았다면, Scale 속성으로 Null을 반환한다(라인 4).

알고리즘 5의 시간 복잡도는 다음과 같다. 알고리즘의 1의 결과 값인 Property 값의 존재 여부에 따라,

값이 존재할 때는 'Qn'을 반환하고, 존재하지 않는 경우에는 null을 반환하므로, Scale을 추출하는 알고리즘도 O(1)의 시간복잡도를 가지게 된다.

**Algorithm 5. Extract Scale from a Local Code**

Input : Localproperty = a property of local code (로컬코드의 검체 측정 단위)  
 Output : scale = a scale of the LOINC (LOINC의 검사 단위)

```

1: Begin
2: {
3:   if exist the property of local code then return 'Qn'
4:   else return null;
5: }
6: End
    
```

알고리즘 6은 로컬코드로부터 Method 속성을 추출하는 알고리즘이다. Method 속성을 추출하기 위해서는 로컬코드의 메소드가 있는 경우와 없는 경우를 구분한다. 만약, 로컬코드에 Method 속성이 있는 경우는 로컬코드의 Method 속성을 반환하고(라인 3), 존재하지 않는 경우는 Null을 반환한다(라인 8). 알고리즘 6의 시간 복잡도는 다음과 같다. 로컬코드의 Method 값의 존재 여부에 따라, 값이 존재할 때는 로컬코드의 Method 속성값을 반환하고, 존재하지 않는 경우에는 Null을 반환하므로, Method을 추출하는 알고리즘도 O(1)의 시간복잡도를 가지게 된다.

**Algorithm 6. Extract Method from a Local Code**

Input : LocalMethod = Method of a local code (로컬코드의 메소드 속성)  
 Output : LOINC\_Method = A Method of LOINC (LOINC의 메소드 속성)

```

1: Begin
2: {
3:   if exist the Method of local code then return the Method;
4:   else return null;
5: }
6: End
    
```

5.2 검색 알고리즘

생성된 대응 코드로부터 LOINC를 검색하는 알고

Component/ Analyte = {a, b, c}  
 Combination = {a,b,c,ab,ac,bc,abc}

Component/Analyte : 대응코드의 검체  
 Combination : 검체코드들의 조합

그림 4. Component/Analyte 조합 생성 규칙

리즘은 다음과 같다. 우선 생성된 대응코드의 Component/Analyte 속성 값들의 조합에 기반하여 LOINC 를 검색한다. Component/Analyte 속성의 조합을 생성하는 방법은 그림 4와 같다.

LOINC 검색은 위의 조합들로부터 LOINC 의 Component/Analyte를 검색하여 검색 결과가 많이 나오는 순으로 결과를 반환한다. 조합의 검색 우선 순위는 가장 많은 속성 조합을 우선으로 검색한다. 예를 들어 그림 4의 조합에서는 abc 조합을 가장 먼저 키워드로 검색하며, 만약 abc키워드의 검색 결과가 전혀 나오지 않는다면 다음으로 ab, bc, ac의 속성 조합들이 검색의 키워드가 되어 검색되며, 마지막으로 a, b, c 가 각으로 검색 결과를 반환하는 것이다. 이러한 검색 방법은 검색하고자 하는 로컬코드와 유사한 코드들의 검색도 가능하도록 하는 장점을 가진다.

만약 검색 결과가 한 개 이상 반환 되지 않을 경우는 System 대응코드가 LOINC 와 일치하는지 검색한다. 만약 System 속성이 일치하는 경우도 없어 검색 결과가 나오지 않는다면, Property 속성이 일치하는지 여부를 검사하여 검색하도록 하였다. 알고리즘 7은 대응코드로부터 LOINC 를 검색하는 알고리즘이다.

**Algorithm 7. Search LOINC**

```

1: Input : component = a component of corresponding code
2: property = a property of corresponding code
3: time = a time of corresponding code
4: system = a system of corresponding code
5: scale = a scale of corresponding code
6: Output : resultList = the result LOINC
7: Begin
8: if exist component & system of corresponding code {
9: if exist property & time & scale{
10: if(search database Keyword table and exist matched attributes) return LOINC }
11: if exist property & scale{
12: if(search database Keyword table and exist matched attributes) return LOINC }
13: if exist time & scale{
14: if(search database Keyword table and exist matched attributes) return LOINC }
15: if exist property & time{
16: if(search database Keyword table and exist matched attributes) return LOINC }
17: if exist property {
18: if(search database Keyword table and exist matched attributes) return LOINC }
19: if exist time{
20: if(search database Keyword table and exist matched attributes) return LOINC }
21: if exist scale {
22: if(search database Keyword table and exist matched attributes) return LOINC }
23: }
24: else return null;
25: End
    
```

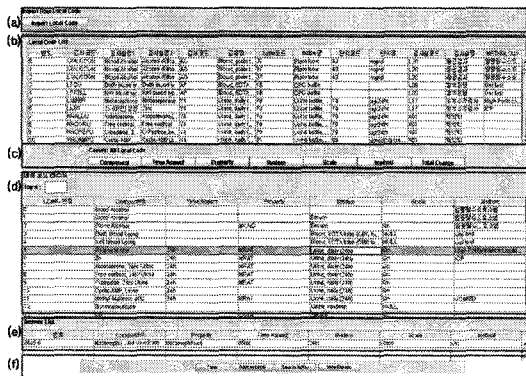


그림 5. LOINC 주요 속성을 추출 및 매핑하는 인터페이스

**6. 알고리즘LOINC 자동 매핑 프레임워크의 구현**

그림 5는 매핑 시스템의 화면이다. 본 장에서는 구현한 시스템의 인터페이스에 대하여 설명한다. 본 시스템은 Eclipse3.2를 이용하여 윈도우 환경 하에서 구동될 수 있도록 구현하였다.

본 시스템은 크게 네 가지 화면으로 구성되며, 매핑 시스템을 시작하는 메인 화면, 기존 로컬코드 데이터로부터 6가지 LOINC 속성을 추출하는 화면, 새로운 로컬코드의 LOINC 변환 및 입력하는 화면, 검색된 LOINC 에 대한 자세한 정보를 보여주는 View Details화면이다.

## 7. 실험

본 장은 LOINC 자동 매핑 시스템의 검색 성능을 두 종류의 실험을 통해 분석한다.

### 7.1 데이터베이스

본 실험에서는 병원의 의료정보검사코드들을 수집하였다. 실험에 사용된 의료정보검사코드들은 진단검사의학과 영역의 로컬코드 1,141개와 핵 의학과 체외검사영역의 로컬코드 98개에서 묶음 코드와 정도 관리 검사용 코드 89개를 제외한 1,150개의 의료정보검사코드를 대상으로 하여 데이터베이스를 구성하였다. LOINC 데이터로는 LOINC Version 2.21[3]를 사용하였으며, LOINC의 추가적인 정보들을 검색하기 위하여 RELMA 시스템에서 사용하고 있는 LOINC의 세부 속성들을 가진 Microsoft Access 파일을 데이터베이스로 변환하여, 데이터베이스를 구축하였다.

### 7.2 수행 환경

실험은 1.00 Gbyte RAM을 가진 Intel® Pentium D CPU 3.40 GHz PC 상에서 Microsoft SQL Server 2005 데이터베이스를 사용하여 실험을 수행하였다. 사용된 비용 측정 기준은 프로그램 수행 시간과 데이터베이스 검색 시간이다.

### 7.3 RELMA 시스템과의 검색 성능 비교

그림 6을 보면, RELMA 시스템의 검색률과 LOINC 매핑 시스템의 검색률의 차이가 약 70% 이상인 결과를 확인할 수 있다.

위와 같은 결과를 보이는 가장 큰 이유는 첫째,

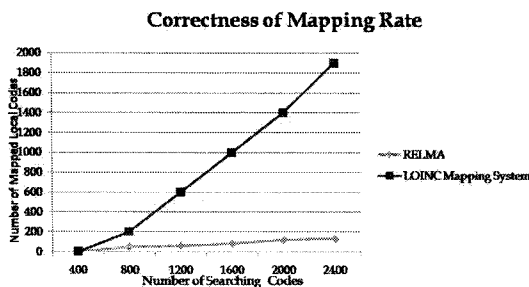


그림 6. RELMA와 LOINC 매핑 시스템의 검색 속도 비교

병원들의 로컬 검사코드를 직접 RELMA 시스템에서 검색하는 경우 로컬코드와 LOINC의 직접 매핑은 거의 불가하므로, RELMA의 매핑 정확률이 매우 낮은 결과를 보인다. 둘째, 제안하는 시스템은 로컬 코드로부터 대응코드를 생성 후, LOINC와 매핑하므로 매핑 정확률이 높은 결과를 보인다.

따라서 구현된 시스템은 로컬 병원들의 의료정보 검사코드의 LOINC 매핑을 위하여 사전 작업으로, 수동으로 대응 코드를 생성한 후, 검색어를 만들어서 검색해야 하는 RELMA에 비하여 검색 속도가 현저히 빠르고, 높은 검색률을 보장함을 보이고 있다. 또한 본 논문에서 제안하는 시스템과 RELMA 시스템의 검색 속도를 비교한 결과 검색 속도에서는 파일 기반의 RELMA와 본 시스템의 속도가 동일한 수준임을 실험을 통하여 알 수 있었다.

### 7.4 인덱스 개수에 따른 검색 속도의 비교

본 실험에서는 인덱스의 개수에 와 데이터베이스 사이즈에 따른 검색 성능을 비교한다. 그림 7은 인덱스 종류에 따른 검색 시간의 차이를 보이는 그래프이고, 그림 8은 인덱스 개수에 따른 데이터베이스 사이즈를 보이는 그래프이다. 그림 9의 그래프의 x-

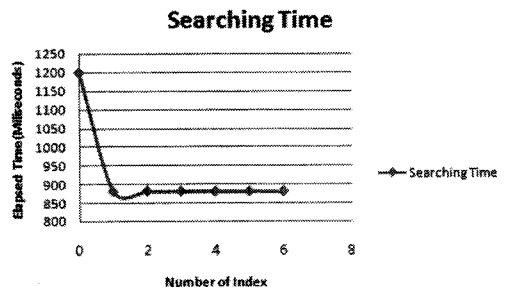


그림 7. 인덱스 종류에 따른 검색 시간 차이

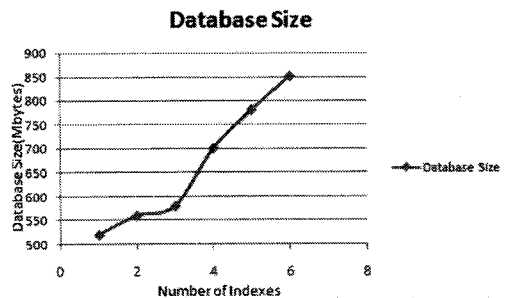


그림 8. 인덱스 증가에 따른 데이터베이스 사이즈



축은 인덱스의 개수를 의미하고, y-축은 검색 시간을 의미한다. 인덱스 개수에 따른 검색 시간의 차이를 보면, 인덱스 개수가 증가하여도 검색 속도는 로그 스케일로 수렴하는 결과를 알 수 있다.

## 8. 결 론

본 논문은 국내 병원들의 표준화 되지 않은 의료정보검사코드 체계의 비 효율성을 해결하고자 로컬코드의 LOINC 매핑 시스템을 구현하였다. 현재까지 대부분의 LOINC 매핑 연구는 LOINC 로의 수동 매핑 방법을 제안하고 있으나, 본 연구에서는 보다 효율적인 의료정보검사코드 표준화를 위해서 LOINC 매핑 과정을 자동화하는 시스템을 구현하였다.

LOINC 매핑 자동화 과정에서는 로컬코드들의 서로 다른 형식으로 인하여 로컬코드와 LOINC 의 직접 매핑이 불가능한 경우가 많은 문제점이 발생하였으나, 본 논문에서는 이를 해결하기 위해 LOINC 6가지 속성을 추출하고 조합하는 모든 과정을 프로그램화하여 자동으로 대응코드를 생성하였고, 생성된 대응코드를 기반으로 LOINC 를 검색하였다.

실험 결과에 따르면, 기존에 가장 인지도 있는 LOINC 매핑 프로그램인 RELMA 와 비교하였을 때, RELMA 에서는 수동 매핑을 사용하지 않으면 전혀 검색되지 않는 국내 병원들의 로컬코드들을 본 시스템에서는 검색할 수 있었으며 검색의 정확성이 향상됨을 검증하였다. 또한 검색의 속도가 RELMA 시스템과 비슷한 결과에 비하여 검색의 정확률은 약70% 높은 결과를 검증하였다. 본 LOINC 매핑 시스템은 향후, 다양한 종류의 로컬코드들을 LOINC 와 자동 매핑 하는데 유용할 것이며, 의료정보검사코드 표준화에 중대한 연구가 될 수 있을 것이라 사료된다.

## 참 고 문 헌

[ 1 ] Emons MF, "Intergrated patient data for optimal patient management," in Proc. *Clin Chem*, the value of laboratory data in quality improvement, pp. 1516-1520, 2001.  
 [ 2 ] Shine Young Kim, Hyung Hoi Kim, Eun Yeoup Lee, Chul L.hune Chang, Hwa Sun Kim and Hune Cho, "Development and Adoption of

a Universal LOINC Mapping Algorithm," *Master Thesis*, Pusan National University, Vol.14, No.2, pp. 123-135, 2008.  
 [ 3 ] LOINC References, <http://www.regenstrief.org>(Updated on Sept. 2008).  
 [ 4 ] Baorto DM, Cimino JJ, Parvin CA and Kahn MG., "Using Logical Observation Identifiers, Names, and Codes (LOINC ) to exchange laboratory data among three academic hospitals," in Proc. *AMIA Annu Fall Symp.*, pp. 96-100, 1997.  
 [ 5 ] RELMA manual, <http://www.regenstrief.org/medinformatics/LOINC/RELMA> (Update d on Aug. 2007).  
 [ 6 ] 2000 Centers for Disease Control and Prevention, "Health Level Seven Specifications for Electronic Laboratory-Based Reporting of Public Health Information," in Proc. *Final Guideline for Implementation with Cancer Registry Comments*, Oct., 1997.  
 [ 7 ] Forrey AW, McDonald CJ and et al., "Logical observation identifier names and codes(LOINC ), Clin Chem, a public use set of codes and names for electronic reporting of clinical laboratory test results," in Proc. *Clin Chem*, pp. 81-90, 1996.  
 [ 8 ] HealthLevelSeven(HL7), <http://www.interfaceware.com/hl7.html?gclid=CN0d1cyyqpQCFRGiiQodRBrkzg>(Updated on Aug. 2007).  
 [ 9 ] Zollo Ka and Huff SM, "Automated mapping of observation codes using extensional definitions," in Proc. *K Am Med Inform Assoc.*, No.7, pp. 586-592, 2000.  
 [ 10 ] Lau LM, Johnson K, Monson K, Lam SH and Huff SM., "A method for the automated mapping of laboratory results to LOINC," in Proc. *AMIA Symp.*, pp. 472-476, 2000.  
 [ 11 ] White MD, Kolar LM and Steindel SJ., "valuation of vocabularies for electronic laboratory reporting to public health agencies," in Proc. *J Am Med Inform Assoc.*, Vol.3, No.6, pp.185-194, 1999.

[12] Yang Y and Chute CG., "An application of least squares fit mapping to clinical classification," in Proc. *Annu Symp Comput Appl Med Care*, Vol.460, No.4, 1992.

[13] Sherertz DD, Tuttle MS, Olson NE, Erlbaum MS and Nelson SJ., "Lexical mapping in the UMLS Metathesaurus," in Proc. *Annu Symp Comput Appl Med Care.*, pp. 494-499, 1989.

[14] Bodenreider O, Nelson SJ, Hole WT and Chang HF., "Beyond synonymy: exploiting the UMLS semantics in mapping vocabularies," in Proc. *AMIA Annu Symp.*, pp. 815-819, 1998.

[15] Rocha RA and Huff SM., "Coupling vocabularies and data structures: lessons from LOINC," in Proc. *AMIA Annu Fall Symp.*, pp. 90-94, 1996.

[16] Fellegi IP and Sunter AB., "A theory for record linkage," in Proc. *J Am Stat Assoc.*, Vol.64, No.328, pp. 1183-1210, 1969.

[17] Jaro MA., "Advances in record-linkage methodology as applied to matching the 1985 census of Tampa," in Proc. *J Am Stat Assoc.*, Vol.84, No.406, pp. 414-20, 1989.

[18] Jaro MA., "Unimatch: A Record Linkage System-User's Manual," *Washington, DC: U.S. Bureau of the Census*, 1978.

[19] Barrows RC Jr, Cimino JJ and Clayton PD., "Mapping clinically useful terminology to a controlled medical vocabulary," in Proc. *Annu Symp Comput Appl Med Care*, pp. 211-215, 1994.

[20] Sowa JF., "Knowledge Representation: Logical, Philosophical, and Computational Foundations," *Pacific Grove, Calif: Brooks/Cole*, 2000.

[21] McDonald CJ, Huff SM, Suico J, and Mercer K (eds.). LOINC user's guide. <http://www.regenstrief.org/LOINC/> (updated on Dec. 2004).

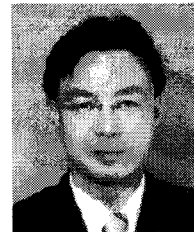
[22] Soo-Young Yoon, Jong-Hyun Yoon, Won Ki Min and et al., "Standardization of Terminology in Laboratory Medicine 1," in Proc. *Korean J Lab Med*, Vol.27, No.2, pp. 151-155, 2007.



**안 후 영**

2003년~2007년 숙명여자대학교  
멀티미디어과학전공  
(학사)  
2007년~2009년 숙명여자대학교  
멀티미디어과학전공  
(석사)

2009년~현재 숙명여자대학교 멀티미디어과학과 박사과정  
관심분야 : 멀티미디어 정보 검색, 대용량 멀티미디어 정보 인덱싱



**박 영 호**

1986년~1992 동국대학교공과대  
학 컴퓨터공학과(학사,  
석사)  
1999년 3월~2005년 8월 한국과  
학기술원 전산학과(공학  
박사)

1993년 8월~1999년 2월 한국전자통신연구원(ETRI)  
교환전송연구단 선임연구원  
2005년 9월~2006년 2월 한국과학기술원 첨단정보기  
술연구센터 연구원  
2006년 3월~현재 숙명여자대학교 이과대학 멀티미  
디아과학과 교수  
관심분야 : 데이터베이스관리시스템, 정보검색, XML,  
Telecommunication System