An Algorithm for Transforming Guidelines from BPMN to Structured Data Analysis (SDA)

Mrs.T.V.Lavanya[1]., J.Akhila[2]

1 Assistant Professor, Department of CSE, Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

2, B.Tech CSE (20RG5A0519),

Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

Abstract :

The clinical procedures described in recommendations may be modelled using the Business Process Modelling and Notation (BPMN), a widely used standard for process modelling. Using a language like Computer-Interpretable Guideline (CIG), computer systems that are based on guidelines must include these clinical activities. However, establishing encoding requirements for a CIG language is a challenging undertaking, often undertaken by technical personnel. In this study, we expand on our prior work describing a transformation mechanism for converting BPMN models into the SDA CIG language. BPMN has the ability to give doctors more agency in the modelling process. An increase in the use of CIG languages, SDA, and others may result from this, especially when combined with the transformation method.

Keywords:

Transformation of clinical guidelines; BPMN; SDA; clinical processes

Introductions

The healthcare industry is one of the most difficult to implement business process modelling (BPM) technology in. 1. The clinical use of BPM is expanding, as shown by recent works2,3. In this context, BPM may provide solutions that aid not just administrative but also medical procedures. BPM's ability to isolate the reasoning behind a process from the specifics of how it's implemented is a major selling point. The Business Process Modelling and Notation (BPMN) is an industry-wide accepted notation for modelling business processes. The fact that it may be used as an instrument and that it is simple to use in models are two of the key reasons for its widespread acceptance. 1. Clinical research utilizing BPMN has shown that non-technical workers can learn the language quickly and intuitively, leading to improved workflows2,3. Despite its apparent ease of use, BPMN is sufficiently formal to serve as the foundation for further implementation. While the BPMN 2.0 specification4 does define certain execution semantics in terms of BPEL, we do not see BPMN as the final destination here; rather, it serves as a starting point for the implementation.

Most often, clinical guidelines take the form of written papers that outline specific therapeutic actions for the diagnosis and treatment of a condition. Any computer system built on recommendations must include the clinical processes described in them, maybe via the use of a CIG language. Although CIG languages may be used in clinical pathways, they were developed with the express purpose of capturing the medical knowledge of clinical recommendations. It is possible to modify and create CIGs visually in the vast majority of languages. Still, it's not easy to do it in a CIG language, which is required for encoding

clinical recommendations. This work makes use of a CIG language for implementing the guidelines specified in BPMN, as well as an algorithm for transforming BPMN guideline models into the CIG language. BPMN might provide subject matter experts the tools they need to tackle the modelling challenge5. Further, the work done to represent the clinical guideline in BPMN may be used for the implementation of the model in other CIG languages, provided that suitable transformation algorithms are created.

Related Papers 2 Graph-oriented languages (such BPMN, XPDL, EPCs), and block-oriented languages, are the two main paradigms for BPM languages, and research on the translation between process modelling languages proposes general transformation methodologies based on this difference. (Such as BPEL and BPML) 6. The so-called structure-identification strategy7, which takes use of the graph-oriented paradigm of the source language, is one of the translation techniques from a graph-oriented language to a block-oriented one. Tasks, gateways, and events are the nodes of the directed graph representing the business process model, and sequence flows are the arcs connecting them. To implement the structure-identification technique, first important structures must be defined in the target language, and then

equivalent structures (components) in the input graph must be identified. The size of the input graph is decreased by mapping each component found in the source graph to the target language and replacing it according to a set of rules. Successful conversions from BPMN to BPEL7 and XPLD to HTP8 have been achieved using this structure identification method. We have successfully used the structure-identification technique to convert BPMN models to Proforma 9 in a previous work5 by adapting it to the field of clinical recommendations.

To produce a semi-automatic translation from BPMN to SDA10, a fundamentally distinct CIG language, we use a similar technique. Proforma's definition of a process is more procedural than SDA's (states-decisions-actions), which describes a process in terms of states and transitions. Clinical recommendations are the focus of both the work of Gonzalez-Ferrer '8, which performs a transformation from XPDL to HTN, and Dominguez '11, which derives Java modules from UML state diagrams. But none of them tackle CIG formalisms. Our publications are, to the best of our knowledge, the only methods ever developed to tackle CIG languages. In BPMN, Clinical Guidelines BPMN is becoming the de facto standard for modelling processes. It offers over 50 modelling components, but research shows that just a common core set and an expanded core set, accounting for roughly 24% of the total, are actually employed by designers most often when creating process models. 12. We have used BPMN to mimic a variety of healthcare procedures. To create them, we've relied mostly on components from the BPMN core and extended core sets, with some help from the 12th specialty set. Start and stop events, tasks, XOR, OR, and AND gateways, sequence flows, conditional and default subprocesses, and loop activities are all examples of BPMN components. Where a process begins and ends is indicated by its corresponding start and end events. Actions are broken down into smaller units called tasks. Additionally, gateways regulate the forking and merging of process flows. Multiple alternative pathways (XOR and OR gates) and parallel control flow topologies are represented by the various gateway types. (AND gateways). In addition, gates may be divided into two types: split and join, reflecting whether or not the process flow is split or joined at the gateway. The steps of a procedure are often linked together using sequence flows. It seems to reason that conditional and/or default sequence flows would come after a split gateway. Each component of a subprocess is represented by its own node in the diagram. Different kinds of activity indicators may be specified in BPMN. We've been using the loop marker, which signals that a task may be repeated, in the context of healthcare standards. In contrast to Ouyang7's work, our focus is on loop activities and subprocesses. The use of subprocesses facilitates the inclusion of potentially complicated processes into clinical recommendations.

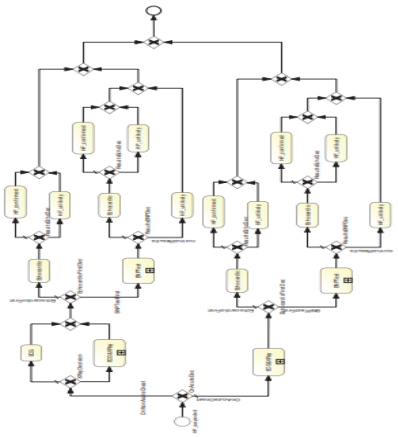

Fig. 1. A BPMN model representing the clinical processes for the diagnosis of chronic heart failure in BPMN.

the simplicity by condensing some elements. Consequently, we work with graphs that have subgraphs. BPMN's loop components are a good fit for describing iterative processes like those that may be found in clinical recommendations. The BPMN representation of a clinical guideline is a directed graph with a single beginning and a single ending. Sequence flows are the connections between tasks, intermediaries, and entryways. Except for gateways, each of these components only has a single sequence flow in either direction. We focus on ordered BPMN diagrams for our modelling needs related to guidelines. Every split gateway must have a matching join gateway of the same kind, and all split-join pairings must be nested appropriately to be considered a structured model. The structured nature of business process graphs is not a drawback, but rather an asset. Figure 1 depicts a business process modelling notation (BPMN) representation of a heart failure diagnostics guideline.

SDA Clinical Practice Guidelines

The SDA10 standard is a formal language for graph-based descriptions of healthcare processes. Its original intent was to reflect the clinical procedures outlined in clinical guidelines, but it has now been expanded to include data mining of the clinical processes recorded in an institution's electronic health records. Concepts of "state," "decision," and "action" are the basis of SDA. Terms representing states, such as "elevated blood pressure," are used to explain the health of a patient, the next step in therapy, or the progression

of a disease. Decision phrases, such as "a BNP test is needed," describe observations that medical practitioners may take into account. Finally, action phrases describe the components of a therapy, and may be either pharmacological or non-pharmacological, such as "make breast X-ray." The above-mentioned terminologies are used by SDA to describe states, actions, and decisions.

A state is defined by its collection of state words, which outline the circumstances that call for a certain response. Similarly, an action is understood to be a collection of action phrases that together stand for the right things to do throughout the medical treatment. Plain connections in the SDA graph may be used to express either temporal or atemporal treatment flows by linking states and actions. Delay intervals, which define the minimum and maximum time between one element in the flow and the next, may be included into connectors. As a final step, choices outline variations in care, such as how to give options for handling risks. Conditioned connections are decision words that specify the circumstances under which a decision must be made. In cases when none of the other options apply, a decision may utilize an otherwise connection. The nodes in an SDA representation of a clinical guideline are states, the edges are actions and choices, and the arcs are connections. There are no hard and fast rules on how the SDA components must be linked together. There are no limits on the number of inputs or outputs on an SDA node. Conditional and otherwise connections can only be used with decisions. Only by clustering the collection of action words that map to the parallel activities in an SDA action can parallelism be represented in control flow structures. Temporal connections or states inserted between each pair of subsequent actions may be used to create sequential control systems. The steps for running an SDA graph are as follows. At first, all states will be able to begin if the existing patient circumstances meet their state's requirements and conditions. If more than one state meets the criteria, the doctor will choose one. Once a state is located whose conditions are not met, the connections will be followed until a delayed connector is reached. As a result, the patient's suggestions will serve as the action words for every step along the route taken. All the outgoing conditioned connections whose decision conditions are met will be eligible to be followed after a decision has been made. The default connection will be used if none of the other requirements are satisfied.

The Evolution from BPMN to SDA

The process revolves on locating and then altering key structures in the input graph. The BPMN model is provided as a structured graph.

The naming of BPMN architectures in SDA

The starting point of this method is an examination of the relevant target language structures. The next step is to locate relevant building blocks, or components, in the source graph. State-based decision-making and action-description (SDA) is made possible. In addition, sequences, non-deterministic flows, and loops may all be described using connections. 10. Concurrencies are represented by AND-gateways, choices by XOR-gateways, non-deterministic actions by OR-gateways, and sequences by sequence flows in the input graph. 4

As we will explain below, we have modified Ouyang7's approach to structure identification. Sequences and parallel components in the input BPMN network are uncovered by the technique. A parallel component typically consists of a split gateway that forks the flow into subcomponents that are later brought back together by a join gateway. And, or, and XOR parallel components are all generated by distinct gateways. The criteria may be labelled on the outgoing arcs of XOR/OR split gateways. The beginning and the end are not included in the sequential or parallel parts. A task or another component may be represented by each node in a parallel component or in a sequence. The whole Figure 1 graph is an XOR-parallel component with two additional XOR-parallel components in its top branch. A new component node is created and used whenever a new component is discovered. It is always possible for structured graphs to be reduced to a single component node; thus, the procedure is repeated until that happens. 6. Use in a Heart Failure Treatment Protocol

Following the ESC recommendation 13, we have implemented the transformation method in a BPMN model for the identification of acute and chronic heart failure. This BPMN architecture, seen in Figure 1, has 23 tasks, 2 subprocesses, 2 ad hoc subprocesses, 10 split XOR-gateways, and 10 join XOR-gateways. The converted SDA model, or transformed SDA, for this BPMN model is shown in Figure 5. A total of 23 states, 14 decision nodes, and 27 actions make up the revised SDA. The converted SDA retains the same structure as the original graph. As a result, most SDA states perform the function of a synchronization hub. This

structural similarity arises because the change is directed.

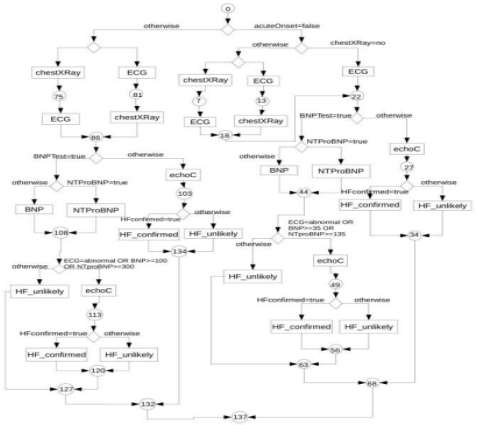

Figure 2. A simplified illustration of the modified SDA model based on the principle presented in Fig. 1. The figure's readability was increased by excluding the state words. by the graph structure, and the fact that they're both graph-oriented languages to boot. We have verified that the SDA model gives recommendations consistent with the clinical guideline by testing it for several groups of patients representing various diagnostic pathways. The more complicated the guideline, the more subprocesses there should be in the BPMN specification. All processes are on the same plane in SDA since there is no way to distinguish between levels of processing. This is not a result of the translation procedure but rather inherent characteristics of the SDA language, yet it might have an adverse effect on the model's readability nevertheless. However, SDA uses numerous phrases within the same action to depict parallelism. This means that parallel processes cannot be represented in any general sense. When comparing the transition from BPMN to Proforma 5, the aforementioned concerns stand out as the most notable variations.

Conclusions

In this study, we discuss the process of translating between two different graph-oriented languages in the context of medical protocols. We have created a technique to translate a BPMN model of a guideline into SDA CIG. The methodology for the transformation is, despite the language barrier, quite comparable to the one we used for the Proforma 5 transformation. This method focuses on extracting relevant structures from the initial graph. Transforming from one language to the other was tested using a clinical guideline on chronic heart failure. Results from the transformation reveal that the generated SDA model has a comparable structure to the BPMN input, while including all components at the same level. Because of this, the resultant SDA graph may be quite large, making it difficult to read. The limitation of SDA to describe parallelism inside an action's numerous periods is another issue. This means that, in general, AND-parallel parts cannot be converted. Despite this caveat, SDA has shown usefulness in a number of therapeutic contexts. Our approaches for transforming data may help make widespread usage of this practice more feasible.

References

1. M. Reichert, What BPM Technology Can Do for Healthcare Process Support, in: M. Peleg, L. N., C. Combi (Eds.), AIME2011, no. 6747 in Lecture Notes in Artificial Intelligence, Springer-Verlag, 2011, pp. 2–13.

2. H. Scheuerlein, F. Rauchfuss, Y. Dittmar, R. Molle, T. Lehmann, N. Pienkos, U. Settmacher, New methods for clinical pathways. Business Process Modeling Notation (BPMN) and Tangible Business Process Modeling (t. BPM), Langenbeck's Archives of Surgery 397 (5) (2012) 755–761.

3. K. Kirchner, C. Malessa, H. Scheuerlein, U. Settmacher, Experience from collaborative modeling of clinical pathways, in: M. Hess, H. Schlieter (Eds.), Modellierung im Gesundheitswesen: Tagungsband des Workshops im Rahmen der Modellierung, 2014, p. 13.

4. L. Dugan, N. Palmer, BPMN 2.0 Handbook, Future Strategies Inc. in association with the Workflow Management Coalition, 2012, Ch. Making a BPMN 2.0 Model Executable, p. 71:92.

5. B. Mart´ınez-Salvador, M. Marcos, A. Sanchez, ´ An algorithm for guideline transformation: from bpmn to proforma, in: Knowledge Representation for Health Care, Springer, 2014, pp. 121–132.

6. J. Mendling, K. B. Lassen, U. Zdun, On the transformation of control flow between block-oriented and graph-oriented process modelling languages, Intl. J. of Business Process Integration and Management 3 (2) (2008) 96–108.

7. C. Ouyang, M. Dumas, W. M. Aalst, A. H. T. Hofstede, J. Mendling, From business process models to process-oriented software systems, ACM Transactions on Software Engineering and Methodology (TOSEM) 19 (1) (2009) 2.

8. A. Gonzalez-Ferrer ´ , J. Fdez-Olivares, L. Castillo, From business process models to hierarchical task network planning domains, Knowl. Eng. Rev. 28 (2) (2013) 175–193.

9. D. R. Sutton, J. Fox, The syntax and semantics of the PROforma guideline modeling language, Journal of the American Medical Informatics Association 10 (5) (2003) 433–443.

10. D. Riano, ˜ The SDA model: A set theory approach, in: Computer-Based Medical Systems, 2007. CBMS'07. 20th IEEE Intl Symp on, IEEE, 2007, pp. 563–568.

11. E. Dom´ınguez, B. Perez, ´ Z. M., Towards a traceable clinical guidelines application. A model-driven approach, Methods Inf Med 49 (6) (2010) 571–580.

12. M. zur Muehle, J. Recker, How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation, in: 20th International Conf on Advanced Information Systems Engineering, LNCS, Springer-Verlag, 2008.

13. J. McMurray, S. Adamopoulos, S. Anker, A. Auricchio, M. Bhm, K. Dickstein, V. Falk, G. Filippatos, C. Fonseca, M. Gomez-Sanchez, T. Jaarsma, L. Kober, G. Lip, A. P. Maggioni, A. Parkhomenko, B. Pieske, B. Popescu, P. Ronnevik, F. Rutten, J. Schwitter, P. Seferovic, J. Stepinska, P. Trindade, A. Voors, F. Zannad, A. Zeiher, ESC Guidelines for the diagnosis and treatment of acute and chronic heart failure, European Heart Journal 33 (2012) 1787–1847