

스트림 데이터 처리를 위한 FHIR 응용의 개발

김소연, 김진향, 강민주, 조윤희, 이정훈

제주대학교 데이터사이언스학과

{carol7378, rkaldswnn, choyunhye05}@naver.com, lousy.kim@seerstech.com,

jhlee@jejunu.ac.k

Development of a FHIR application
for stream data processingSoyeon Kim, Jinhyang Kim, Minju Kang, Yunhye Cho, Junghoon Lee
Dept. of Data Science, Dept. of Computer Education
Jeju National University

요 약

본 논문에서는 클라우드 혹은 개별 서버에 저장된 스트림 기반 FHIR 객체를 연속적으로 읽어와서 디스플레이 하는 응용을 설계하고 구현한다. 파이썬의 웹 인터페이스, 애니메이션, JSON 로딩 라이브러리 기능을 활용하여 기능의 확장성을 기하고 병렬처리를 위해 웹과 디스플레이 부분을 각각의 프로세스로 설계하며 이중 버퍼에 의해 모듈간 동기화를 구현한다.

1. 서론

의료정보의 표준 모델인 HL7의 FHIR (Fast Healthcare Interoperability Resource)의 확산 가능성이 가시화됨에 따라 구글과 아마존 클라우드에서도 FHIR 모델을 지원하고 있다[1]. 또, 각 의료기관마다 독립적으로 개발되어 사용되던 의료정보 모델을 표준 방식으로 재구성하는 방안도 고려되고 있다. 이런 표준 정보모델들이 많은 기관에 의해 수용되면 강력한 개인정보 보호 메커니즘의 전제하에 정보들이 공유될 수 있어서 빅데이터 분석을 촉진함은 물론 다양한 관련 응용들도 출현할 것으로 예상된다. 본 논문에서는 FHIR 표준에 따라 저장된 심전도 데이터에 대해 서버에 이를 요청하고 가시화하는 응용을 설계하고 구현한다.

기본적으로 FHIR 서버들은 보안 정책에 있어서는 OAuth 기능을 탑재하도록 하고 있으며, FHIR 객체들에 대한 생성, 조회, 수정, 삭제 요청을 Post, Get, Put, Delete 등의 웹 메시지로 서버에 전송하고 응답을 받는다. FHIR 자원들은 JSON 포맷으로 기술되어 있어서 파이썬의 딕셔너리와 쉽게 연동되며 파이썬은 풍부한 웹 인터페이스, JSON 처리, 그래프 가시화 및 인공지능 기반 데이터 분석 기능을 제공한다. 따라서 의료분야에 전문화된 플랫폼이나 라이브러리를 사용하지 않더라도 많은 응용들을 작성할 수 있다.

2. FHIR 객체의 특성

FHIR 에서는 환자, 의료진, 검사 기록 등 다양한 자원들의 속성과 상호 연관 관계가 정의되어 있다. 심전도와 같은 스트림 데이터는 하나의 객체가 아닌 일련의 객체들에 나뉘어 저장되는데 <표 1>은 한 객체의 예를 보이고 있다. JSON 포맷으로 기술된 바와 같이 객체 ID, 샘플링 주기, 하한값, 상한값, 타임스탬프 등의 속성과 아울러 유한한 길이의 측정값들의 스트림이 각 객체마다 저장된다. 외부 응용에서는 특정 조건을 제시하거나 객체의 ID를 제시하여 특정 객체를 읽어올 수 있다.

<표 1> Observation 자원의 예

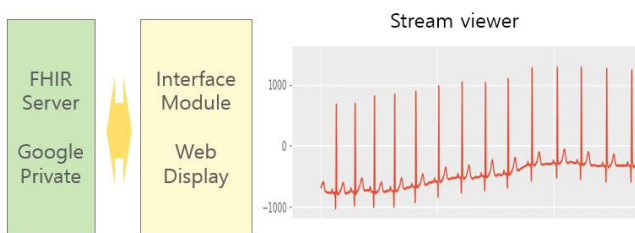
```

"valueSampledData": {
  "origin": {
    "value": 0,
    "unit": "unit",
    "system": "http://unitsofmeasure.org",
    "code": "U"
  },
  "period": 60,
  "factor": 1,
  "lowerLimit": 100,
  "upperLimit": 200,
  "dimensions": 1,
  "data": "100 200 100 200 140 100 200 100 200 140 100
200 100 200 140 100 200 100 200 140 100 200 100 200 140
100 200 100 200 140 100 200 100 200 140 100 200 100 200
140 100 200 100 200 140 100 200"
},
"effectiveDateTime": "2024-09-04T12:00:00Z"

```

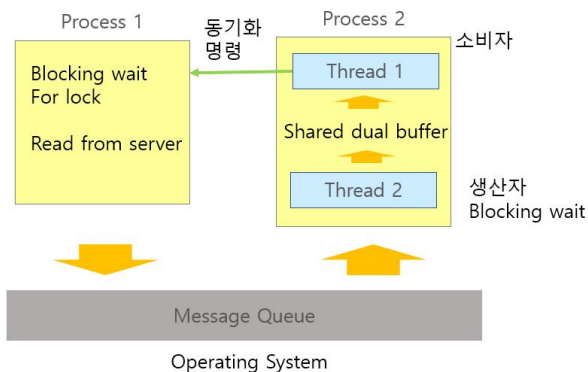
3. 조회 응용의 구성

<그림 1>은 전체적인 FHIR 응용의 구조를 보이고 있으며 FHIR 객체들은 Google의 Healthcare API나 자체 서버에 저장되어 Restful API에 기반한 웹 요청을 통해 Read 연산을 수행한다. 클라이언트들은 조회 및 가시화 대상인 ECG Observation 객체들의 ID들을 미리 파악하고 있을 수 있으며 이들을 순차적으로 읽어와서 다양한 처리를 수행할 수 있다. 결과적으로 클라이언트의 인터페이스 모듈은 한 객체에 포함된 스트림이 다 소비되기 전에 다음 객체를 읽어와 연속적으로 가시화 혹은 분석이 진행될 수 있도록 하여야 한다.



<그림 1> 스트림 객체 가시화의 동작

파이썬 프로그램에서는 요청 생성, 데이터 처리를 위해 requests, scikit learn 등 풍부한 라이브러리를 제공해주고 있긴 하지만 수행속도가 타 언어에 비해 느리고, 쓰레드에 의한 병렬 처리에 어려움이 있다. 스트림 데이터를 처리하기 위해서는 웹 인터페이스 부분과 가시화 부분이 프로세스로 분리된 병렬 수행 구조를 설계하고 구현한다. <그림 2>는 두 개의 프로세스의 연관관계를 보이고 있는데 웹으로부터 Process 1은 Process 2의 요청에 의해 서버로부터 FHIR 객체를 읽어오며 이는 운영체제 내부의 메시지 큐를 통해 Process 2에 전달된다.



<그림 2> 클라이언트 모듈의 구조

기본적으로 모든 객체를 모두 다운로드한 후 디스플레이할 수는 없으므로 Process 2는 animation 라이브러리를 활용하여 한 객체씩 처리하여야 한다. 따라서 한 객체를 디스플레이하면서 다음 객체를 읽어오도록 Process 1에 신호를 보낸다. Process 1은 이 신호를 대기하고 있다가 신호가 들어오면 웹으로 객체를 읽어오고 메시지 큐로 보낸후 다시 대기한다. 이 과정에서 Process 1과 Process 2를 쓰레드로 구현한다면 메시지 교환은 공유 변수를 통해 쉽게 수행될 수 있지만 병렬 수행의 속도 개선을 기할 수 없다.

Process 2는 스트림을 화면에 표시하는 과정에 내부의 메시지 큐에서 다음 객체를 읽어들이어야 한다. 이 과정도 병렬 수행의 속도 개선을 기대할 수는 없지만 원격서버로부터 읽어오는 오버헤드가 수행시간에 포함되는 상황보다는 디스플레이에 주는 영향이 현저히 적다. 디스플레이와 큐로부터의 읽기를 동시에 수행하기 위해 이중 버퍼를 구현하고 있으며 한 버퍼의 디스플레이가 종료되면 버퍼가 교체되고 이전 버퍼에는 웹으로부터 읽어들이는 객체가 채워지게 된다. 이 Process 2에서 쓰레드 기반 수행은 일반적인 생산자-소비자 문제처럼 복수 작업의 동기화 목적으로 활용된다.

3. 결론 및 추후 과제

의료정보 표준의 도입 확대 가능성에 따라 FHIR 기반 데이터를 활용한 응용들이 활발히 개발될 것으로 예상되고 있으며 본 논문에서는 그 예로 심전도 스트림을 FHIR 서버로부터 읽어와 디스플레이하는 응용을 구현하였다. 이 응용들은 일반화된 다양한 라이브러리를 활용하여 구축할 수 있는데, 이와 관련된 추후 과제로 심전도 기반 신원파악과 연계된 데이터 조회, 원격 모니터링에 있어서의 이상치 탐지 등에 관한 연구가 진행될 예정이다[2].

Acknowledgment

본 결과물은 제주대학교 산학협력단의 지원을 받아 수행된 연구결과입니다.

참고문헌

- [1] <https://www.hl7.org/fhir/>
- [2] P. Melzi, et al., "ECG Biometric Recognition: Review, System Proposal, and Benchmark Evaluation," *IEEE Access*, pp. 15555-15566, 2023.