

심전도 측정 스트림과 의료 클라우드의 연동

조민정, 김진향, 이정훈
 제주대학교 경영정보학과 데이터사이언스학과

whalswjd0820@naver.com, louisy.kim@seerstech.com, jhlee@jejunu.ac.kr

Integration of ECG-generated stream to medical cloud

Minjeong Jo, Jinhyang Kim, Junghoon Lee
 Dept. of Management Information System, Dept. of Data Science,
 Jeju National University

요 약

본 논문에서는 다양한 심전도 패치 장치에서 발생하는 데이터 스트림을 수집하고 이를 FHIR의 표준 포맷으로 변환하여 의료 클라우드에 업로드하는 과정을 구현한다. FHIR 자원으로 변환하는 과정에서 클라우드에서 요구하는 기본 필드를 정확히 채워 무결성을 유지하며 각각의 Observation 자원에 스트림 데이터를 포함시켜 Bundle로 생성한다. 이후 Cloud shell에서 CURL 명령을 수행시켜 수백 개의 심전도 스트림 자원이 JSON으로 표준화된 포맷으로 클라우드에 고속으로 저장된다.

스트림을 FHIR 자원으로 변환하고 Google 클라우드 플랫폼에 저장하는 과정을 구현한다.

1. 서론

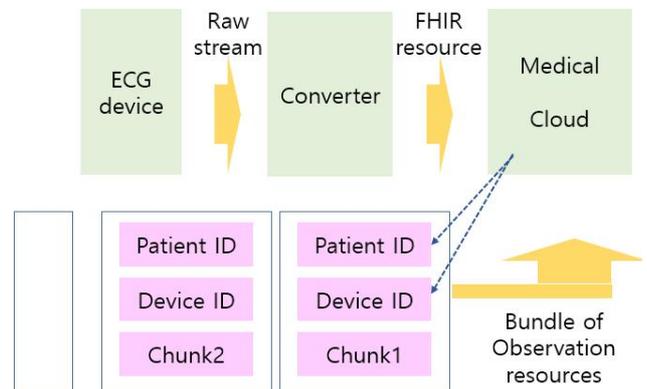
코로나 등의 영향으로 개인용 의료 장비의 성능이 크게 개선됨에 따라 가정용에서 혹은 휴대용으로 헬스케어 모니터링 장비의 부착이 가능해지고 있다. 특히 심전도 패치는 환자 또는 사용자가 몸에 부착하고 이동도 가능하며 통신이 가능한 지역에서는 인터넷 연결 등을 통해 원격 서버에 전송도 가능하다. 몸이 불편한 관광객의 경우도 이런 장치를 부착하고 관광을 할 수 있으며 이 장치가 이상을 발견한 경우 신속하게 보고하여 구급 조치가 행해질 수 있도록 한다.

다양한 장치들이 보급되어 관련 데이터들이 생성되고 이들에 대한 다양한 이해당사자들의 활용이 늘어나게 된다면 데이터의 표준화가 시급한 문제로 대두된다. 의료 도메인에서는 HL7의 FHIR (Fast Health-care Interoperability Resource)에 의해 환자, 측정, 내원 등 다양한 의료 자원들의 표준화가 조금씩 활성화되고 있다[1]. 또, 국내에서도 KR core에 의해 이런 표준의 도입이 시도되고 있다. 더욱이 최근 Google Healthcare API와 같은 클라우드 서비스에서도 FHIR 자원의 저장과 분석을 지원하고 있다. 따라서 심전도 패치 장치들도 이 표준에 따라 데이터를 생성하는 것이 바람직하다.

본 논문에서는 심전도 패치에서 발생하는 원시 스트

2. 프로세스 구성

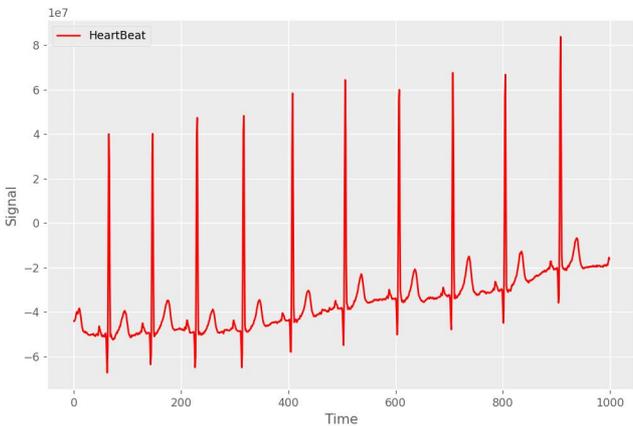
<그림 1>는 본 논문에서 구현하는 스트림 변환구조를 보이고 있다. 먼저 ECG 장치에서 발생하는 Raw 스트림은 4바이트 정수의 연속이며 이는 little endian 방식을 따른다. 디바이스에서 업로드된 파일 혹은 실시간 레코드들에서 4바이트씩 분할하여 정수 스트림을 생성한다. 다음 컨버터는 이 스트림을 적절한 크기로 잘라 복수의 FHIR Observation 자원으로 만든다. 이후 Google Cloud에 Cloud shell을 통해 업로드한다.



<그림 1> 변환 프로세스

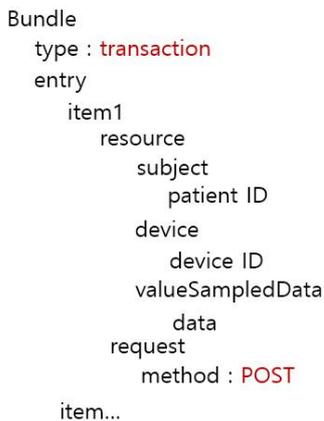
이 과정에서 자원은 OAuth에 기반한 강한 권한 체크와 데이터의 무결성 검사를 거친다. 예를 들어, 심전도 데이터의 환자가 이미 등록되어 있는지 UUID 형태의 레퍼런스가 정확한지, 측정값의 오류가 없는지 등 모두 적절한 데이터만 허가된다. 이때 각 Chunk들은 bundle 객체로 복수의 Observation 자원들의 결합으로 변환된다.

<그림 2>는 raw 스트림의 일부분을 플로팅한 것으로 일반적인 스트림의 경우 주기적으로 피크가 발생한다. PTB-DB와 같은 기존의 학습 스트림의 경우 첫 값이 이 피크에 해당하므로 chunk로 분할할 때 이 지점을 기준으로 설정하여 하나의 자원에 1000 원소 정도가 포함되도록 한다.



<그림 2> raw 스트림의 플로팅

<그림 3>은 FHIR Bundle 자원의 계층구조를 보여주고 있다. 한 엔트리에 여러 개의 Observation들이 소속되며 각 Observation에는 resource와 request가 포함되는데 Bundle에서 request는 반드시 POST, type은 transaction을 가져야 한다.



<그림 3> Bundle에 의한 속성 구조

이후 resource에는 valueSampledData에 측정값들이 스트림의 형태로 포함된다. 각 엔트리는 분할된 chunk의 수만큼 생성되며 FHIR 자원은 기본적으로 JSON 포맷을 따르는데 이는 Python의 Dictionary로 자연스럽게 호환된다.

이 변환 객체가 파일로 생성되면 이를 클라우드에 업로드하며 다음에서 보는 바와 같은 curl 명령에서 권한 토큰 제시, 파일 명세 등을 지정한 후 수행시키면 다양한 무결성 검사 후 클라우드에 저장된다.

```

curl -X POST -H "Content-Type: application/fhir+json; charset=utf-8" -H "Authorization: Bearer $(gcloud auth application-default print-access-token )" --data @EcgBdl.json "https://healthcare.googleapis.com/v1/projects/fhirfhir/locations/asianortheast3/datasets/FirstFHIR/fhirStores/FHIRStorage/fhir"
    
```

<그림 4>는 Google 클라우드에 심전도 측정 데이터가 동시에 저장된 결과를 보이고 있다.

<그림 4> 구글 클라우드 FHIR Viewer 화면

3. 결론 및 추후 과제

본 논문에서 심전도 데이터를 FHIR 자원으로 변환하여 구글 클라우드에 연동하는 과정을 구현하였다. 각 자원은 기본적으로 연속된 스트림인데 특정 위치로 혹은 속성을 갖는 부분을 검색하기 위해서는 적절한 인덱스와 검색 메카니즘이 지원되어야 하며 추후 연구 과제로 진행될 예정이다.

Acknowledgment

본 결과물은 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 3단계 산학협력 선도대학 육성사업 (LINC3.0)의 연구결과입니다.

참고문헌

[1] <https://www.hl7.org/fhir/>