

Using BPMN for medical guidelines that integrate with FHIR-RDF

Gerhard Kober^{1,*}, Livio Robaldo² and Adrian Paschke³

¹Visuapps GmbH, 1100, Vienna, AT

²Legal Innovation Lab Wales, Swansea University, UK

³Fraunhofer FOKUS and Freie Universitaet Berlin, Berlin, DE

Abstract

Medical guidelines are a core concept within evidence-based medicine and provide information on how patient treatment should be applied. The combination of medical guidelines, distributed Fast Healthcare Interoperability Resources (FHIR) resources describing available patient data, and related medical ontologies is substantial for a clinical decision support system to provide the most relevant information for the patient and the physician. In this work, we explore the integration of Business Process Modeling Notation (BPMN), FHIR, and Semantic Web technologies to achieve decision support for the medical use case. By using FHIR-RDF representation and SPARQL, we can provide relevant information for the business process execution engine, which takes decisions and triggers feeds of new observations to a FHIR store for creating new patient-related data. The business model describes which information is relevant for decision tasks. By extending the process engine's code, we were able to separate the high-level workflow from the very technical tasks, such as querying for data or sending notifications. The application of medical guidelines is critical for patient care while technologies are emerging, and lots of information is available at different locations. With the relatively new standard FHIR, we can use discrete data, and by storing these data in RDF, we can integrate with already existing medical ontologies that allow further inferences to process information about the patients.

Keywords

BPMN, FHIR, RDF, Medical guidelines, Clinical decision support

1. Introduction

Medical guidelines are essential for applying evidence-based medicine to patients. Guidelines are usually available as textual descriptions of medical treatment recommendations for medical personnel to ensure appropriate care for particular illnesses or injuries. These textual descriptions sometimes contain workflow diagrams as explanations for the physician who exerts these guidelines on patients, but sometimes these workflows are only described in natural language.

The well-known modeling language "Business Process Modeling Notation" (BPMN) is widely used as a graphical representation of medical guideline workflows and other processes in the healthcare domain [1].

SWAT4HCLS 2023: The 14th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences

*Corresponding author.

✉ gerhard.kober@visuapps.com (G. Kober); livio.robaldo@swansea.ac.uk (L. Robaldo); adrian.paschke@fokus.fraunhofer.de (A. Paschke)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

On the other hand, Fast Healthcare Interoperability Resources (FHIR) [2] is a relatively new standard that fosters healthcare interoperability and provides a data format to store and provide patient medical information. Furthermore, it defines the exchange protocols among different participant systems. FHIR might be encoded in RDF format and thus queried via SPARQL¹.

From a system-architectural perspective, these FHIR resources can be distributed in multiple physical locations. Therefore, collection and pre-processing of data is usually necessary.

There is a need to provide a temporary RDF triplestore over the generated full graph that grows over time since different patient monitoring devices are inserting data regularly. The short-term triplestore is an integral component because in production used FHIR stores are not capable of providing SPARQL endpoints.

The two aspects, i.e., the BPMN model, which provides an intuitive description of a medical guideline, and the FHIR-Data, which provides the basis for a future guideline execution, need to interact to create a working clinical decision support system.

Decoupling the medical guideline from the medical data allows to better organize the (joint) work of all actors involved in a healthcare process. Imagine the physician has the expertise to focus on the clinically relevant aspects, and by using a simple interface, he can define the workflow. At the same time, technical experts can resolve the challenge of medical data integration to a BPMN execution engine. Medical treatment can take advantage of the proposed model: by separating the concerns of medical content and medical data representations since each involved actor can focus on their respective tasks. This paper is not concerned with the automatic translation of medical guidelines and the validation of such cases.

In this paper, we want to show how we can separate the medical objective from the technical challenges of data acquisition and data integration involving FHIR and BPMN. The paper is structured as follows: section 2 describes existing approaches using medical guidelines; section 3 presents our approach to express medical guidelines as BPMN models and their execution. Section 4 describes our results that we discuss in section 5, where we finally conclude.

2. Related Work

Editors of medical guidelines document the current findings of medical studies and recommendations in natural language and sometimes include graphical representations of workflows. These workflows need a representation in a computer-executable form that may be used in practical applications. In [3], BPMN was used for modeling and presenting the clinical pathway in the form of a workflow. Their system used an Enterprise Service Bus to integrate an Electronic Health Record (EHR) and alerts when a certain pulmonary disease diagnosis occurred.

BPMN is an OMG standard for business process modeling in industry and economy, with the benefits of generating a graphical representation of complex processes and being precise enough to be transformed into software applications executing such processes². Medical guidelines can easily be integrated into such a process model [1]. Furthermore, BPMN diagrams can be used for representing the control flow of processes, relevant events, and the corresponding sequence flows. However, this paper only describes general best practices for using BPMN in

¹<https://www.w3.org/TR/rdf-sparql-query/>

²<https://www.omg.org/spec/BPMN/2.0/About-BPMN/>

the medical domain, leaving out the technical integration with patient medical data. It turns out that current Hospital Information Systems (HIS)³ are using workflows, but more for general purposes during a patient hospital period than for the execution of medical guidelines.

There are various existing models for a formal representation and execution of medical guidelines like Asbru [4], EON[5], GLIF3[6] and ProForma[7]. However, these models are very specialized for the medical domain and execution environments, and the complexity of their institutional adoptions seems to be complicated. These adoptions are highly dependent on the patient's individual record, which is not using standards across different hospitals. So for each hospital, another integration needs to be done to apply the guideline to each individual patient. Furthermore, cross-community access to patient data is not foreseen.

Another approach using Activiti⁴, which is a BPMN process engine for Java, was used to call Arden-Syntax-MLM (Medical Logic Modules)⁵ over REST communication[8]. This solution eases complex clinical guidelines by splitting them into the evidence-based general workflow layer and a specific layer defined by particular institutions. Although this approach simplifies complex clinical guidelines by dividing them into the general workflow and an institute-specific medical decision-making layer, the Arden-Syntax is very explicit to the medical domain. Technical integration of current clinical decision support systems in existing, heterogeneous health information systems turns out complex due to the different data formats used[9]. There exists a commercial implementation of Arden syntax that provides different implementations to multiple data sources, also integrating with FHIR or openEHR [10]. However, this is not defined in the Arden syntax standard.

Beckmann et al. [11] describe a particular medical guideline, using Camunda modeler to create the BPMN and also using patient FHIR-Resources for making decisions. Nevertheless, the link to already existing medical knowledge (as ontologies) is missing, but it is essential to ensure the correct selection of FHIR resources for the particular use case.

In earlier research [12], we used Shape Constraint Language (SHACL)⁶ to express a simple medical guideline that existed as sequences and exclusive choices. The issue is that by using SHACL, we missed the option of the arbitrary circle workflow pattern⁷, that would help for expressing recurring tasks that are widely done in different medical guidelines.

In summary, different approaches for specific purposes exist to use BPMN in the medical domain. The BPMN model is used for representation to clarify the medical guideline. However, the patient's medical data integration from an EHR is either proprietary or not addressed.

3. Our approach: combining BPMN and FHIR-RDF

Figure 1 shows our architectural approach to integrate medical guidelines represented as BPMN models with the patients' FHIR resources. The core is a so-called "Distributed Medical Service Engine" (DMSE), which provides RESTful services to receive client requests for triggering the execution of a medical guideline. The DMSE loads and executes a BPMN model, which

³e.g., https://www.cgm.com/aut_de/produkte/krankenhaus/cgm-clinical/workflowsteuerung.html

⁴<https://github.com/Activiti/Activiti>

⁵https://www.hl7.org/implement/standards/product_brief.cfm?product_id=372

⁶<https://www.w3.org/TR/shacl>

⁷<http://www.workflowpatterns.com/patterns/control/structural/wcp10.php>

models a medical guideline. The DMSE can also retrieve patient-related information from multiple distributed FHIR stores and convert it to FHIR-RDF. The represented FHIR-RDF data is temporarily stored in a local (DMSE-Service) RDF triplestore that may be queried via SPARQL.

For the implementation of the BPMN execution engine, we are using Camunda. Camunda is an open-source platform for Business Process Management and provides a “Modeler” (for creation of BPMN models) and an “Execution engine” (for the automated processing of BPMN models). Furthermore, Camunda allows the association of the BPMN tasks with specialized Java classes that can be implemented and injected into the application’s runtime.

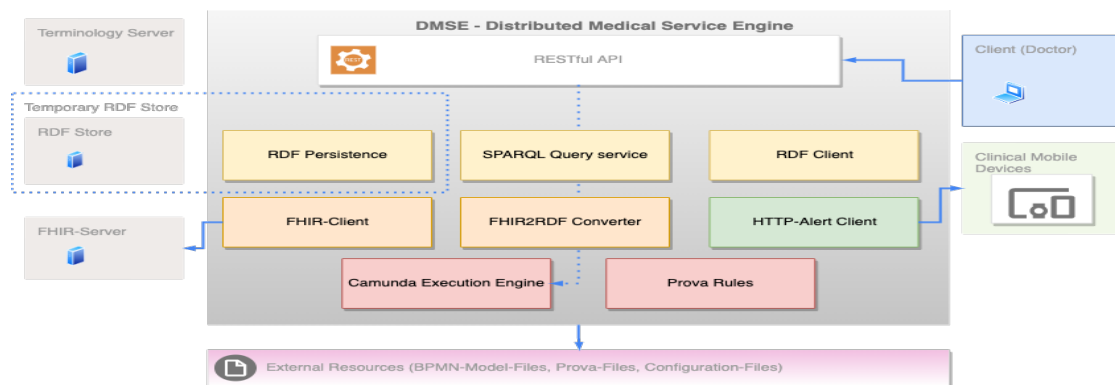


Figure 1: Systemarchitecture Distributed Medical Service Engine

The DMSE can append patient-relevant information in the form of (JSON-)FHIR resources that occur during the execution of a medical guideline.

For example, suppose the BPMN suggests applying medication for treatment. In that case, this act needs documentation for two main reasons: the legal perspective and the need to further process the BPMN with new information about the patient. The need for insertion during the runtime of a medical guideline does not only apply to medication administration; it is also highly relevant for observation resources (e.g., heart rate, blood pressure) to check in later repetitions in the execution path if the patient’s status changes.

3.1. Case study

In this paper, we exemplify our model on a particular reduced medical guideline: the “Management of asthma in adults in primary care” [13]. This guideline takes care of both the assessment and the treatment. The assessment part evaluates based on clinical features (e.g., speaking up, being confused, poor respiration effort) as well as clinical parameters (e.g., respiratory rate, pulse rate, oxygen saturation) the severity of the acute exacerbation of asthma. Depending on the severity, the treatment contains different medications to be applied.

The severity status causes, depending on local prerequisites for medical treatment, informing specialists for taking care of the medical case. The guideline mentions reevaluating the patient’s status after drug application and even during transport from the emergency scene to the hospital.

We modeled the BPMN model for the guideline, as seen in Figure 2⁸. The initial assessment

⁸https://github.com/gkober/MedicalGuidelineCamunda/blob/main/Figure_2_BPMN_reduced_Asthma_care.png

takes all the available inputs, calculates, and delivers them to the BPMN gateway for the decision about which severity status to choose (“Mild to moderate”, “Severe”, or “Life-Threatening”). The severity check is done since, during runtime, an alert is generated (to call for an emergency physician), or a notification to the emergency control center that no additional resources are needed is sent. The specific alert is defined in the BPMN model. This definition includes the facts of “participants to alert” and the criteria that need to match for raising an alert.

Subsequently, the management (treatment) process starts with recommendations on which medication to apply. After the real-life application of drugs, the execution of the BPMN process continues and decides on the transport option. The severity check starts again in case the patient is not stable enough for transportation.

To have patient-related medical information available during the execution time of the BPMN model, we added a technical task at the beginning of the process: “Calling FHIR servers and generating RDF”. This task collects all available patient-related observations and prepares the data for the initial and continuous assessment. Furthermore, these values are propagated to the subsequent tasks occurring during the workflow, allowing them to be accessed at any time.

Another essential remark for our case is that we assume that IoT devices (ECG, pulse oximeter, etc.) capture (new) medical measurements and delivers them as FHIR resources to an FHIR store. We simulate this IoT device by an integrated handler in the code that generates measurements and submits them as FHIR-observation-resources.

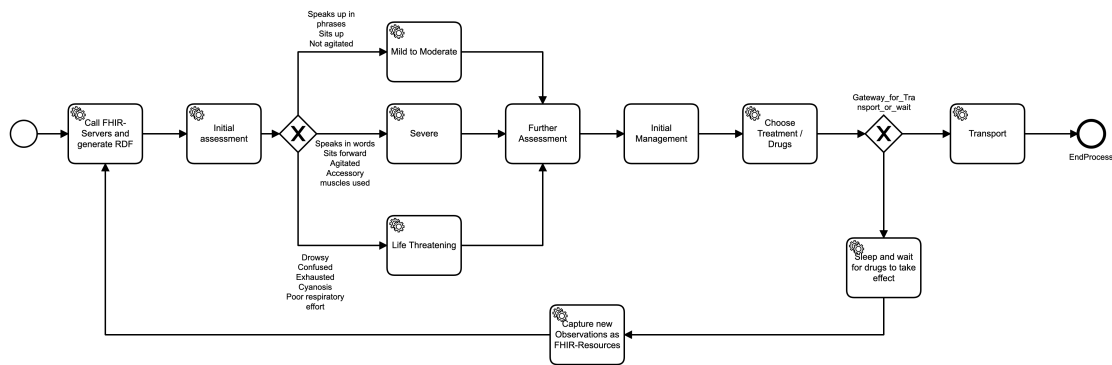


Figure 2: Reduced BPMN of Management of Asthma in primary care

3.2. Expressing FHIR as an ontology for decisions

The FHIR specification defines multiple representations for FHIR resources: JSON, XML, and RDF/Turtle [14]. The FHIR-RDF representation allows semantic linkage and integration with other available RDF data [15]. Furthermore, the FHIR-RDF representation allows rich queries that are not doable using FHIR-JSON. This means we can link the FHIR-RDF data with external ontologies like SNOMED-CT (Systematized Nomenclature of Medicine Clinical Terms), or LOINC (Logical Observation Identifiers Names and Codes). Moreover, in case a patient was in multiple hospitals and data is persisted in FHIR stores, these different, distributed FHIR stores contain patient-specific data. The FHIR standard does not allow querying all existing patient data in one single query. The RDF web data representation, having defined URIs, allows

distributed access and queries, including other (external) resources.

Contained in the FHIR resources is a reference to the type that describes what was observed. This reference description is highly relevant for understanding what kind of observation was made [16]. This description has a terminology binding to LOINC codes. For example, the meaning for “heart rate” is coded in LOINC: 8867-4⁹. This coding allows sharing the same meaning for the type of observation among everyone using the data set. These codes remove ambiguity as they state clearly the meaning of the specific values within the FHIR resource.

Another important fact is that FHIR also delivers an ontology that provides formal definitions for relationships in the RDF format. This allows using the FHIR ontology and other ontologies, like SNOMED-CT, to submit SPARQL queries or perform reasoning tasks [2].

In [17] a smart-eco-system that monitors the patient’s health status using medical wearables, intelligent agents as well as complex event processing. This paper describes also the use of reasoning, which is not yet utilized in our solution. However, the data generated and persisted during runtime is not using FHIR or other medical standard formats.

With FHIR-RDF in place, we have the option to express that a patient has a specific illness and make use of already existing medical ontologies¹⁰.

The use of SPARQL allows the BPMN executor to instantiate the needed variables with data and decide on the path. In our implementation, we use the “SELECT form where” clauses and their bindings are returned. We are formulating a specific SPARQL query to receive not all patient observations but only the needed ones relevant for the BPMN task to decide. Since the decision point in the BPMN needs a particular value, we are fetching it by expressing a SPARQL-query as shown in listing 1. In FHIR a so-called “Bundle” is a collection of FHIR resources (e.g. Observations) that can have an independent existence¹¹. In our case, this Bundle contains all observations of a specific patient and is then converted to an FHIR-RDF structure.

Listing 1: SPARQL query for Observation

```
PREFIX fhir:<http://hl7.org/fhir/>
SELECT ?value
WHERE {?root fhir:Bundle.entry ?resource.
      ?resource fhir:Bundle.entry.resource/fhir:Observation.code/fhir:CodeableConcept.
              coding/fhir:Coding.code/fhir:value ?code.
      ?value ^fhir:value/^fhir:Quantity.value/^fhir:Observation.valueQuantity/^fhir:Bundle
              .entry.resource ?resource.
      FILTER(?code="\"+code+"\").}
```

3.3. The dynamic aspect during the execution

Medical guidelines describe best practices, the values to evaluate, and which treatments to apply. In conjunction with concrete patient data, the interpretation of facts and the application of, e.g., medication applies. The values allow the BPMN execution engine to decide and trigger subsequent tasks. However, the medical guideline is usually not a simple sequence, but there are,

⁹<https://loinc.org/8867-4/>

¹⁰<https://bioportal.bioontology.org/>

¹¹<https://hl7.org/fhir/bundle.html>

among others, choices, repetitions, and merges happening. For example, when a patient receives medication, this medication impacts the patient's health and influences the observations taken. So, the measurements from a first (initial) assessment might not be correct at a later time (after providing treatment), so we need to take care of the latest observations made to ensure correct decisions during guideline execution.

In our implementation of the use case, we have to submit medication to impact the patient's observation (the execution of the BPMN has to wait for the medication to take effect). After the waiting time, new observations are captured and taken into account in the upcoming evaluation phase by re-executing the SPARQL queries. The result of these queries might lead to other decisions, which depend on the defined criteria in the BPMN, during the entire process and other outcomes of the guideline execution.

The injection of Java classes to Camunda, the BPM execution engine, allows us to implement handlers to create JSON-FHIR-Resources, and submit them to an FHIR store. We are using JSON for data exchange, in the direction from the DMSE to a remote FHIR store.

From a technical perspective, the submission has two advantages: firstly, the FHIR resource is usable for other applications only relying on JSON objects, and secondly, it persists in a reliable FHIR store and does not need to be held permanently in the DMSE. Since our application only holds data temporarily, we choose to re-query the entire data. This re-query is necessary since other IoT devices could have added new information to the FHIR store, that is relevant to the guideline execution process - therefore we need to regenerate the RDF structure.

If the DMSE acts as a proxy for FHIR-JSON resources, these re-queries would not be necessary since the DMSE can capture the incoming data and apply it to medical guidelines.

The code for execution and the synthetic datasets are available on Github¹² and this project contains the classes that extend the capabilities of Camunda, as well as the initiating FHIR Observation samples. Also, the BPMN of the implemented medical guideline is provided.

4. Evaluation and Results

In our work, we managed to separate the content of the medical perspective and the technical challenges from each other. The BPMN model defines the flow, decisions, and specialized execution classes. These execution classes manage the computer-science-relevant aspects, like how the queries to FHIR stores, transformations to FHIR-RDF are done, and even the executed SPARQL queries are abstracted from the user's perspective.

The BPMN allows us to express the medical guideline as workflow patterns: we have seen the following patterns in the current medical guideline: *Sequence*, *Exclusive Choice* (supported by X-OR-Gateway), *Simple Merge* (represented as X-OR-Join gateway), and *Arbitrary Cycles*.

For evaluating the software solution and the BPMN, we created different samples of synthetic datasets that simulate a medical emergency and the appropriate patient information in the local FHIR store. The combination of these datasets resolve to the different types of outcomes for the decisions to be taken: a mild-to-moderate illness, a severe illness, and a life-threatening illness. The acceptance of the guideline execution was done by manual validation. The synthetic datasets are generated as FHIR-resources, containing the relevant aspects for a FHIR-observation

¹²<https://github.com/gkober/MedicalGuidelineCamunda>

(e.g., code, subject, valueCodeableConcept). The creation was based on the possible values for the medical guideline. In the beginning we created randomly FHIR-resources for 25 patients. As initial tests failed, we had to align the samples, so that from a medical perspective the content of the FHIR-resources was coherent and valid.

When using these generated datasets, a situation occurred that data was not applicable for the BPMN to find a decision. This was the case when patient data were not clearly separated but represented a mixture of outcome options for the decision point. The BPMN execution engine can not take care of such a combined condition.

Another finding during the execution of the BPMN and the integration of the different FHIR data samples was, that we noticed that errors might occur if the modeler of the BPMN is not precise in the definition. The ambiguous definition of the X-OR-gateways led to the problem that the execution engine could not decide, and the entire process threw an exception.

Finally, if the BPMN and the FHIR data are available in the correct format, the test solutions provided the expected results in terms of correctness and completeness.

5. Discussion & Conclusion

The use of medical guidelines is highly relevant for patients to receive evidence-based medical treatment. Since patients are approaching different hospital facilities, their individual data is stored in multiple locations. A treating physician faces the challenge of taking care of the available data and also keeping track of the current medical state of the art. Therefore we propose a framework that applies current medical guidelines to accessible clinical (FHIR) data by addressing the technical capabilities of Semantic Web technologies.

This paper shows a possible approach to implement and assess a medical guideline in a formal executable workflow. The approach presented here made use of multiple Semantic Web sources while modeling a clinical guideline as BPMN workflows. By using FHIR, it was possible to include LOINC-codes, which define the exact meaning of the FHIR resource's values.

Manual implementation of the medical guideline can lead to inaccurate definitions and misunderstanding. Formally defined values, e.g., LOINC codes, can reduce the risk of ambiguity.

FHIR is an HL7 specification that defines the way of exchanging and representing data in the medical domain. In our use case, we are only using *FHIR-Observations* and *FHIR-MedicationAdministration* resources. In particular, we use different defined representation formats in the different phases of the execution process. For example, for creating additional FHIR resources, we use FHIR-JSON, in order to provide additional data that would stem in the real world from another (IoT-) system. Secondly, the system relies on standard FHIR search queries (HTTP-GET) for multiple FHIR stores. Thirdly, we combine the fetched content to RDF and perform SPARQL queries.

In contrast to RDF and SPARQL, the FHIR specification partly defines specialized HTTP-GET queries, which allows query parameters for date and codes, as well as sorting and counting. The behavior of such an FHIR-HTTP-Get query is heavily dependent on the used implementation of the FHIR specification and the configured capabilities of the used FHIR servers. Furthermore, the result of such queries is an FHIR resource¹³ and can not be used efficiently for a decision.

¹³<http://hl7.org/fhir/search.html>

A particular value must be extracted for further processing in the BPMN execution. So there are many dependencies on the FHIR store to allow such queries, and if such type of search is not enabled, the entire solution will fail. This means that the plain FHIR-specified capabilities are causing problems regarding standardized queries, concrete implementations, and necessary additional processing for data extraction.

To resolve these insufficiencies, we are using the RDF approach of the FHIR standard. By using this approach, we can retrieve all patient-relevant data from different FHIR stores and convert the JSON representation to RDF, including the FHIR ontology. This also allows for integrating other ontologies like LOINC and using it for reasoning tasks. This ensures that the meaning of an individual observation is the same for all participants.

For performing queries towards the RDF representation, we use SPARQL queries returning single values to the BPMN execution for the decision points. We find these values by linking the value to the code using the SPARQL property path [18] and adding a relation to the date (in the observation case, the effectiveDateTime). These (SPARQL) standard queries are easier to understand for a developer than FHIR queries, because these are very specific for the medical domain; thus SPARQL allows integration of other non-medical resources from the Web.

The medical guideline mentions these textual descriptions that can be found as values in LOINC codes for the patient survey. The use of LOINC codes increases the understanding and meaning of values and removes the ambiguity that would disrupt the BPMN execution. We use these codes as input for the SPARQL query in the patient data and then use them in the BPMN execution environment to make decisions on the clinical treatment path.

The combination of BPMN, FHIR, and Semantic Web technologies allows for the creation of a system that backs clinical decision support, ensuring the usage of evidence-based medical guidelines. In comparison to [1], we extend the usage of BPMN to medical data, and in contrast to [8], we also incorporate FHIR-data, ontologies, and SPARQL for taking decisions.

In the future, it is foreseen to integrate in our framework rules in the RuleML standard [19] or its extensions (e.g., LegalRuleML). RuleML allows to encode and reason with more complex constraints, such as obligations and prohibitions, of both medical and legal nature. Representing these in modal logic has always been problematic from the computational complexity point of view, while the use of rules offers a simpler and more practical solution [20].

With respect to related contemporary proposals, e.g., [17], the resulting framework will offer a solution grounded on widely-acknowledged standards (FHIR, RDF, BPMN, and RuleML), which will facilitate and encourage funnelling of efforts from people with heterogeneous competences (doctors, pharmaceutical companies, lawyers, etc.), as well as sharing and reuse of the resources.

Acknowledgments

Livio Robaldo has been supported by the Legal Innovation Lab Wales operation within Swansea University's Hillary Rodham Clinton School of Law. The operation has been part-funded by the European Regional Development Fund through the Welsh Government.

References

- [1] L. Pufahl, F. Zerbato, B. Weber, I. Weber, BPMN in healthcare: Challenges and best practices, *Information Systems* 107 (2022).
- [2] FHIR-RDF, HL7 FHIR: RDF, 2022. URL: <https://www.hl7.org/fhir/rdf.html>.
- [3] S. Rodriguez-Loya, A. Aziz, C. Chatwin, A Service Oriented Approach for Guidelines-based Clinical Decision Support using BPMN, *Studies in Health Technology and Informatics* 205 (2014).
- [4] M. Balser, C. Duelli, W. Reif, Formal semantics of Asbru - an overview, in: *Proceedings of the International Conference on Integrated Design and Process Technology*, 2002.
- [5] S. W. Tu, M. A. Musen, Modeling data and knowledge in the EON guideline architecture, in: *MEDINFO 2001*, IOS Press, 2001.
- [6] M. Peleg, A. A. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, et al., GLIF3: the evolution of a guideline representation format., in: *Proc. of the AMIA Symposium*, American Medical Informatics Association, 2000.
- [7] J. Fox, N. Johns, A. Rahmzadeh, Disseminating medical knowledge: the PROforma approach, *Artificial intelligence in medicine* 14 (1998).
- [8] J. S. De Bruin, K. P. Adlassnig, H. Leitich, A. Rappelsberger, Separating Business Logic from Medical Knowledge in Digital Clinical Workflows Using Business Process Model and Notation and Arden Syntax, *Studies in Health Technology and Informatics* 248 (2018).
- [9] M. Samwald, K. Fehre, J. De Bruin, K.-P. Adlassnig, The Arden Syntax standard for clinical decision support: Experiences and directions, *Journal of biomedical informatics* 45 (2012).
- [10] Clinical applications and Arden Syntax, Accessed: 2022-11-01. URL: https://www.medexter.com/ardensuite/educational_material_part5_clinical_applications_and_arden_syntax.pdf.
- [11] C. L. Beckmann, G. Lodde, E. Livingstone, D. Schadendorf, B. Böckmann, Guideline-Based Context-Sensitive Decision Modeling for Melanoma Patients, *Studies in Health Technology and Informatics* (2022).
- [12] G. Kober, L. Robaldo, A. Paschke, Modeling Medical Guidelines by Prova and SHACL Accessing FHIR/RDF., in: *dHealth 2022*, IOS Press, 2022.
- [13] A. Y. L. Ban, A. Omar, L. Y. Chong, H. Lockman, Z. A. Ida Zaliza, I. A. H. Ali, etc., Management of asthma in adults in primary care, *Malaysian Family Physician* 13 (2019).
- [14] FHIR-Formats, HL7 FHIR: Formats, 2022. URL: <https://www.hl7.org/fhir/formats.html>.
- [15] E. Prud'hommeaux, J. Collins, D. Booth, K. J. Peterson, H. R. Solbrig, G. Jiang, Development of a fhir rdf data transformation and validation framework and its evaluation, *Journal of Biomedical Informatics* 117 (2021).
- [16] O. Bodenreider, R. Cornet, D. J. Vreeman, Recent developments in clinical terminologies—snomed ct, loinc, and rxnorm, *Yearbook of medical informatics* 27 (2018).
- [17] S. Costantini, F. Persia, L. De Laurentis, An Intelligent Ecosystem to improve Patient Monitoring using Wearables and AI, *CEUR Workshop Proceedings* (2022).
- [18] E. V. Kostylev, J. L. Reutter, M. Romero, D. Vrgoč, Sparql with property paths, in: *International Semantic Web Conference*, Springer, 2015.
- [19] H. Boley, A. Paschke, O. Shafiq, RuleML 1.0: The Overarching Specification of Web Rules, in: M. Dean, J. Hall, A. Rotolo, S. Tabet (Eds.), *Semantic Web Rules*, Springer, 2010.
- [20] X. Sun, L. Robaldo, On the complexity of I/O logic, *Journal of Applied Logic* 25 (2017).