

Article

Semantic Integration of BPMN Models and FHIR Data to Enable Personalized Decision Support for Malignant Melanoma

Catharina Lena Beckmann^{1,*}, Daniel Keuchel^{2,†}, Wa Ode Iin Arliani Soleman¹, Sylvia Nürnberg² and Britta Böckmann^{1,2}

¹ Department of Computer Science, University of Applied Sciences and Arts Dortmund (FH Dortmund), 44227 Dortmund, Germany; britta.boeckmann@fh-dortmund.de (B.B.)

² Institute for Artificial Intelligence in Medicine, University Hospital Essen, Girardetstraße 2, 45131 Essen, Germany; daniel.keuchel@uk-essen.de (D.K.); sylvia.nuernberg@uk-essen.de (S.N.)

* Correspondence: catharina.beckmann@fh-dortmund.de

† These authors contributed equally to this work.

Abstract: With digital patient data increasing due to new diagnostic methods and technology, showing the right data in the context of decision support at the point of care becomes an even greater challenge. Standard operating procedures (SOPs) modeled in BPMN (Business Process Model and Notation) contain evidence-based treatment guidance for all phases of a certain diagnosis, while physicians need the parts relevant to a specific patient at a specific point in the clinical process. Therefore, integration of patient data from electronic health records (EHRs) providing context to clinicians is needed, which is stored and communicated in HL7 (Health Level Seven) FHIR (Fast Healthcare Interoperability Resources). To address this issue, we propose a method combining an integration of stored data into BPMN and a loss-free transformation from BPMN into FHIR, and vice versa. Based on that method, an identification of the next necessary decision point in a specific patient context is possible. We verified the method for treatment of malignant melanoma by using an extract of a formalized SOP document with predefined decision points and validated FHIR references with real EHR data. The patient data could be stored and integrated into the BPMN element 'DataStoreReference'. Our loss-free transformation process therefore is the foundation for combining evidence-based knowledge from formalized clinical guidelines or SOPs and patient data from EHRs stored in FHIR. Processing the SOP with the available patient data can then lead to the next upcoming decision point, which will be displayed to the physician integrated with the corresponding data.

Keywords: standard operating procedures; clinical pathways; computer-interpretable clinical guidelines; BPMN; HL7 FHIR; model transformation; clinical decision-making; patient-specific modeling; malignant melanoma



Citation: Beckmann, C.L.; Keuchel, D.; Soleman, W.O.I.A.; Nürnberg, S.; Böckmann, B. Semantic Integration of BPMN Models and FHIR Data to Enable Personalized Decision Support for Malignant Melanoma. *Information* **2023**, *14*, 649. <https://doi.org/10.3390/info14120649>

Academic Editor: Evaggelos Karvounis

Received: 6 November 2023

Revised: 29 November 2023

Accepted: 30 November 2023

Published: 6 December 2023

Corrected: 4 September 2024



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Providing the right data at the right time during the patient treatment process helps improve and accelerate decision-making [1,2]. The significant influx of patient data due to digitization in general, but also new data-intensive methods and therapies, has made the selection of the right data and integration with evidence-based knowledge an even greater challenge. Thus far, the integration of patient data with clinical guidelines and standard operating procedures (SOPs) as a means to provide evidence-based knowledge has been limited in practice [3]. Furthermore, the use of patient data to determine the correct section within large guideline documents is essential to enhance usage supporting the right decision at the point of care. Physicians still in practice must work with text-based guidelines and SOPs alongside a vast amount of patient data to link both manually, a time-consuming and uncertain practice that calls for automation. To link digitized health

data with digitally modeled clinical guideline knowledge represents a promising approach to bridging this gap and to unlocking new perspectives to move toward a data-driven and responsive healthcare system.

This proposed integration needs to be developed using existing standards to be reproducible. In the field of healthcare, a range of standards already exists, holding the potential to facilitate this convergence and finally support decision-making processes.

A widely adopted standard for process visualization and modeling is the intuitive graphical notation BPMN (Business Process Model and Notation) [4,5], initially developed by the Business Process Management Initiative (BPMI) and maintained by the Object Management Group (OMG). Due to its intuitive notation, BPMN is widely used for modeling clinical processes using Clinical Practice Guidelines [6,7]. A recent systematic review assessed the impact of BPMN on the healthcare sector, demonstrating its efficacy as a tool for standardizing and redesigning clinical processes [8].

For data exchange and the storage of healthcare information, a variety of standards, e.g., HL7 (Health Level Seven) version 2 (“v2”) and version 3 (“v3”) or openEHR, exist. The newest and widely adopted communication standard is HL7 FHIR (Fast Healthcare Interoperability Resources) [9], also motivated by different governmental initiatives for standardization. Thus, several consecutive projects have concentrated on the representation of clinical guidelines within the FHIR framework. For instance, the CPG-on-FHIR project [10,11] focuses on representation, the development of a FHIR implementation guide for machine-readable clinical practice guideline recommendations (see the Clinical Practice Guidelines (CPGs) on the EBMonFHIR project [10]), and the integration of clinical practice guidelines and data for decision support, as demonstrated by the CELIDA (COVID-19 Clinical Guidelines to Data Mapper) project [12]. Building upon these previous endeavors, the clinical decision support system GREvaluator (Guideline Recommendation Evaluator) [12,13] facilitates the verification of individual guideline adherence. Additionally, both standards have found applications in workflow orchestration or (partial) process automation, exemplified by tools such as [14–16], as well as in automated cohort size estimation for research purposes within HiGHmed [17–22], and by additional incorporation of Semantic Web technologies to improve decision support (see [23]).

To provide meaningful decision support at the point of care, both standards, BPMN and FHIR, are required. The need for a direct link between BPMN models and real patient data has previously been highlighted [24] as the only way to accurately identify a specific section or decision point within a formalized SOP or clinical guideline based on the data of a particular patient. Once identified, to support the physician, we need to integrate the available patient data for this decision point and present the missing but required data.

Over the past three years, significant efforts have been dedicated to harnessing the synergies of these two pivotal standards within the healthcare sector. To facilitate computer-interpretable representations of clinical-guideline-based knowledge, combined BPMN–FHIR modeling approaches have emerged [25,26]. Other works have focused on mapping approaches, encompassing mapping different ontologies to FHIR (see [27]) or mapping BPMN core elements to FHIR elements, as demonstrated in [28–33]. One approach addresses our problem statement by offering a mapping solution and bidirectional transformation between BPMN and FHIR [34]. Building upon this fundamental concept, we additionally enable simultaneous patient-specific FHIR data access during the transformation process.

In summary, we find significant ongoing research in the field of computer-interpretable guidelines, as well as in the mapping or transformation between BPMN and FHIR. However, combining this mapping with the integration of electronic health record (EHR) data is still unsolved. This work presents a model to prepare the identification and presentation of a specific decision point within an SOP based on the available patient data. Therefore, the SOP has been modeled in BPMN and the patient data are stored in FHIR. Based on this model, the algorithm can then traverse all decision points using the available patient data until a decision point is reached where no more data are available. Thus, the current

position of the patient in the SOP according to the EHR data can be determined. In clinical practice, medical staff will then no longer have to spend time searching for patient data in hospital information systems to figure out the next treatment steps. Also, the quality of patient care might be improved by eliminating human errors and ensuring that tests already performed are not overlooked or repeated.

We needed to address the following challenges—to ensure loss-free TRANSFORMATION between BPMN and FHIR and incorporate the DATA STORAGE aspect into BPMN.

2. Materials and Methods

The BPMN standard is used for modeling processes in an intuitive way. It consists of semantically precise graphical symbols for tasks, gateways, events, and their various types. Each element represents an individual step in a process, contributing to the process flow. In BPMN, subprocesses serve as essential elements for grouping steps or restructuring complex processes.

FHIR basically consists of resources and APIs (Application Programming Interfaces). The resources are a collection of information models that define the data elements, constraints, and relationships for the “business objects” most relevant to healthcare. Therefore, they serve as the foundation for all interchangeable content. Each resource includes a standard definition and human-readable description of use. Additionally, every resource comes equipped with a set of common and resource-specific metadata, which ensures its use is clear and unambiguous. These resources can be further customized into profiles to align with the requirements of specific use cases. The APIs build a collection of well-defined interfaces for interoperating between two applications.

Our overall approach (Figure 1) starts with a data preprocessing step (DATA STORAGE) extending an existing BPMN model using FHIR patient data taken from the EHR. The EHRs are sourced from the Smart Hospital Information Platform (SHIP) [35], which consolidates patient data from diverse information systems, including Hospital Information Systems (HIS), Laboratory Information Systems (LIS), Radiology Information Systems (RIS), and Pathology Information Systems (PIS). This extended model is then further processed by a TRANSFORMATION process, mapping the different elements of the standards, and subsequently transforming between BPMN and FHIR.

To initiate the TRANSFORMATION process, we first validated the BPMN model to ensure the correct BPMN syntax. Based on the results, we conducted an analysis to determine the most suitable modeling approach for integrating patient-specific data during the DATA STORAGE phase by performing manual annotations on the model. The TRANSFORMATION process starts with the mapping of a total of 14 BPMN elements to FHIR fields, followed by the actual transformation. FHIR provides the ‘PlanDefinition’ (see [31]) resource, which is particularly valuable for defining clinical pathways and has already been utilized by researchers [16,33,34]. The resulting BPMN-to-FHIR transformed model underwent syntax validation using Postman. Subsequently, we performed the reverse transformation back into BPMN, comparing the input BPMN model with the BPMN model obtained from FHIR to ensure a loss-free transformation approach. For the transformation process, we employed two Camunda libraries (see [36]) to work with the BPMN elements in XML (Extensible Markup Language) format. For handling the FHIR data, we used the HAPI library [37], which is designed to make it easier to work with FHIR data, both for retrieving and storing healthcare information, and for building applications that use FHIR as their data format. It provides a set of tools and libraries for working with FHIR resources, making it a valuable resource for healthcare software development and interoperability in the medical field. To ensure proper access to patient data, we utilized FHIR Release 4, as Release 5 had not yet been implemented here. For the syntax check of the FHIR resources, we utilized the ‘validate’ method in Postman for the HAPI FHIR.

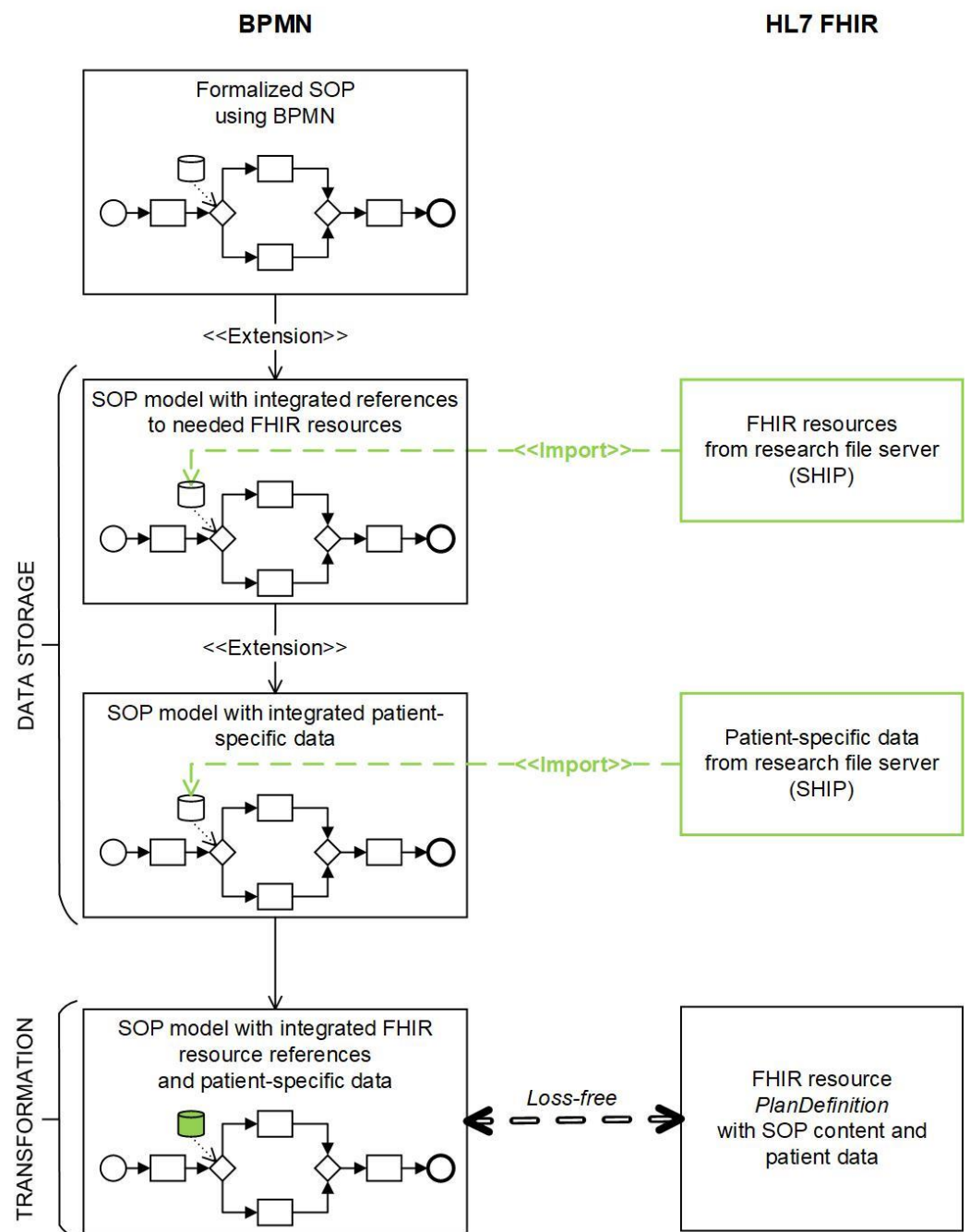


Figure 1. The overall approach of embedding FHIR patient data into BPMN models and the transformation between both standards—BPMN models (left-side) and FHIR-related data (right-side).

For DATA STORAGE, we then verified the FHIR resources associated with the defined decision points in the BPMN model against real EHR data from the University Medicine Essen (UME), ensuring the correct positioning of the data parameters within the JSON (Java Script Object Notation) structure. If necessary, we adjusted the FHIR resources and manually integrated them into the appropriate BPMN modeling elements. The Python package FHIR-PYrate [38] was used to query the FHIR data from the FHIR server and return it as structured data.

To verify our model, we chose malignant melanoma as the medical model and used an SOP transformed into a BPMN model from a previous study [25].

The data used for verification are anonymized patient data from SHIP, retrospectively obtained for research purposes. SHIP includes EHRs in FHIR in JSON data format con-

taining 3429 melanoma patients (as of 23 October 2023), including 51 adjuvant-treated patients for the SLNE (sentinel lymph node excision) treatment section presented here, each based on ICD-10-GM (International Statistical Classification of Diseases and Related Health Problems, 10th revision, German Modification) code C43. We selected the patient cohort receiving adjuvant treatment from the Department of Dermatology, specifically gathering data required for the SLNE treatment step.

Additionally, we conducted syntax checks on the BPMN models before and after the transformation using bpmnlint [39], which validates the diagram against a defined set of rules and reports these as errors or warnings.

3. Results

3.1. Loss-Free Mapping as a Foundation for the TRANSFORMATION Approach between BPMN and FHIR

For the implementation of the BPMN elements, we adhered to the structure outlined in the BPMN 2.0 specifications [4]. To focus on incorporating patient data, our implementation of BPMN elements was limited to the Basic Control Flow Patterns [40,41], alongside data elements. These patterns encompass Sequence, Parallel Split, Synchronization, Exclusive Choice, and Simple Merge.

In the developed mapping (Table 1), we utilized the FHIR resource ‘PlanDefinition’ as recommended by an OMG presentation [31] and the FHIR Clinical Practice Guidelines Implementation Guide (CPG IG) [42]. The BPMN elements listed in the table are categorized under flow objects, connecting objects, artifacts, and data, aligning with the ‘five basic element categories’ defined by the OMG (see [4], p. 27). Each BPMN element (Table 1, left column) was mapped to the corresponding field in the FHIR resource ‘PlanDefinition’ (Table 1, right column).

Table 1. General table of all BPMN elements mapped to FHIR ‘PlanDefinition’ fields.

BPMN Elements	HL7 FHIR ‘PlanDefinition’
Flow Objects	
Activities	
Task	action
User Task	action.description=‘userTask’
Service Task	action.description=‘serviceTask’
Subprocess	The preceding BPMN element references the start element of the subprocess and all subsequent elements of the subprocess are subactions of it in FHIR. The end element of the subprocess references the subsequent BPMN element.
Events	action
Start	action which has no actions refers to it
End	action with no related action
Gateways	action
Parallel Gateway	action.selectionBehavior=‘all’
Exclusive Gateway	action.selectionBehavior=‘exactly-one’
Connecting Objects	
Sequence Flow	action.relatedAction
Data Input Association, Data Output Association	Not needed (Comment: These elements point to ‘DataStoreReference’, and because the ‘DataStoreReference’ in FHIR is directly included in action, no association is needed.)
Artifacts	
Annotation	action.relatedArtifact
Data	
DataStoreReference	action.input.requirement/ action.output.requirement
Name	DataRequirement.codeFilter.path
Value	DataRequirement.codeFilter.searchParam

Based on this mapping, a transformation was implemented to convert BPMN models into FHIR code and back (see [43]). The transformation of all BPMN elements of the Basic Control Flow Pattern could be performed loss-free to FHIR and back to BPMN (see Section 3.3). During transformation, focus was on content information, such as BPMN elements, order, and name. Technical information such as the IDs of elements or the BPMN Diagram Interchange (BPMN DI), which contains information on the basic visual layout of BPMN diagrams, was not considered.

The mapping between the BPMN element ‘DataStoreReference’ and the FHIR resource ‘PlanDefinition’ is shown in Table 2.

Table 2. Example BPMN element ‘DataStoreReference’ including patient data and the corresponding HL7 FHIR resource ‘PlanDefinition’.

BPMN	<pre> <serviceTask id="Activity_1" name="Extract tumor thickness"> <incoming>Flow_1</incoming> <outgoing>Flow_2</outgoing> <property id="Property_1" name="__targetRef_placeholder"/> <dataInputAssociation id="DataInputAssociation_1"> <sourceRef>DataStoreRef_1</sourceRef> <targetRef>Property_1</targetRef> </dataInputAssociation> </serviceTask> <dataStoreReference id="DataStoreRef_1" name="Tumor thickness"> <extensionElements> <properties> <property name="fhir:Observation.where(code.coding. code='B1AAYPBTUDI').valueQuantity.value" value="2.9"/> </properties> </extensionElements> </dataStoreReference> </pre>
HL7 FHIR ‘PlanDefinition’	<pre> <action id="Activity-1"> <title value="Extract tumor thickness"></title> <description value="serviceTask"></description> <input> <type value="DataRequirement"></type> <mustSupport value="Tumor thickness"></mustSupport> <codeFilter> <path value="fhir:Observation.where(code.coding.code= 'B1AAYPBTUDI').valueQuantity.value"> </path> <searchParam value="2.9"></searchParam> </codeFilter> </input> <relatedAction> <actionId value="Gateway-1"></actionId> <relationship value="before"></relationship> </relatedAction> </action> </pre>

The top row contains the BPMN element ‘DataStoreReference’ holding the patient data. Here, the task named ‘Extract tumor thickness’ is linked to the parameter ‘Tumor thickness’. The name of the property pertains to the FHIR resource for tumor thickness in the database. In our case, tumor thickness in SHIP is located under “fhir:Observation.where (code.coding.code=‘B1AAYPBTUDI’).valueQuantity.value” (Table 2).

To access patient information, we utilized FHIR Search to execute a query with the patient ID in the required resource of the database. The corresponding value of the patient's tumor thickness is then inserted into the 'Value' field.

3.2. DATA STORAGE Aspect Realized in BPMN

To establish a reference to patient data, BPMN already offers options for accessing external data. These options include Data Objects, Data Inputs, Data Outputs, and Data Stores ([4], p. 28). We chose Data Stores in our case, as the patient data originate from an external database. According to OMG, a Data Store is defined as follows: "A DataStore provides a mechanism for Activities to retrieve or update stored information that will persist beyond the scope of the Process. The same DataStore can be visualized, through a Data Store Reference, in one or more places in the Process." ([4], p. 208).

The 'DataStoreReference' elements can be used to incorporate generic or technical information using name-value pairs and can be extended with any number of property fields (Figure 2) and can thus be linked to all BPMN tasks relating to patient data. For each data field (e.g., tumor thickness, ulceration, tumor stage, patient age), a property is created in the corresponding 'DataStoreReference'. These properties, in turn, consist of two fields: 'Name', and 'Value', where we specify the FHIR path (FHIR resource and field in the SHIP database) linking to the patient data from the 'Name' field. This mapping is performed manually by associating the relevant FHIR resources with the patient data to be reviewed at subsequent decision points (Figure 2 center, at the gateway 'Which tumor thickness (TT) and risk factors are present?'). The value is retrieved from the selected patient's FHIR resource using the FHIR-PYrate package and written into the 'Value' field. This enables the incorporation of the context-sensitive FHIR patient data at the respective gateways.

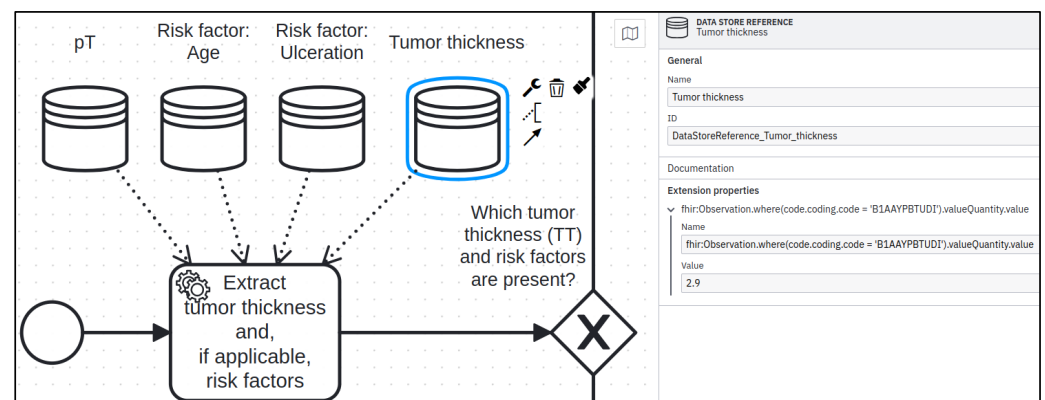


Figure 2. Screenshot (translated into English) of the tool Camunda Modeler [44] with the BPMN element 'DataStoreReference' including integrated patient data using the tool's 'properties panel' (example parameter 'tumor thickness').

3.3. Generic Transformation Algorithm to Enable Personalized Decision Support

This section describes the merging of our mapping approach with the DATA STORAGE aspect in order to provide a general algorithm in pseudocode for a loss-free BPMN-to-FHIR transformation with real EHR data.

In Figure 3, we present a pseudocode representation of our general BPMN-to-FHIR transformation process aiming to generate a FHIR resource 'PlanDefinition' out of a BPMN model. During the process, all BPMN elements are mapped to action fields of the 'PlanDefinition'. Each transformation process starts with the provision of a BPMN input file and the initialization of an output file in XML format. Next, we use Camunda libraries [36] to parse all the BPMN elements into Java objects, creating a list of BPMN elements in Java. The transformation steps are different for three categories of requirements:

- BPMN elements with direct transformation: These elements, including tasks, events, gateways, as well as 'DataStoreReferences' existing at the top level (not within sub-

- processes), can directly be transformed into the corresponding FHIR fields (Figure 3, lines 3–5).
- BPMN elements requiring adjustments: This category contains elements like subprocesses, elements following gateways, as well as text annotations needing adjustments during transformation (Figure 3, lines 6–9).
 - Mandatory FHIR requirements: Specific requirements outlined by the FHIR standard, including certain characters required for the FHIR server, unique IDs, and the mandatory field “fhir:PlanDefinition.status” (Figure 3, lines 10–12).

```

Procedure TransformBPMNtoFHIR(inputFileBPMN):
1. Initialize outputFileFHIR as XML
2. javaObjects = parseBPMNtoJavaObjects(inputFileBPMN)
3. javaObjects = setDataRequirement()
4. lastElementsCache = getLastElementsInSubprocesses()
5. javaObjects = ConvertTopLevelElements(lastElementsCache)

6. javaObjects.setSubActions()
7. javaObjects.setCondition()
8. javaObjects.mergeActionsAndSubActions()
9. javaObjects.setRelatedArtifacts()

10. javaObjects.replaceCharacter()
11. javaObjects.deleteIdDuplicates()
12. javaObjects.setPlanDefinitionStatus()

13. javaObjects.validateFHIR()
14. javaObjects.serializeToXML(outputFileFHIR)

End Procedure

```

Figure 3. Pseudocode of the general algorithm of loss-free BPMN-to-FHIR transformation using real EHR data.

More specifically, after parsing, the algorithm performs as follows: First, the BPMN ‘DataStorageReference’ elements, including the EHR patient data, are mapped into “fhir:PlanDefinition.action.input.requirement” (Figure 3, line 3). During this phase, the patient data are then processed according to the mapping described in Table 1 with an example shown in Table 2. We then proceed to cache all end events from subprocesses to preserve their endpoints (Figure 3, line 4). This is crucial, as subprocesses cannot be translated directly into FHIR. Consequently, all elements within a subprocess are then set as subactions of the start event within the subprocess. Subsequently, all BPMN elements at the top level, excluding those within subprocesses, are converted into “fhir:PlanDefinition.action” and added to the Java objects (Figure 3, line 5).

To ensure consistency in the sequence of BPMN elements after reverse transformation, we set all actions directly following a gateway as subactions of the corresponding gateway (Figure 3, line 6). It is also essential to ensure that sequence flows from divergent gateways reference the correct next element, which is addressed in FHIR using the “fhir:PlanDefinition.action.condition” field (Figure 3, line 7). All converted elements, regardless of their modeling level, are merged (Figure 3, line 8). For BPMN elements with additional documentation in text annotations, we assign “fhir:PlanDefinition.relatedArtifact” (Figure 3, line 9).

The next phase involves adjustments mandated for FHIR servers. This encompasses character replacement for element IDs not permissible for FHIR servers (Figure 3, line 10), the removal of ID duplicates caused by the transformation of diverging gateways for actions, the storage of the ID of an action after a gateway in the URL field (Figure 3, line 11), and compliance with the mandatory FHIR criteria for setting “PlanDefinition.status” (Figure 3, line 12).

Finally, the created ‘PlanDefinition’ resource is validated and serialized into XML format (Figure 3, lines 13–14).

The pseudocode for the other direction—generating a BPMN model from a FHIR resource—is shown in Figure 4. The objective is to transform all information contained in a FHIR resource ‘PlanDefinition’, including FHIR patient data, into a BPMN model without any loss of data.

Procedure TransformFHIRtoBPMN(inputFileFHIR):

1. Initialize outputFileBPMN
 2. javaObjects = parseFHIRtoJavaObjects(inputFileFHIR)
 3. javaObjects.setID()
 4. javaObjects.replaceCharacter()
 5. javaObjects.setStartEvent()
 6. javaObjects.setEndEvent()
 7. javaObjects.setSubProcess()
 8. subProcessElementsCache = storingSubProcessElements()
 9. javaObjects.setTask()
 10. javaObjects.setGateway()
 11. javaObjects.setMergeGateway()
 12. javaObjects.setSequenceFlow()
 13. javaObjects.setSubProcessElements(subProcessElementsCache)
 14. javaObjects.validateBPMN()
 15. javaObjects.serializeToBPMN(outputFileBPMN)
- End Procedure

Figure 4. Pseudocode of the general algorithm of loss-free FHIR-to-BPMN reverse transformation, using real EHR data.

In general, the reverse FHIR-to-BPMN transformation is performed processing the following steps: provision of a FHIR input file and initialization of an output file in BPMN format, parsing (Figure 4, lines 1–2), readjustments due to mandatory requirements of the FHIR standard (Figure 4, lines 3–4), preprocessing of the BPMN top-level elements (Figure 4, lines 5–7), followed by the transformation into the individual BPMN elements (Figure 4, lines 9–12), handling subprocesses (Figure 4, lines 8,13), validation, and serialization.

In more detail, the reverse transformation from FHIR into BPMN starts with converting XML objects into Java objects. This step includes assigning IDs to actions lacking IDs in FHIR due to multiple actions sharing the same ID (Figure 4, line 3). This regulation has been addressed previously in the forward transformation (Figure 3, line 11). In addition, the characters from FHIR are substituted with acceptable ID characters (Figure 4, line 4), corresponding to the inversion of the forward transformation (Figure 3, line 10).

To identify the start and end events, the algorithm searches for fields in “PlanDefinition.action” either with no actions referring to them (start events) or no related actions (end events). These events are then transformed into BPMN (Figure 4, lines 5–6). Subsequently, the basic structure of the subprocesses is created at the top level of BPMN modeling, along with their respective start and end events (Figure 4, lines 7). All subprocess information is cached, with the start event acting as the key element (Figure 4, line 8).

Iterating through all FHIR actions, the transformation into the BPMN elements for tasks and gateways is executed first (Figure 4, lines 9–10), followed by the merge gateways and sequence flows (Figure 4, lines 11–12). Subsequently, all created BPMN elements belonging to a subprocess are placed as child elements (Figure 4, line 13) to restore the lower modeling level created by the BPMN subprocesses. For all set BPMN elements allowing a link to a ‘DataStoreReference’ according to the specification, the FHIR fields `action.input.requirement` and `action.output.requirement` are checked to see whether values are available. If found, these values are extracted from the FHIR fields `DataRequirement.codeFilter.path` and `DataRequirement.codeFilter.searchParam` and inserted into the BPMN property extensions ‘Name’ and ‘Value’. The transformation concludes with validating the created file and serializing it into a BPMN file (Figure 4, lines 14–15).

As a final adjustment, we have supplemented our implemented FHIR-to-BPMN transformation algorithm with the functionality of data associations. Thus, the linking of ‘DataStoreReferences’ to the related BPMN elements are ensured in the terms of modeling.

To validate our presented approach, data from the 51 melanoma patients being treated in the adjuvant setting have been added into the BPMN model of the SLNE section of the SOP document. In the next step, data from the generated BPMN model have been transformed into FHIR and back into BPMN. All information of the BPMN elements—the order, names, as well as the retrieved patient data relevant to this SOP—could be preserved during the whole transformation process.

4. Discussion

With this work, we have demonstrated an approach to identifying a decision point by transforming a BPMN-formalized SOP into FHIR, including FHIR-based patient data. We verified the model using the formalized document ‘SOP Malignes Melanom’. For transformation, we considered all the BPMN-specific data inclusion options provided by OMG, as well as all the BPMN elements defined by OMG as Basic Control Flow Patterns. The FHIR resource ‘PlanDefinition’ has been chosen to rely on an already established standard for modeling clinical pathways. We showed that a loss-free transformation is possible and so created the prerequisites for the identification of decision points using the algorithm.

Despite this success, we also encountered a few limitations. As gateways are typically not represented as direct resources in FHIR, we were not able to directly map ‘merge gateways’ (two tasks pointing to one task after the gateway) in FHIR. Instead, the task preceding the gateway needs to reference all tasks following the gateway, along with additional information indicating whether it is a parallel or an exclusive gateway. If there is a case of two gateways being merged sequentially, the results are a loss of information regarding which gateway was merged first. To address this challenge, we created a subprocess that encompasses all elements from the diverging gateway to the ‘merge gateway’. The preceding element points to the start element of the subprocess, and the subsequent ‘merge gateway’ follows the end event of the subprocess. While this approach increases the size of the model due to the additional subprocesses, it only affects visual representation of the BPMN model. The FHIR code remains unchanged. We developed a different method from Helm et al. [34] to ensure that our models remained as compact as possible, enhancing the readability.

Another limitation is the loss of IDs of the BPMN elements. These exact IDs for sequence flows, Data Associations, and ‘DataStoreReferences’ are lost during transformation into FHIR, while new ones are generated during the transformation back into BPMN. These

not identical IDs do not impact semantics or the transformation process in any way. To address the limitations of the technical information, additional fields of the FHIR resource 'PlanDefinition' could be selected in the future to map this information.

As we used 'DataStoreReferences', including Data Associations, to embed patient-specific data, some manual adjustments to visualizations in Camunda had to be made, because here the linkage between BPMN elements and 'DataStoreReferences' is achieved using Data Input Associations or Data Output Associations.

The generated BPMN models are intended for visual representation in medical contexts, specifically at the point of care. Our transformation approach establishes a framework for navigating to the appropriate section of the BPMN model based on the current patient-specific data, thus presenting only the currently relevant section to the physician. In real clinical settings, our mapping and transformation approach can be integrated into a medical dashboard. In Essen, dermatooncologists will, e.g., prepare tumor conferences with this tool, potentially saving time and enhancing quality.

5. Conclusions

Our transformation approach provides a mapping of BPMN models and FHIR with special emphasis on the semantic integration of patient-specific data, with the purpose of enabling patient-data-specific decision support at the point of care.

To embed patient-specific FHIR data into clinical pathways modeled using BPMN, we developed a method to transform BPMN elements into FHIR fields and vice versa. With an additional framework for navigating to the appropriate section of the BPMN model based on the current patient-specific data, only the currently relevant section can be presented to the physician. Furthermore, this approach offers the possibility to access patient FHIR data at certain decision points and to integrate it into BPMN data elements.

Moving forward, the next step could be to determine the position of a patient in the clinical pathway by a means of integrating patient-specific EHR data stored in FHIR into formalized clinical guidelines. This information can then be visualized for physicians using BPMN, providing a standardized method for clinical decision-making at the point of care.

Overall, our approach offers a promising method for integrating patient FHIR data into BPMN-modeled clinical pathways. Future studies should aim at validating the effectiveness of our approach in clinical practice and evaluating its impact on patient care at the point of care.

Author Contributions: Conceptualization, C.L.B., D.K. and B.B.; methodology, C.L.B. and D.K.; software, C.L.B., D.K. and W.O.I.A.S.; validation, C.L.B., D.K. and W.O.I.A.S.; formal analysis, C.L.B. and D.K.; investigation, C.L.B.; resources, C.L.B.; data curation, C.L.B.; writing—original draft preparation, C.L.B. and D.K.; writing—review and editing, S.N. and B.B.; visualization, C.L.B. and D.K.; supervision, B.B.; project administration, C.L.B. and D.K.; funding acquisition, C.L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by a PhD grant from the DFG Research Training Group 2535, Knowledge- and data-based personalization of medicine at the point of care (WisPerMed), the University of Duisburg-Essen, Germany. We additionally acknowledge the support of the Open Access Publication Fund of the University of Duisburg-Essen.

Data Availability Statement: The data presented in this work are openly accessible from the GitLab of the Medical Informatics program at University of Applied Sciences and Arts Dortmund at [43]. Restrictions apply to the availability of the patient data, which were used under a research agreement for the current work and so are not publicly available.

Acknowledgments: We thank the physicians of the Department of Dermatology (University Hospital Essen, Germany) for their cooperation and the SHIP team (University Hospital Essen, Germany) for data provision and alignment. We also thank Birte Damm, Johannes Wutzkowsky (both Department of Computer Science, University of Applied Sciences and Arts Dortmund (FH Dortmund), Germany), and Matthias Paalik (Department of Computer Science, University of Applied Sciences and Arts Dortmund (FH Dortmund), Germany; and BASF Coatings GmbH, Münster, Germany) for the technical exchange. In addition, DeepL (<https://www.deepl.com/translator> (accessed on 29 November 2023)), ChatGPT (<https://chat.openai.com/> (accessed on 29 November 2023)), and a native English speaker provided assistance in revising the sentences.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Walsh, S.; de Jong, E.E.C.; van Timmeren, J.E.; Ibrahim, A.; Compter, I.; Peerlings, J.; Sanduleanu, S.; Refaee, T.; Keek, S.; Larue, R.T.H.M.; et al. Decision Support Systems in Oncology. *JCO Clin. Cancer Inform.* **2019**, *3*, 1–9. [[CrossRef](#)] [[PubMed](#)]
- Johnson, A.; Zeng, J.; Bailey, A.M.; Holla, V.; Litzenburger, B.; Lara-Guerra, H.; Mills, G.B.; Mendelsohn, J.; Shaw, K.R.; Meric-Bernstam, F. The right drugs at the right time for the right patient: The MD Anderson precision oncology decision support platform. *Drug Discov. Today* **2015**, *20*, 1433–1438. [[CrossRef](#)] [[PubMed](#)]
- Lödel, S.; Ostgathe, C.; Heckel, M.; Oechsle, K.; Gahr, S. Standard Operating Procedures (SOPs) for Palliative Care in German Comprehensive Cancer Centers—An evaluation of the implementation status. *BMC Palliat. Care* **2020**, *19*, 62. [[CrossRef](#)] [[PubMed](#)]
- Object Management Group. Business Process Model and Notation (BPMN). Version 2.0.2. Available online: <https://www.omg.org/spec/BPMN/2.0.2/PDF> (accessed on 23 October 2023).
- Weske, M. *Business Process Management*; Springer: Berlin/Heidelberg, Germany, 2019.
- Martínez-Salvador, B.; Marcos, M.; Palau, P.; Domínguez Mafé, E. A model-driven transformation approach for the modelling of processes in clinical practice guidelines. *Artif. Intell. Med.* **2023**, *137*, 102495. [[CrossRef](#)] [[PubMed](#)]
- Zerbato, F.; Oliboni, B.; Combi, C.; Campos, M.; Juarez, J.M. BPMN-Based Representation and Comparison of Clinical Pathways for Catheter-Related Bloodstream Infections. In Proceedings of the 2015 International Conference on Healthcare Informatics, Dallas, TX, USA, 21–23 October 2015; pp. 346–355. [[CrossRef](#)]
- De Ramón Fernández, A.; Ruiz Fernández, D.; Sabuco García, Y. Business Process Management for optimizing clinical processes: A systematic literature review. *Health Inform. J.* **2020**, *26*, 1305–1320. [[CrossRef](#)] [[PubMed](#)]
- Bender, D.; Sartipi, K. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. In Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, Porto, Portugal, 20–22 June 2013; pp. 326–331. [[CrossRef](#)]
- FHIR Clinical Guidelines. Available online: <http://hl7.org/fhir/uv/cpg/index.html> (accessed on 14 July 2023).
- Lichtner, G.; Alper, B.S.; Jurth, C.; Spies, C.; Boeker, M.; Meerpohl, J.J.; von Dincklage, F. Representation of evidence-based clinical practice guideline recommendations on FHIR. *J. Biomed. Inform.* **2023**, *139*, 104305. [[CrossRef](#)] [[PubMed](#)]
- CODEX+ CELIDA—COVID-19 Clinical Guidelines to Data Mapper. Available online: <https://github.com/CODEX-CELIDA/> (accessed on 17 July 2023).
- Guideline Recommendation Evaluator (GREvaluator). Available online: <https://github.com/CEOsys/grevaluator> (accessed on 17 July 2023).
- HL7 International. Using Resources with Services and Service-Oriented Architecture. Available online: <https://build.fhir.org/services.html> (accessed on 7 September 2023).
- Krauss, O.; Holzer, K.; Schuler, A.; Egelkraut, R.; Franz, B. Challenges and Approaches to Make Multidisciplinary Team Meetings Interoperable—The KIMBo Project. *Stud. Health Technol. Inform.* **2017**, *236*, 63–69. [[CrossRef](#)] [[PubMed](#)]
- Trisotech. How BPM+ Health Complements FHIR. Available online: <https://www.trisotech.com/how-bpm-health-complements-fhir/> (accessed on 7 September 2023).
- Hund, H.; Wettstein, R.; Hampf, C.; Bialke, M.; Kurscheidt, M.; Schweizer, S.T.; Zilske, C.; Mödinger, S.; Fegeler, C. No Transfer Without Validation: A Data Sharing Framework Use Case. *Stud. Health Technol. Inform.* **2023**, *302*, 68–72. [[CrossRef](#)] [[PubMed](#)]
- Hund, H.; Wettstein, R.; Heidt, C.M.; Fegeler, C. HiGHmed Data Sharing Framework (HiGHmed DSF). 2020. Available online: <https://github.com/highmed/highmed-dsf> (accessed on 7 September 2023).
- Hund, H.; Wettstein, R.; Heidt, C.M.; Fegeler, C. Executing Distributed Healthcare and Research Processes—The HiGHmed Data Sharing Framework. *Stud. Health Technol. Inform.* **2021**, *278*, 126–133. [[CrossRef](#)] [[PubMed](#)]
- Wettstein, R.; Hund, H.; Fegeler, C.; Heinze, O. Data Sharing in Distributed Architectures—Concept and Implementation in HiGHmed. *Stud. Health Technol. Inform.* **2021**, *283*, 111–118. [[CrossRef](#)] [[PubMed](#)]
- Wettstein, R.; Hund, H.; Kobylinski, I.; Fegeler, C.; Heinze, O. Feasibility Queries in Distributed Architectures—Concept and Implementation in HiGHmed. *Stud. Health Technol. Inform.* **2021**, *278*, 134–141. [[CrossRef](#)] [[PubMed](#)]
- Wettstein, R.; Kussel, T.; Hund, H.; Fegeler, C.; Dugas, M.; Hamacher, K. Secure Multi-Party Computation Based Distributed Feasibility Queries—A HiGHmed Use Case. *Stud. Health Technol. Inform.* **2022**, *296*, 41–49. [[CrossRef](#)] [[PubMed](#)]

23. Kober, G.; Robaldo, L.; Paschke, A. Using BPMN for medical guidelines that integrate with FHIR-RDF. In Proceedings of the 14th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences (SWAT4HCLS 2023), Basel, Switzerland, 13–16 February 2023; pp. 1–10.
24. Scheuerlein, H.; Rauchfuss, F.; Dittmar, Y.; Molle, R.; Lehmann, T.; Pienkos, N.; Settmacher, U. New methods for clinical pathways-Business Process Modeling Notation (BPMN) and Tangible Business Process Modeling (t.BPM). *Langenbeck's Arch. Surg.* **2012**, *397*, 755–761. [[CrossRef](#)] [[PubMed](#)]
25. Beckmann, C.L.; Lodde, G.; Livingstone, E.; Schadendorf, D.; Böckmann, B. Guideline-Based Context-Sensitive Decision Modeling for Melanoma Patients. *Stud. Health Technol. Inform.* **2022**, *296*, 50–57. [[CrossRef](#)] [[PubMed](#)]
26. Haug, P.; Huff, S.; Bledsoe, J. Diagnosing and Treating Pulmonary Embolus Using HL7 FHIR and the OMG's Business Process Modeling Standards. Available online: <https://www.bpm-plus.org/pdf/intermountain-fhir-bpm-6-25-20.pdf> (accessed on 7 September 2023).
27. Kiourtis, A.; Mavrogiorgou, A.; Kyriazis, D. FHIR Ontology Mapper (FOM): Aggregating Structural and Semantic Similarities of Ontologies towards their Alignment to HL7 FHIR. In Proceedings of the 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), Ostrava, Czech Republic, 17–20 September 2018; pp. 1–7. [[CrossRef](#)]
28. Gißke, C.; Liu, J.; Gand, K. Applying Goal-Oriented Modelling for Machine Learning Based Rehabilitation Care. *Stud. Health Technol. Inform.* **2022**, *294*, 342–346. [[CrossRef](#)] [[PubMed](#)]
29. HAPI FHIR Library. Class Plan Definition. Plan Definition Action Input Component. Available online: <https://hapifhir.io/hapi-fhir/apidocs/hapi-fhir-structures-r5/org/hl7/fhir/r5/model/PlanDefinition.PlanDefinitionActionInputComponent.html> (accessed on 7 September 2023).
30. Health Level Seven International. FHIR Workflow Minutes CC 20201116. Available online: <https://confluence.hl7.org/display/FHIR/FHIR+Workflow+Minutes+CC+20201116> (accessed on 7 September 2023).
31. IBM. Modeling Cancer Treatment Processes with BPMN and HL7 FHIR: A National Comprehensive Cancer Network (NCCN) Case Study. Available online: <https://docplayer.net/64810938-Modeling-cancer-treatment-processes-with-bpmn-and-hl7-fhir.html> (accessed on 23 October 2023).
32. Van der Heijden, J.J.M.J. *Standardized Software Solution for Guidance of Clinical Workflows*; Technische Universiteit Eindhoven: Eindhoven, The Netherlands, 2021. Available online: <https://research.tue.nl/en/publications/standardized-software-solution-for-guidance-of-clinical-workflows> (accessed on 7 September 2023).
33. vCare Project. Available online: <https://vcare-project.eu/> (accessed on 7 September 2023).
34. Helm, E.; Pointner, A.; Krauss, O.; Schuler, A.; Traxler, B.; Arthofer, K.; Halmerbauer, G. FHIR2BPMN: Delivering Actionable Knowledge by Transforming between Clinical Pathways and Executable Models. *Healthc. Future* **2022**, *292*, 9–14. [[CrossRef](#)]
35. Tewes, R.; Matzke, U.C. *Innovative Personalentwicklung im In- und Ausland: Für Einrichtungen im Gesundheitswesen*; Springer: Berlin/Heidelberg, Germany, 2021; p. 294. [[CrossRef](#)]
36. camunda-bpmn-model-7.17.0-alpha3.jar, and camunda-xml-model-7.17.0-alpha3.jar. Available online: <https://jar-download.com/download-handling.php> (accessed on 20 January 2022).
37. hapi-fhir-base-6.2.1.jar. Available online: <https://mvnrepository.com/artifact/ca.uhn.hapi.fhir/hapi-fhir-base> (accessed on 17 January 2023).
38. Hosch, R.; Baldini, G.; Parmar, V.; Borys, K.; Koitka, S.; Engelke, M.; Arzideh, K.; Ulrich, M.; Nensa, F. FHIR-PYrate: A data science friendly Python package to query FHIR servers. *BMC Health Serv. Res.* **2023**, *23*, 734. [[CrossRef](#)] [[PubMed](#)]
39. bpmnlint. Available online: <https://github.com/bpmn-io/bpmnlint> (accessed on 2 June 2023).
40. Russell, N.; ter Hofstede, A.; van der Aalst, W.; Mulyar, N. Workflow Control-Flow Patterns: A Revised View. Available online: <http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf> (accessed on 8 September 2023).
41. Van der Aalst, W.; ter Hofstede, A.; Kiepuszewski, B.; Barros, A.P. Workflow Patterns: Distributed and Parallel Databases. Available online: <http://www.workflowpatterns.com/documentation/documents/wfs-pat-2002.pdf> (accessed on 8 September 2023).
42. HL7 International. CPG Home. Available online: <https://build.fhir.org/ig/HL7/cqf-recommendations/> (accessed on 23 October 2023).
43. BPMN2FHIR: BPMN and FHIR Transformation with Integrated Patient FHIR Data. Available online: <https://git.mi.fh-dortmund.de/guide2treat/bpmn2fhir/> (accessed on 24 October 2023).
44. Camunda Platform 7.15. BPMN 2.0 Specification. Available online: <https://camunda.com/de/download/modeler/> (accessed on 3 August 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.