Appunti stage

Riccardo Cambianica

June 25, 2024

Contents

1 Appunti vari

Ho trovato una repository su GitHub, sviluppata da un team dell'azienda olandese Philips, che contiene una applicazione android sviluppata in Kotlin che simula la comunicazione di osservazioni tra alcuni PHD virtuali ed un gateway (un cellulare android). E' inerente al mio ambito di progetto, poiché sviluppa nella pratica uno standard IEEE astratto (**IEEE 1073/10206 - ACOM**). Ho quindi provveduto a convertire alcune delle classi del progetto nel linguaggio di programmazione **Java**, per poi studarne il contenuto, al fine di avere un punto di partenza per lo sviluppo del prototipo dello stage.

2 Studio delle classi tradotte (package generichealthservice)

2.1 Observation e classi connesse

2.1.1 Observation

La classe base per le osservazioni in ACOM. Le variabili presenti sono:

- patientId, una variabile intera che serve a tenere traccia nell'osservazione a quale paziente si riferisce
- supplementalInfo, una lista di tipo ObservationType
- isCurrentTimeline, un booleano utile per il timestamp dell'osservazione

I metodi presenti sono:

- i get per le variabili: getId, getType, getTimestamp, getValue e getUnitCode (capire perché ritorna sempre la costante UNKNOWN, c'è un TODO)
- i "convertitori" da informazioni normali a byte array delle variabili (ghsByteArray, patientIdByteArray, timestampByteArray, flagsByteArray, supplimentalInfoByteArray e getClassByte)

E' presente anche un'enumerazione interna, ObservationClass, che associa ad ogni tipo di osservazione presenti nello standard ACOM.

2.1.2 ObservationType

Enumerazione che contiene tutte le costanti relative al tipo medico di osservazione; a livello concettuale è importante distinguere i tipi di osservazioni con le classi di osservazioni.

2.1.3 ObservationValueType

Enumerazione che contiene 11 costanti con valori in byte che possono essere associati al ObservationType; è presente anche una classe ObservationTypeExtensions che sembra associare i valori delle costanti.

2.1.4 SimpleNumericObservation

Una delle classi che estendono Observation; contiene le variabili

- id, tipo short
- type, tipo ObservationType
- value, tipo float (siamo nel caso di un'osservazione numerica)
- valuePrecision, tipo int
- UnitCode, tipo Unitcode, devo capire in pratica a cosa serve
- timestamp, tipo date

I metodi presenti sono i get delle variabili, getValueByteArray che traduce il valore dell'osservazione in byte array ed infine ObservationClass che ritorna la classe a cui l'osservazione appartiene (capire perché è necessario, invece di utilizzare il valore per distinguere i diversi tipi di osservazioni, probabilmente a causa del formato in byte non è fattibile).

2.1.5 SimpleStringObservation

Osservazioni riportate in un formato leggibile. Uniche differenze con SimpleNumericObservation sono: value è ovviamente di tipo String, non è presente la variabile di tipo UnitCode.

2.1.6 SimpleDiscreteObservation

Osservazione di un o piu eventi "discreti" nel mondo reale dello stato "discreto" di un oggetto. Uguale a SimpleStringObservation, unica differenza il tipo della variabile value, in questo caso di tipo int. Nel progetto sembra non essere mai usata, probabilmente è presente per completezza rispetto al modello ACOM.

2.1.7 SampleArrayObservations

La classe contiene attributi (in particolare l'array di byte e altri float) che consentono la rappresentazione di grafici (tipo elettrocardiogramma credo) Oltre ai soliti metodi get, implementa i set per le variabili, permettendo quindi il controllo totale dei grafici che rappresenta.

2.1.8 CompoundNumericObservation

In ACOM, un'osservazione compound è usata per modellare osservazioni dove una singola misurazione può non essere sufficiente per descrivere totalmente l'osservazione. Nel progetto sono presenti due classi che implementano questa specifica. Come il nome suggerisce, la classe CompoundNumericObservation si occupa di numeri, quindi contiene al suo interno una specifica variabile value, di tipo SimpleNumericValue che potrebbe essere vista come un array di valori numerici.

2.1.9 BundledObservation

Come si può intuire dal nome, questa classe modella un insieme di osservazioni, contenute nella lista value. Resta da capire la sua utilità a livello pratico nel progetto, dato che non è presente nella specifica ACOM.

2.2 SimpleNumericValue

Creata per l'utilizzo in CompoundNumericObservation, modella un valore numerico. Non capisco perché al suo interno è presente una variabile di tipo ObservationType, poiché già presente in CompoundNumericObservation.

2.3 CompoundDiscreetEventObservation

Contiene una lista di ObservationEvent di nome value, i quali modellano degli eventi che possono verificarsi durante la rilevazione delle osservazioni, ad esempio malfunzionamenti o disconnessioni del GHS.

2.4 BluetoothValueFormatType

Enumerazione formata da una serie di costanti in formato esadecimale (?), che vengono contenuti poi dalla variabile value, infine ci sono due metodi get e set.

2.5 TLValue

TLV è l'acronimo di Type-Length-Value, un formato di codifica usato per strutturare e trasmettere dati, solitamente in binario, poiché rende il parsing più semplice e i dati prodotti più piccoli. Due variabili, una di tipo ObservationType ed una di tipo Object.
I metodi sono:

- asGHSBytes: utilizza un oggetto della classe BluetoothBytesParser per incapsulare i dati in byte secondo il formato TLV, richiamando il resto dei metodi della classe
- valueByteArray: analizza la variabile value e in base al suo valore richiama uno dei metodi di conversione da tipi primitivi a byte array
- formatType: restituisce il tipo di formato necessario in base al tipo di valore; attenzione a non confondere la variable type con il tipo di variabile che contiene value
- intToByteArray: usato da valueByteArray per convertire int in byte array
- floatToByteArray: usato da valueByteArray per convertire float in byte array

2.6 TLVObservation

Estende TLValue

2.7 UnitCode

Enumerazione che contiene tutti i codici dei dispositivi medici, composti da valore, simbolo e descrizione

3 Considerazioni sull'analisi delle classi

- Non confondere ObservationType con ObservationClass
- Un evento non è un'osservazione
- Devo ancora capire con certezza l'utilizzo effettivo della classe UnitCode
- Le classi che implementano il formato TLV sono utilizzate esclusivamente per descivere il dosaggio di medicinali, perciò non sono sicuro che siano utili per il progetto di stage

Altre considerazioni sono sparse nelle descrizioni delle singole classi. Riguardo la costruzione di un convertitore da ACOM a FHIR, devo chiarire se è necessario passare dalle classi Java del progetto analizzato. Perché l'abbiamo preso in esame? Proponeva un'implementazione di un formato astratto di rappresentazione dei dati che i PHD possono generare. Ma ora il problema è che la conversione è in sostanza un semplice mapping tra gli attributi dei due formati

Scheletro relazione

- abstract (spiegazione problema interoperabilità?)
- IEEE 11073-10206 (ACOM)
  - a cosa serve
  - funzionamento, compatibilità con fhir
  - osservazioni (devo metterle tutte?)
  - spiegare implentazione in java (con paragrafi dedicati alle specializzazioni implementate)
  - esempi illustrati di un'osservazione ACOM ?
- FHIR
  - a cosa serve (standard dati sanitari elettronici, interoperabilità)
  - risorse, profili (struttura "logica" FHIR)
  - la risorsa osservazione
- Mapping e traduzione (converter)
- conclusioni