Setting Standards for Personal Health Data in the Age of 5G and 6G Networks

Ana Koren[1,*] and Ramjee Prasad[2]

[1]*University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia*
[2]*CTIF GLOBAL CAPSULE, Department of Business Development and Technology, Aarhus University, Herning, Denmark*
E-mail: ana.koren@fer.hr; ramjee@btech.au.dk
[*]*Corresponding Author*

Received 05 October 2023; Accepted 23 March 2024

Abstract

Electronic health records (EHRs) play a vital role in simplifying thorough and effective patient treatment, promoting smooth exchange of information between medical professionals, and enhancing the process of making clinical decisions. With the increasing adoption of sensor-embedded smart wearables and home automation devices, new opportunities arise for innovative solutions in various sectors, such as eHealth. In the age of 5G and 6G, the potential of utilizing user-collected health data becomes vast, promising significant improvements in people's health and well-being. Realizing continuous healthcare access takes a step closer to reality by equipping EHRs to effectively store and interpret data collected by these sensors. This would result in personalized medical services that adhere to standardized practices. This paper presents a comprehensive review of contemporary advancements in the realm of standardization methods aimed at managing personal health data. The study delves into an extensive analysis of state-of-the-art solutions

Journal of ICT Standardization, Vol. 12_1, 47–70.
doi: 10.13052/jicts2245-800X.1213

that have emerged to address the intricate challenges associated with the harmonization and uniformity of personal health information. By systematically examining these cutting-edge approaches, the review elucidates the diverse strategies employed to establish a cohesive framework for organizing, storing, and exchanging personal health data. Furthermore, the review critically evaluates the effectiveness and limitations of each solution in terms of promoting interoperability, safeguarding data privacy, and facilitating seamless data sharing among healthcare stakeholders. Furthermore, this paper then presents an approach to standardize the data by establishing semantic constraints for healthcare data types and proposing a validation procedure to ensure compliance with relevant standards and regulations.

Keywords: Wearable sensors, eHealth, healthcare, Internet of Things, Internet of Medical Things, electronic health record, EHR, 5G, 6G.

1 Introduction

Modern electronic health record (EHR) systems are, in major part, web-oriented, utilizing both client- and server-side elements, employing relational databases, enabling secure data accessing through authorized mechanisms, and facilitating interoperability across a spectrum of healthcare entities such as hospitals, doctors, and pharmacies. With the ongoing rapid growth of the worldwide market for wearable fitness trackers, integrating their data into EHRs in a standardized manner offers substantial potential for improving personalized healthcare provision, encouraging preventive medicine, and propelling advancements in medical research and patient outcomes. In accordance with the findings of a study [1], the wearable fitness tracker industry is poised to experience noteworthy growth of around 80 million euros between 2020 and 2027, displaying a CAGR of 22.49%. Furthermore, as outlined in report [2], the global smartwatch market is anticipated to witness a remarkable surge, with shipments reaching 43.89 million in 2018 and projected to escalate up to 110 million by 2024, manifesting a CAGR of 14.55% during the projection period spanning between 2019 and 2024. Furthermore, according to reports [3] and [4], there is a noteworthy expansion projected in the market for the Internet of Medical Things (IoMT), with an expected CAGR of 20% and a projected value surpassing 455 billion euros by 2025. The upcoming Fifth Generation (5G) and anticipated Sixth Generation (6G) mobile networks are predicted to display enhanced performance attributes, including swifter speeds, decreased latency, heightened reliability,

and expanded accessibility. With these superior capabilities, the technologies of 5G and 6G networks are positioned to substantially amplify communication between IoMT devices and cloud platforms, consequently resulting in enhanced performance and heightened Quality of Service (QoS) in the domain of telemedicine. The integration of these advanced network systems with IoMT and Artificial Intelligence (AI) solutions holds the potential to streamline patient monitoring procedures, rendering them more convenient and effective. Importantly, it should be noted that data can originate from diverse origins, encompassing varied sensor types and multiple manufacturers that employ their distinct data processing algorithms. The initial phase encompasses addressing discrepancies, aligning the data, merging diverse datasets, and consolidating them into a coherent and unified entity. Once the data aggregation is achieved, it becomes imperative to ensure its conformity to a standardized structure, irrespective of its source. The most prevalent EHR standards include Health Level 7 (HL7), with the latest incarnation being HL7 Fast Health Interoperability Resources (FHIR). Section 2 provides a concise overview of pertinent EHR architecture and sensors in healthcare, presenting a snapshot of the latest advanced solutions. Section 3 covers standardization and delineates the suggested definition of semantic data limitations specific to chosen health-related data categories. Section 4 delves into the executed validation procedure, while Section V delivers the final thoughts and conclusions.

2 Related Research: EHR and Sensors in Healthcare

During 1970s, the first EHR was developed at The Regenstrief Institute in Indianapolis after consulting with computer science experts from Purdue University [5] and integrated prescriptions and medication orders, procedures done, laboratory tests, etc. Development of EHRs between 1972 and 1992 included hierarchical or relational databases and where EHRs were deployed on large mainframe computers with limited storage [6]. Web-based EHRs started to appear between 1980 and 1990, as a consequence of hardware becoming more affordable, powerful, compact and the appearance of Internet [7]. As third-party applications were beginning to be used within EHRs, standards were required. In early 2000s, Health Level Seven (HL7) and IEEE P1157 MEDIX served as the primary interface standards. This was necessary to disambiguate data element definitions and use standardized dictionary codes. Eventually, the HL7 standard was updated and expanded to incorporate numerous systems. Between 1991 and 2005, large healthcare providers,

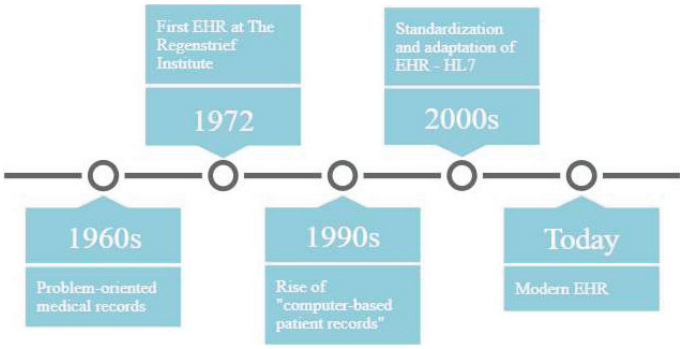

Figure 1 EHR timeline.

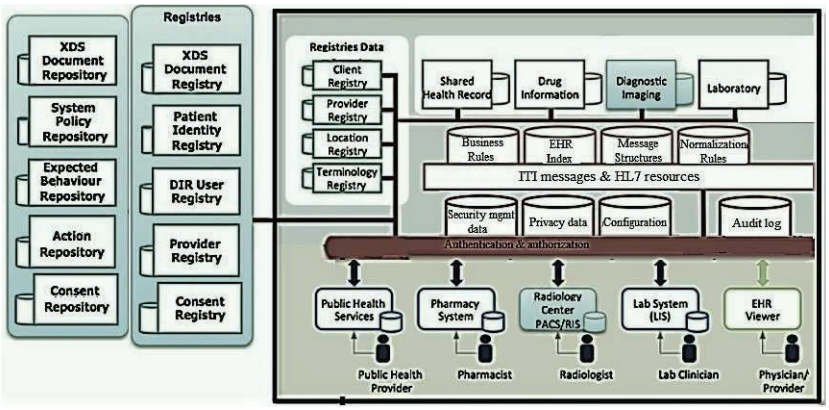

Figure 2 EHR architecture.

academic researchers, and government organizations started to push for the use of EHR, first in the USA and, to lesser extent, Canada, proceeded by the United Kingdom, Switzerland, the Netherlands and Norway [8]. Since 2005, EHR use has been steadily increasing in majority of European countries, Australia, and Asia. Most modern EHRs are web-based with client and server side, use relational databases, provide secure authorization-based data access, and allow interoperation of multiple entities, such as hospitals, physicians, or pharmacies. Figure 1 shows the timeline of EHR development, while architecture of a modern EHR is given in Figure 2.

Sensors are ubiquitous in today's world, and are used in lifestyle, healthcare, fitness, manufacturing, and daily life [9]. Sensors are the most commonly used monitoring technology, offering the collection of data from

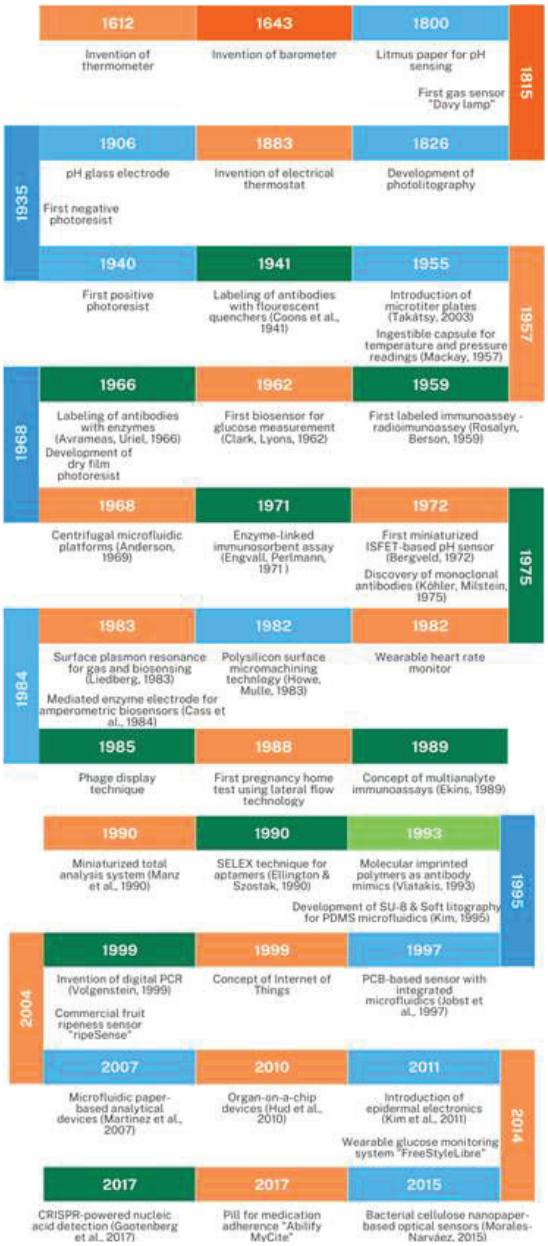

Figure 3 Historical timeline of the development of various sensors with respect to materials (blue), sensor technologies (orange) and biotechnology (green).

the environment within short time period, and often connect to the cloud through the use of various communications and transport modes, e.g., mobile networks, satellite networks, Bluetooth, broad-based networks, low-energy wide-band networks, etc. Historical timeline of the discovery of various sensors and their development with respect to materials (blue), sensor technologies (orange), and biotechnology (green) is given in Figure 3. Discovery and improvment of various (bio)materials and sensing technologies has played an essential part in developing new sensors. The development of new functional materials frequently needs to be paired with advances in other fields to produce completely novel classes of sensors. Materials play a fundamental role in the development of advanced disposable sensing devices [10], for cost reduction, environmental impact and improvment of performance and usability. Today, modern wearable fitness trackers contain various sensors and keep track of various health-related parameters. Table 1 lists sensors which can be commonly found in modern wearable activity trackers.

The data collected via sensors is then processed using manufacturer's proprietary algorithms in order to generate more detailed information, e.g., the data collected by the 3-axis accelerometer is used to calculate how many steps the user has taken, what was their speed, and at what pace, as well as the calculation of how many calories were likely burned. Most common measured metrics are given in Table 2.

However, medical devices and related services need to consistently fulfill the regulatory requirements specified in standard ISO 13485:2016 [11]. This encompasses design and development, manufacture, storage, as well as

Table 1 Most common sensors in wearable activity trackers

3-axis accelerometer	Senses motion and movement on three axes, using measurements of velocity and position. It senses inclination, tilt and orientation of the body. Ubiquitous in fitness trackers.
Gyroscope	Measures orientation and rotation, used for navigation and measuring angular velocity. 3-axis gyroscope combined with 3-axis accelerometer provides "6 degree of freedom" motion tracking system that's used in majority of fitness trackers as it is useful when tracing workout motions.
Temperature sensor	Senses temperature. Combined with motion readings, measures physical activity.
Bioimpedance sensor	Measures galvanic skin response (resistance to small electric current). Used to interpret activity and collect heartrate data.
Optical sensor	Preferred way to measuring heart rate using light.
Altimeter	Measures altitude by using pressure sensing.

Table 2 Most common metrics measured or calculated by wearable activity trackers

Steps taken	3-axis accelerometer
Distance covered	3-axis accelerometer and gyroscope
Floors climbed	Altimeter. This metric is also used for calculating calories expenditure and workout.
Heart rate	Optical sensor uses light and reflection to check the speed of blood flow on the wrist.
Body temperature	Measures temperature, also used to calculate physical activity and menstrual cycle as well as detect health issues (e.g., fever).
Oxygen saturation (SpO2)	Deoxygenated blood in veins is of a darker red color than the oxygen-filled blood in the arteries. Sensor measures relative reflection of red and infrared light. SpO2 value is estimated taking into account heartbeat rate as well.
Exercise time and calories burned	Both are calculated taking into account steps taken, distance covered, movement, velocity, and altitude, as well as heart rate, and body temperature.
Sleep duration and sleep quality	Estimated by monitoring body movements, changes in heartbeat rate, body temperature, and oxygen saturation.

distribution, installation, and maintenance of the device. Standards regarding diagnostic equipment, including medical monitoring equipment, medical thermometers and related materials are under ICS code 11.040.55, such as ISO 80601-2-56:2017 [12] for clinical thermometers for measurement of body temperature. Testing, sampling, and calibration are generally covered by the standard ISO 17025:2017 [13]. The data must be accurate, precise, and error-free in order to be used in a formal medical practice. In this context, it is necessary to revise how to guarantee the quality of data collected by fitness trackers and ensure the data is in compliance to relevant medical standards and regulations.

3 Standardization and Specification of Semantic Data Constraints

The importance of standardization lies in its capacity to enable smooth compatibility and interaction between EHR systems and other healthcare platforms. Two notable standards applied in EHR systems are Integrating the Healthcare Enterprise (IHE) and the previously mentioned HL7. IHE sets up an all-encompassing structure of technical standards to integrate healthcare systems, while HL7 defines a series of communication standards for sharing healthcare data. The Croatian national Health Information Service (HIS),

referred to as eKarton [14], is built upon the bedrock of HL7 standards. These standards are also employed in various other nations including Finland, Norway, Sweden, Iceland, and India. Within the HIS, the handling of health information exchange between databases and electronic health records (EHRs) is overseen by the Messages management system. These messages, often referred to as transactions, are precisely delineated in the IHE Technical Frameworks (IHE TF) [15]. IHE TF outlines the implementation of pre-established standards to ensure reliable medical data exchange, facilitate practical and efficient system integration, and enhance patient care quality. The HL7 standard forms the basis for file sharing. A collection of documents along with associated metadata are exchanged through the ITI-41 transaction, known as Provide and Register Document Set-b. Incoming requests are transformed through unmarshalling, which converts the incoming data stream into an HL7 Resource object (for write operations) or message header parameters (for search operations). In case of unmarshalling failure, an exception is triggered. Depending on the actors and workflows involved in the transaction, the documents and metadata can be managed, processed, and stored for future retrieval. These documents and related metadata are dispatched from a content sender to a content receiver. These occurrences can be initiated through human decisions or automated actions performed by applications aiming to submit documents to a Content Receiver, like an HIS repository. For retrieving a document from an HIS repository, the ITI-43 transaction, Retrieve Document Set, is employed. This process is illustrated in Figure 4. The IHE Profile utilizes HTTP, Web Services, IT presentation formats, and the HL7 Clinical Documentation Architecture (CDA). This facilitates the handling of HL7 Resources within the EHR system. The xds-iti43 component provides interfaces for entities sending and receiving ITI-43 messages. The endpoint URI format for the ITI-43 component is defined as:

```
xds-iti-43://hostname:port/service-path[?params]
```

Here, hostname refers to the domain name or IP address, service represents the service path, and params are optional parameters. An instance of the exposed FHIR REST Service endpoint would consequently be, for example:

```
http://ekarton-server.org:8888/IHE/xds/iti43
```

Existing EHR implementations already utilize ITI messages in conjunction with HL7 standards, featuring predefined and implemented service endpoints for various ITI messages.

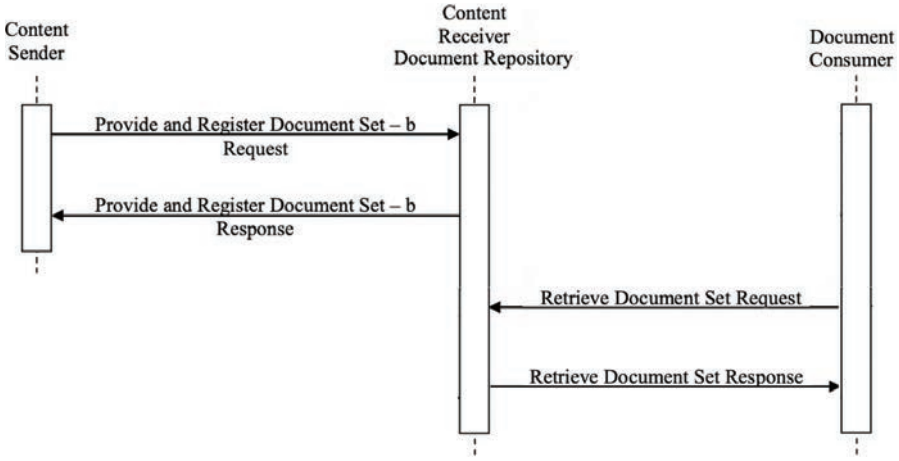

Figure 4 ITI-41 and ITI-42 messages.

Previous research [16] and [17] thoroughly investigate the essential prerequisites and potential remedies for seamlessly incorporating external data into EHR systems, as they have evolved into a fundamental element of the core health information infrastructure in a significant number of European countries.

FHIR stands as a protocol for healthcare data exchange, with its publication by HL7 and application in the electronic transfer of medical information. FHIR serves as a critical bridge in the modern healthcare landscape, enabling the secure and efficient sharing of patient-related data between different systems, applications, and healthcare providers. Its fundamental element is a Resource, as illustrated in Figure 5, encompassing metadata, standardized data, and a human-readable section. In the realm of wearable sensor technology, the proliferation of devices has led to an explosion in data generation. On a daily basis, a single device can produce an overwhelming excess of 4 million data points through its sensors. This constant stream of data translates into substantial data files that require careful processing to extract meaningful insights. To achieve this, the raw health data collected from wearable devices is first consolidated and harmonized. It then undergoes a comprehensive processing phase where it is cleansed, organized, and transformed into a format suitable for further analysis. This processed data is then mapped into the HL7 FHIR standard, effectively translating the wealth of sensor data into a structured framework that is

Figure 5 HL7 FHIR Resource [18].

consistent with modern healthcare practices. By aligning the processed data with the HL7 FHIR format, healthcare professionals gain the ability to seamlessly integrate this information into a patient's EHR. In essence, FHIR provides the structure and standardization necessary to bridge the gap between wearable device-generated data and the formal Electronic Health Record.

A majority of commercially available wearable sensor trackers, such as Fitbit [19], typically store the data they gather in the JSON (JavaScript Object Notation) format. This format is widely recognized for its flexibility and ease of use, making it a preferred choice for encapsulating various types of information. However, when dealing with the significant amounts of data

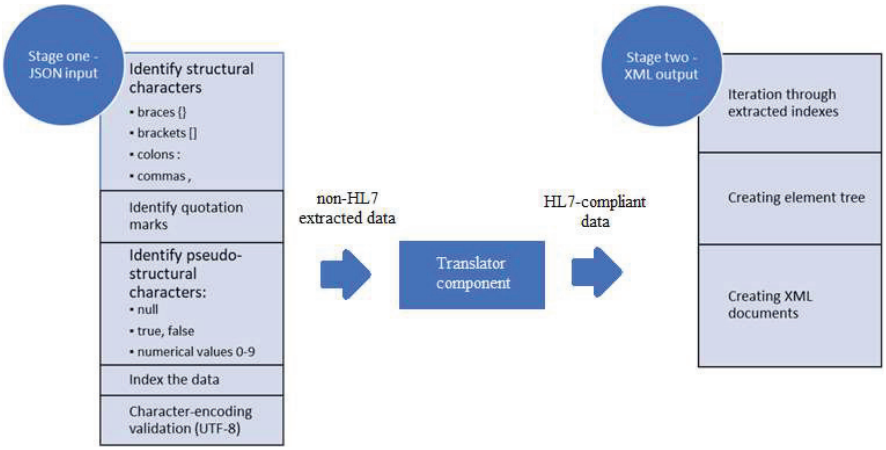

Figure 6 Process of data parsing.

generated by these devices, efficient handling becomes paramount. The parser responsible for extracting and organizing this data must be robust and capable of managing substantial data volumes without crashing or compromising performance. Effective support for large volumes of data is crucial for the parser's functionality, ensuring both reliability and stability. To address this, a JSON parser (depicted in Figure 6) was devised and implemented in Python using the pysimdjson module, with acceleration through SIMD (single instruction, multiple data) technology, enabling data processing speeds of up to 2.2 GB/s. This high processing efficiency is crucial for wearable sensor trackers, which can generate a substantial amount of data points over time. The parser operates in two main stages, with a translator component in between: the initial stage handles input data in 64-byte batches, while the subsequent stage constructs a "tape representation" post-translation. This two-stage parsing structure was tested using the OxyBeat dataset [20], containing heart rate, body temperature, and oxygen saturation (SpO2) data collected over two months from a Fitbit Versa 3 device, resulting in over 2 million datapoints. The development of this accelerated JSON parser addresses the crucial need for efficient data processing in wearable sensor trackers. By enabling rapid and reliable extraction of information from the JSON format, this parser facilitates the seamless integration of data into various applications, such as electronic health records, enabling improved healthcare monitoring and analysis.

Pseudocode for data parsing process, including the algorithm used to extract and convert data within the Translator component is also given as follows:

```
1. Identify structural characters (braces, brackets, colons, commas)
2. Identify pseudostructural characters (null, true, false, numerical value)
3. Index the data
Let data be an empty array
For each character in JSON_data:
    If the character is a brace or bracket, push it onto the data array
    If the character is a colon or comma, push it onto the data array
    If the character is a quotation_mark, push the entire string within the quotes onto the data array
    If the character is a pseudostructural character or numerical value, push it onto the data array
4. Validate UTF-8 encoding
5. Translate data into HL7 compliant format
Extract non HL7 data and convert to HL7-compliant data
        Input: list of indexed data lines from JSON (L) that need to be translated
        Input: name ID to verify type of Resource (R)
        Output: data mapped to correct HL7 Resource element (M)
Use name ID to identify Resource type (R) using keywords from Resource IDs (K)
var: current line (l) to convert
loop for each l ϵ L:
        var: list of elements (E)
        split(s) into E
        tokenize(s) into E
        var e ϵ E is current element
        loop for each e ϵ E:
                var current map element m(K,e)
                correctFormat(e)
                tag(m(K,e))
                add to output map M(k,e)
        done
done
6. Creating element tree
Let root be a new XML element called "health_data"
For each item in HL7_data:
    Let element be a new XML element with the name of the HL7 element and the value of the HL7 element
    Append the element to the root element
7. Create XML document
    Let xml_doc be a new XML document
    Append the root element to the xml_doc
    Return xml_doc as a string of XML-formatted data
```

The algorithm's input parameters consist of the indexed data lines retrieved from the JSON parser (L), as well as an identifying name ID to designate the specific Resource (R). The desired outcome is a data map encompassing elements of a specific HL7 Resource. Each data line is fragmented into separate tokens. For every token, the corresponding element within the Resource is located (K). These elements are then incorporated into an output map, where the values (e) are associated with the specific elements (K) of the designated Resource (R). Subsequently, this assembled information is fed into an XML parser. Consequently, the accumulation of readings spanning two months for each of the three scrutinized data types results in JSON files exceeding 50MB, containing a dataset comprising more than 2 million datapoints. This data was efficiently processed by a Ryzen 5 5600X six-core processor, operating at a clock speed of 3.7 GHz in 64-bit mode, completing the analysis within a 58-second timeframe. This thorough evaluation encompassed all data points, and it achieved a reliability rate of 100 percent.

4 Validation Process

In its initial stages, the exclusive approach to validating XML documents relied on schema validation. This validation process established the validity of an XML document solely if it adhered strictly to the specified schema. While schema validation ensured the structural accuracy of the document, it fell short in verifying conditional and integrity-related criteria. To effectively validate a personal healthcare record, the recommended method involves a two-step process: initially validating the document's structure, followed by assessing the content and inherent characteristics. Additionally, any supplementary limitations present must be subjected to validation. Schematron, an advanced structural schema validation language, operates on a foundation of rules expressed in Extensible Markup Language (XML). It stands out by enabling assertions regarding the presence or absence of specific patterns within XML trees. Unlike other XML schema languages like XML Schema and Document Type Definition (DTD), Schematron is uniquely capable of imposing restrictions in a manner that goes beyond what these other languages can achieve (as depicted in Figure 7).

The foundational structure of ISO Schematron is constructed as a series of four sequential stages within an Extensible Stylesheet Language Transformations (XSLT) pipeline [21]. XSLT functions as a declarative language specifically designed for the transformation of XML documents into various other formats, such as plain text, HTML, or another XML format. Forming an integral part of the Extensible Stylesheet Language (XSL) family, XSLT is frequently employed alongside Extensible Stylesheet Language Formatting

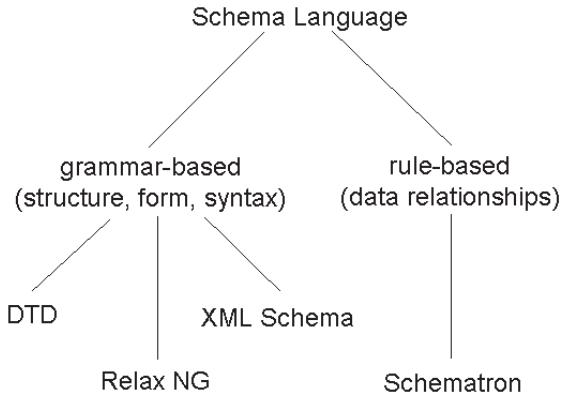

Figure 7 Classification of schema languages.

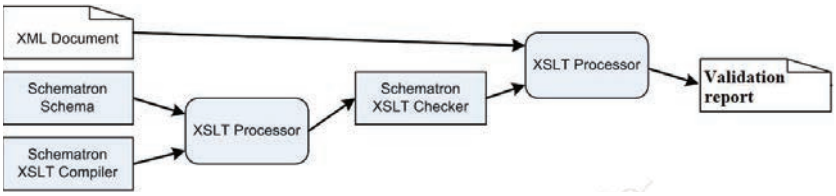

Figure 8 Process of Schematron validation.

Objects (XSL-FO) to generate printable documents. Employing XML-based templates and rules, XSLT delineates how to convert an input XML document into the desired output format. The mechanism of XSLT involves the application of templates to nodes within the input XML document, aligning with specified patterns within the XSLT stylesheet. Each template incorporates instructions on how to convert the matched node and its associated children into the targeted output format. The initial two stages serve as macro-processors, offering value primarily when intricate features are in use. In summary, the process of Schematron validation involves the creation of a Schematron schema, subsequent compilation of this schema into an XSLT stylesheet, application of the stylesheet to the XML document utilizing an XSLT processor, processing the intermediate result with an XSLT checker, and eventually producing a comprehensive validation report that highlights any encountered errors or warnings. This multifaceted process is illustrated in Figure 8. To ensure the smooth transfer and integration of personal health data, it's crucial to recognize that distinct XML syntaxes are employed by different devices to express similar health-related information, such as heart rate measurements. For effective data interchange, XML documents must satisfy the criteria of constituting a comprehensive FHIR resource while maintaining acceptable syntax. To streamline this process, the objective is to create and maintain a single Schematron document capable of accommodating analogous health data originating from diverse personal tracking devices.

Schematron proves instrumental in establishing relationships or constraints where the existence of one element depends on another, as well as in enforcing the presence of attributes within specific elements. The power of Schematron lies in its ability to incorporate intricate rules and restrictions necessary for semantic validation. Schematron rules are articulated using the rule element, which includes a context property that employs an XPath Expression to specify one or more nodes within the document. This context

```
<rule context="Observation">
    <assert test="@id">The element Observation must have an id attribute.</assert>
<assert test="count(*) = 2 and count(Category) = 1 and count(Code)= 1">
    The element Observation must have the child elements Category and Code.</assert>
</rule>
```

Figure 9 Rule instance for FHIR resource observation.

```
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron" queryBinding="xslt2">
  <sch:ns prefix="f" uri="http://hl7.org/fhir"/>
  <sch:ns prefix="h" uri="http://www.w3.org/1999/xhtml"/>
  <sch:pattern id="observation">
    <sch:title>Observation</sch:title>
    <sch:rule context="f:Observation">
      <sch:assert test="not(exists(f:dataAbsentReason)) or (not(exists(*[starts-with(local-name(.), 'value')])))">dataAbsentReason
        SHALL only be present if Observation.value[x] is not present (inherited)</sch:assert>
      <sch:assert test="not(exists(f:component/f:code)) or
        count(for $coding in f:code/f:coding return parent::*/f:component/f:code/f:coding[f:code/@value=$coding/f:code/@value
        and f:system/@value=$coding/f:system/@value])=0">Component code SHALL not be same as observation code (inherited)</sch:assert>
      <sch:assert test="f:id/@value = 'heart-rate'">Observation is not heartrate type observation</sch:assert>
      <sch:assert test="exists(f:subject/f:reference/@value)">Patient must exist and be uniquely defined</sch:assert>
      <sch:assert test="matches(f:effectiveDateTime/@value, '^(\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}\+\d{2}:\d{2})$')">Observation
        needs to have proper format for dateTime</sch:assert>
      <sch:assert test="exists(f:valueQuantity/f:value/@value)">Observation needs to have value measured</sch:assert>
    </sch:rule>
  </sch:pattern>
  <sch:pattern id="category">
    <sch:rule context="f:Observation/f:category/f:coding/f:code">
      <sch:assert test="@value = 'vital-signs'">Vital signs must defined by correct observation code</sch:assert>
      <sch:assert test="count(@value) = 1">Code must exist and be uniquely defined</sch:assert>
    </sch:rule>
    <sch:rule context="f:Observation/f:category/f:coding/f:system">
      <sch:assert test="@value = 'http://terminology.hl7.org/CodeSystem/observation-category'">Vital signs must
        defined by correct system</sch:assert>
    </sch:rule>
  </sch:pattern>
  <sch:pattern id="code">
    <sch:rule context="f:Observation/f:code/f:coding">
      <sch:assert test="f:code/@value = '8867-4'">Heartbeat rate must defined by correct observation code</sch:assert>
      <sch:assert test="f:system/@value = 'http://loinc.org'">Heartbeat rate must defined by correct system</sch:assert>
      <sch:assert test="count(f:code/@value) = 1">Code must exist and be uniquely defined</sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>
```

Figure 10 Heartbeat rate data validation.

attribute delineates the scope where the assertion applies. In the illustrated example, the context is set to the Observation element, signifying that the Schematron rule pertains to the Observation element (as illustrated in Figure 9). Within the Schematron schema, the assertion element comes into play, defining data constraints to be assessed within the specified context of the XML document. This schema encapsulates a dynamic toolset to facilitate the standardization and validation of health data, ensuring interoperability across a diverse range of personal tracking devices. The FHIR HeartRate Structure Definition furnishes an outline of data pertaining to heart rate measurements. The Schematron rules for validating heart rate data are illustrated in Figure 10. For effective validation of heart rate information, the Schematron framework must be equipped with the designated rules to meet all the previously mentioned criteria and to align with established standards.

Line-by-line explanation of the schematron above is the following:

1. This is the opening tag of the Schematron schema. It declares the start of the schema definition, specifies the XML namespace for Schematron and specifies that the query language used is XSLT 2.0.
2. Defines a namespace with prefix "f" and URI "http://hl7.org/fhir". This namespace will be used in XPath expressions throughout the schema. This namespace is used throughout the schema to specify elements and attributes from the FHIR data format.
3. Defines a namespace with prefix "h" and URI "http://www.w3.org/1999/xhtml". This namespace will also be used in XPath expressions throughout the schema. This namespace is used to specify elements and attributes from the XHTML data format.
4. Line starts a new pattern with ID "observation", which contains rules for validating FHIR Observation resources.
5. Specifies the title of the pattern as "Observation". Provides a human-readable title for the pattern.
6. Starts a new rule within the pattern, which applies to all elements in the document with the FHIR Observation resource type.
7-8. This line defines an assertion (a condition that must be true for the data to be considered valid). It checks that either the dataAbsentReason element is not present, or if it is present, then none of the value elements in the Observation element start with the word "value".
9-11. This line defines another assertion. It checks that either the code element in the component element is not the same as the code element in the Observation element, or if they are the same, then the system attribute in the code element of the component element is not the same as the system attribute in the code element of the Observation element.
12. Defines an assertion that checks that the "id" attribute of the "Observation" element is equal to "heart-rate".
13. Defines an assertion that checks that the "subject" element is present and has a "reference" attribute with a value.
14-15. Defines an assertion that checks that the "effectiveDateTime" element has a value in the correct format.
16. This asserts that an Observation resource must have a measured value, which is represented by the valueQuantity element with a nested value attribute.
17. End of rule definition for Observation resource.
18. End of pattern definition for Observation resource.
19. Declares a pattern with an id of "category". This pattern will contain rules for validating the Observation category code.
20. Declares a rule that applies to the category coding code element within the Observation resource.
21. Declares an assertion that the value of the code element must be "vital-signs".
22. Declares an assertion that the code element must exist and be uniquely defined.
23. End of rule definition for category coding code.
24. Declares a rule that applies to the category coding system element within the Observation resource.
25-26. Declares an assertion that the value of the system element must be http://terminology.hl7.org/CodeSystem/observation-category, i.e., that Vital signs must be defined by correct system.
27. End of rule definition for category coding system.
28. End of pattern definition for category coding.
29. Declares a pattern with an id of "code". This pattern will contain rules for validating the Observation code.
30. Declares a rule that applies to the code element within the Observation resource.
31. Declares an assertion that the value of the code element must be "8867-4", i.e., Heartbeat rate must be defined by correct observation code.
32. Declares an assertion that the system element must have a value of http://loinc.org, i.e., Heartbeat rate must be defined by correct system.
33. Declares an assertion that the code element must exist and be uniquely defined.
34.-35. End of rule/pattern definition for code.
36. End of schema.

The dataset employed includes two additional health-related data types, body temperature (Figure 11) and oxygen saturation (Figure 12), which underwent identical processes of constraint specification, verification, and validation.

Below is the data after the transformation process, adhering to all the rules in the abovementioned schematron for its particular data type (heartbeat rate).

The procedure of validating XML data against a Schematron leads to the generation of a report indicating any instances of rule violations, if present. Numerous libraries exist for XML validation with Schematron across different programming languages, Java included. Among the widely used

```
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron" queryBinding="xslt2">
  <sch:ns prefix="f" uri="http://hl7.org/fhir"/>
  <sch:ns prefix="h" uri="http://www.w3.org/1999/xhtml"/>
  <sch:pattern id="observation">
    <sch:title>Observation</sch:title>
    <sch:rule context="f:Observation">
      <sch:assert test="not(exists(f:dataAbsentReason)) or (not(exists(*[starts-with(local-name(.), 'value')])))">dataAbsentReason
        SHALL only be present if Observation.value[x] is not present (inherited)</sch:assert>
      <sch:assert test="not(exists(f:component/f:code)) or
        count(for $coding in f:code/f:coding return parent::*/f:component/f:code/f:coding[f:code/@value=$coding/f:code/@value
        and f:system/@value=$coding/f:system/@value])=0">Component code SHALL not be same as observation code (inherited)</sch:assert>
      <sch:assert test="f:id/@value = 'body-temperature'">Observation is not body temperature type observation</sch:assert>
      <sch:assert test="exists(f:subject/f:reference/@value)">Patient must exist and be uniquely defined</sch:assert>
      <sch:assert test="matches(f:effectiveDateTime/@value, '^(\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}\+\d{2}:\d{2})$')">Observation
        needs to have proper format for dateTime</sch:assert>
      <sch:assert test="exists(f:valueQuantity/f:value/@value)">Observation needs to have value measured</sch:assert>
    </sch:rule>
  </sch:pattern>
  <sch:pattern id="category">
    <sch:rule context="f:Observation/f:category/f:coding/f:code">
      <sch:assert test="@value = 'vital-signs'">Vital signs must defined by correct observation code</sch:assert>
      <sch:assert test="count(@value) = 1">Code must exist and be uniquely defined</sch:assert>
    </sch:rule>
    <sch:rule context="f:Observation/f:category/f:coding/f:system">
      <sch:assert test="@value = 'http://terminology.hl7.org/CodeSystem/observation-category'">Vital signs must
        defined by correct system</sch:assert>
    </sch:rule>
  </sch:pattern>
  <sch:pattern id="code">
    <sch:rule context="f:Observation/f:code/f:coding">
      <sch:assert test="f:code/@value = '8310-5'">Body temperature must defined by correct observation code</sch:assert>
      <sch:assert test="f:system/@value = 'http://loinc.org'">Body temperature must defined by correct system</sch:assert>
      <sch:assert test="count(f:code/@value) = 1">Code must exist and be uniquely defined</sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>
```

Figure 11 Body tamperature Schematron for data validation.

```
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron" queryBinding="xslt2">
  <sch:ns prefix="f" uri="http://hl7.org/fhir"/>
  <sch:ns prefix="h" uri="http://www.w3.org/1999/xhtml"/>
  <sch:pattern id="observation">
    <sch:title>Observation</sch:title>
    <sch:rule context="f:Observation">
      <sch:assert test="not(exists(f:dataAbsentReason)) or (not(exists(*[starts-with(local-name(.), 'value')])))">dataAbsentReason
        SHALL only be present if Observation.value[x] is not present (inherited)</sch:assert>
      <sch:assert test="not(exists(f:component/f:code)) or
        count(for $coding in f:code/f:coding return parent::*/f:component/f:code/f:coding[f:code/@value=$coding/f:code/@value
        and f:system/@value=$coding/f:system/@value])=0">Component code SHALL not be same as observation code (inherited)</sch:assert>
      <sch:assert test="f:id/@value = 'sat02'">Observation is not oxygen saturation type observation</sch:assert>
      <sch:assert test="exists(f:subject/f:reference/@value)">Patient must exist and be uniquely defined</sch:assert>
      <sch:assert test="matches(f:effectiveDateTime/@value, '^(\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}\+\d{2}:\d{2})$')">Observation
        needs to have proper format for dateTime</sch:assert>
      <sch:assert test="exists(f:valueQuantity/f:value/@value)">Observation needs to have value measured</sch:assert>
    </sch:rule>
  </sch:pattern>
  <sch:pattern id="category">
    <sch:rule context="f:Observation/f:category/f:coding/f:code">
      <sch:assert test="@value = 'vital-signs'">Vital signs must defined by correct observation code</sch:assert>
      <sch:assert test="count(@value) = 1">Code must exist and be uniquely defined</sch:assert>
    </sch:rule>
    <sch:rule context="f:Observation/f:category/f:coding/f:system">
      <sch:assert test="@value = 'http://terminology.hl7.org/CodeSystem/observation-category'">Vital signs must
        defined by correct system</sch:assert>
    </sch:rule>
  </sch:pattern>
  <sch:pattern id="code">
    <sch:rule context="f:Observation/f:code/f:coding">
      <sch:assert test="f:code/@value = '2708-6'">Oxygen saturation must defined by correct observation code</sch:assert>
      <sch:assert test="f:system/@value = 'http://loinc.org'">Oxygen saturation must defined by correct system</sch:assert>
      <sch:assert test="count(f:code/@value) = 1">Code must exist and be uniquely defined</sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>
```

Figure 12 Oxygen saturation Schematron for data validation.

```
<?xml version="1.0" encoding="UTF-8"?>
<Observation xmlns="http://hl7.org/fhir">
  <id value="heart-rate"/>
  <meta>
    <profile value="http://hl7.org/fhir/StructureDefinition/vitalsigns"/>
  </meta>
  <text> <status value="generated"/> <div xmlns="http://www.w3.org/1999/xhtml">Generated Narrative:
     Observation "heart-rate" </text>
  <category>
    <coding>
      <system value="http://terminology.hl7.org/CodeSystem/observation-category"/>
      <code value="vital-signs"/>
      <display value="Vital Signs"/>
    </coding>
    <text value="Vital Signs"/>
  </category>
  <code>
    <coding>
      <system value="http://loinc.org"/>
      <code value="8867-4"/>
      <display value="Heart rate"/>
    </coding>
    <text value="Heart rate"/>
  </code>
  <subject>
    <reference value="Marija Horvat"/>
  </subject>
  <effectiveDateTime value="2023-03-02"/>
  <valueQuantity>
    <value value="44"/>
    <unit value="beats/minute"/>
    <system value="http://unitsofmeasure.org"/>
    <code value="/min"/>
  </valueQuantity>
</Observation>
```

Figure 13 HL7 compliant heartbeat rate data.

Schematron validation libraries for Java, the "Saxon" library stands out, offering comprehensive support not only for Schematron validation but also for XSLT and XQuery processing. Saxon is an open-source library suitable for deployment in both commercial and non-commercial software applications. For performing Schematron validation using Saxon in a Java context, the following approach was adopted:

```
// Load the XML and Schematron files
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document xml = builder.parse(new File("sample.xml"));
Source schematron = new StreamSource(new File("schematron.sch"));

// Create a Schematron validator
Processor processor = new Processor(false);
XsltCompiler compiler = processor.newXsltCompiler();
XsltExecutable exec = compiler.compile(schematron);
XsltTransformer transformer = exec.load();

// Validate the XML
transformer.setSource(new DOMSource(xml));
transformer.setDestination(new NullDestination());
transformer.transform();
```

This piece of code utilizes the Saxon library to load both the XML and Schematron files. It then proceeds to establish a Schematron validator, employing it to validate the XML against the designated Schematron rules. Comparable libraries for Schematron validation are also available in other programming languages. Examples include "libxml2" for C/C++, "lxml" for Python, and "Xerces" for both Java and C++. The selection of the appropriate library and corresponding code hinges on the programming language and environment of the Health Information System (HIS) in question, in this scenario, Java.

When the message contains data as displayed in the aforementioned Figure 11, the resulting report confirms the data's validity. Conversely, if the data, as illustrated in Figure 14, fails to adhere to all stipulated rules, the outcome is depicted in Figure 15.

Upon inspection, it is clear that the observation code does not match the necessary heartbeat rate LOINC code. Furthermore, the patient is not defined.

```
<Observation xmlns="http://hl7.org/fhir">
  <id value="heart-rate"/>
  <category>
    <coding>
      <system value="http://terminology.hl7.org/CodeSystem/observation-category"/>
      <code value="vital-signs"/>
    </coding>
  </category>
  <code>
    <coding>
      <system value="http://loinc.org"/>
      <code value="8867-5"/>
    </coding>
  </code>
  <subject></subject>
  <effectiveDateTime value="07-02"/>
  <valueQuantity>
    <value value="44"/>
    <unit value="beats/minute"/>
    <system value="http://unitsofmeasure.org"/>
    <code value="/min"/>
  </valueQuantity>
</Observation>
```

Figure 14 Non-compliant heartbeat rate data.

Errors

Severity	Location	Filename	Message
⚠	Line 12	sample.xml	Pattern 'code' Failed : Heartbeat rate must defined by correct observation code.
⚠	Line 15	sample.xml	Pattern 'Observation' Failed : Patient must exist and be uniquely defined.
⚠	Line 16	sample.xml	Pattern 'Observation' Failed : Observation needs to have proper format for dateTime.

Figure 15 Negative report of Schematron validation or heartbeat rate.

Finally, the datetime field doesn't have proper format. Thus, the received data is not a proper FHIR resource and does not comply with the specification and HL7 standard.

5 Conclusion

The potential for enhancing personalized healthcare through the utilization of data collected via personal wireless trackers is substantial. One key factor contributing to this potential is the widespread availability of these trackers, such as smartwatches and fitness bands. These devices have gained popularity due to their affordability and practicality, leading to their increasing prevalence. Moreover, these trackers continuously gather data, offering healthcare professionals more comprehensive and detailed insights into individuals' daily conditions. This wealth of information empowers healthcare practitioners to make better-informed decisions. Additionally, the advancement of 5G and 6G network technologies further enhances communication between Internet of Medical Things (IoMT) devices and cloud platforms. This advancement translates to improved performance and Quality of Service (QoS). Nevertheless, the integration of personal health data into formal medical information systems comes with challenges. These challenges necessitate important steps such as syntax verification and semantic validation of medical data. Adhering to standards and regulations, along with ensuring proper data structure definition (DSD), becomes crucial for seamless communication. Successfully addressing these challenges would enable Electronic Health Records (EHRs) to accommodate processed sensor data that aligns with established standards and holds clinical relevance. The outcome would be a standardized approach to personalizing medical services, representing a notable stride towards ensuring accessible and uninterrupted care.

This research introduces a comprehensive approach that involves specifying semantic data constraints and validating the information obtained from wearable smart devices. This validation process is executed through a Schematron that aligns with internationally recognized Electronic Health Record (EHR) standards and regulations as defined by Integrating the Healthcare Enterprise (IHE) and Health Level 7 (HL7). A pivotal element in contemporary healthcare data exchange is Fast Healthcare Interoperability Resources (FHIR), which serves as a crucial link between data generated by wearable devices and the structured, standardized format required by formal Electronic Health Records. The integration of FHIR signifies a significant advancement in healthcare technology, as it establishes a bridge between

the data generated by wearable devices and the formal Electronic Health Record system. This bridging process holds immense potential, not only for enhancing patient care but also for driving forward medical progress through the utilization of data-driven insights. It marks the advent of a new era in healthcare, one that is fueled by the valuable information gathered from wearable devices. To ensure adherence to the set standards and regulations, semantic constraints were meticulously defined for healthcare data types. This involved creating a framework that outlines the permissible limits and characteristics of the collected data. Additionally, a well-structured validation process was meticulously developed and modeled. This process ensures the seamless transfer and integration of the acquired data into an official Electronic Health Record (EHR) system, maintaining data accuracy, consistency, and compliance throughout. The result is a robust foundation for leveraging wearable technology's potential to revolutionize patient care and accelerate medical advancements.

References

[1] Global Industry Analysts, Inc. (June 2020). Fitness Bands – Global Market Trajectory & Analytics.

[2] Mordor Intelligence. (January 2020). Smart Watch Market – Growth, Trends, and Forecast (2020 – 2025).

[3] Grand View Research, Inc. (2019). Internet of Things in Healthcare Market Size, Share & Trends Analysis Report by Component, by Connectivity Technology, by End Use, by Application, And Segment Forecasts, 2019 – 2025.

[4] Markets and markets. (2020). IoT in Healthcare Market by Component (Medical Device, Systems & Software, Services, and Connectivity Technology), Application (Telemedicine, Connected Imaging, and Inpatient Monitoring), End User, and Region – Global Forecast to 2025.

[5] Weed LL. Medical records that guideand teach. N Engl J Med. 1968 Mar 14;278(11):593–600.

[6] McDonald CJ, Tierney WM. The Medical Gopher – a microcomputer system to help find, organize and decide about patient data. West J Med 1986; 145(6): 823–9.

[7] Evans, R. S. (2016). Electronic Health Records: Then, Now, and in the Future. Intermountain Healthcare & Biomedical Informatics, University of Utah School of Medicine, Salt Lake City, USA.

[8] Harrington, J. (1991). Application of open systems to health care communications. IEEE Medical Data Interchange (MEDIX).
[9] Javaid, M., Haleem, A., Rab, S., Singh, R. P., and Suman, R. (2021). Sensors for daily life: A review. Sensors International, Volume 2.
[10] Dincer, Can, B., Richard, R., Estefanía, F.-A., M. Teresa, M., Arben, M., Güder, F. G. (2019). Disposable Sensors in Diagnostics, Food, and Environmental Monitoring. Advanced Materials, Volume 31.
[11] Medical devices – Quality management systems – Requirements for regulatory purposes, ISO 13485:2016, Geneva, Switzerland: ISO.
[12] Medical electrical equipment – Part 2-56: Particular requirements for basic safety and essential performance of clinical thermometers for body temperature measurement, ISO 80601-2-56:2017, Geneva, Switzerland: ISO.
[13] General requirements for the competence of testing and calibration laboratories, ISO/IEC 17025:2017, Geneva, Switzerland: ISO.
[14] Cezih eKarton, http://www.cezih.hr/eKarton.html, accessed 5.7.2023.
[15] IHE Resources, Technical Frameworks. https://www.ihe.net/resources/technical_frameworks/#IT, accessed 5.7.2023.
[16] A. Koren, M. Jurčević and R. Prasad, "Semantic Constraints Specification and Schematron-Based Validation for Internet of Medical Things' Data," in IEEE Access, vol. 10, pp. 65658–65670, 2022, doi: 10.1109/ACCESS.2022.3182486.
[17] A. Koren and R. Prasad, "Standardization of Third-party Data in Electronic Health Records," 2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC), Herning, Denmark, 2022, pp. 453–458, doi: 10.1109/WPMC55625.2022.10014929.
[18] HL7 FHIR, https://fhir-ru.github.io/summary.html, accessed 5.7.2023.
[19] Fitbit Research Library. Fitabase, https://www.fitabase.com/research-library/, accessed 5.7.2023.
[20] Koren, A. (2021). OxyBeat Dataset. https://github.com/korenana/oxybeat-dataset, accessed 7.12.2022.
[21] Siegel E., Schematron: A Language for Validating XML, XML Press, November 2022.

Biographies

Ana Koren completed her PhD in October 2023 at the Faculty of Electrical Engineering and Computing, University of Zagreb. She was a visiting researcher at TU Graz (Austria), Universidad de Zaragoza (Spain) and Universidad Nacional de Colombia (in Bogota, Colombia). Main areas of interest include e-Health and wireless personal communications. She worked on implementing Croatia's Central Health Information System, including the Electronic Health Record (EHR).

Ramjee Prasad, Fellow IEEE, IET, IETE, and WWRF, is a Professor of Future Technologies for Business Ecosystem Innovation (FT4BI) in the Department of Business Development and Technology, Aarhus University, Herning, Denmark. He is the Founder President of the CTIF Global Capsule (CGC). He is also the Founder Chairman of the Global ICT Standardization Forum for India, established in 2009. He has been honored by the University of Rome "Tor Vergata", Italy as a Distinguished Professor of the Department of Clinical Sciences and Translational Medicine on March 15, 2016. He is an Honorary Professor of the University of Cape Town, South Africa, and the University of KwaZulu-Natal, South Africa. Dr. Prasad has received Ridderkorset of Dannebrogordenen (Knight of the Dannebrog) in 2010 from the Danish Queen for the internationalization of top-class telecommunication research and education. He has received several international awards such as IEEE Communications Society Wireless Communications Technical Committee Recognition Award in 2003 for making contribution in the field of "Personal, Wireless and Mobile Systems and Networks", Telenor's Research Award in 2005 for impressive merits, both academic and organizational within the field of wireless and personal communication, 2014 IEEE AESS Outstanding Organizational Leadership Award for: "Organizational Leadership in developing and globalizing the CTIF (Center for TeleInFrastruktur) Research Network", and so on. He has been the Project Coordinator of several EC projects namely, MAGNET, MAGNET Beyond, eWALL. He has published more than 50 books, 1000 plus journal and conference publications, more than 15 patents, over 140 Ph.D. Graduates and a larger number of Masters (over 250). Several of his students are today worldwide telecommunication leaders themselves.