Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 231 (2024) 427–432

Procedia Computer Science

www.elsevier.com/locate/procedia

The 4th International Workshop on Healthcare Open Data, Intelligence and Interoperability (HODII)
November 7-9, 2023, Almaty, Kazakhstan

OpenEHR logic module: a tool for creating decision rules

Beatriz Silva[a], Francini Hak[a], Manuel Santos[a], António Abelha[a,*]

[a]*ALGORITMI/LASI Research Center, University of Minho, Guimarães, Portugal*

Abstract

In clinical decision-making, data are fundamental elements for extracting knowledge for knowledge-based systems. Clinical Decision Support Systems (CDSS) need to ensure that stored data are semantically interoperable to, in turn, represent clinical knowledge and improve the quality of healthcare delivery. The openEHR standard provides a set of specifications that allow health information systems to manage electronic health records in a structured and standardized format. Therefore, this article aims to present a decision support tool capable of creating decision rules, including rules, decision tables, and scores. based on the decision language specifications of the openEHR.

Keywords: Clinical decision support system; decision language; decision logic; electronic health records; openEHR.

1. Introduction

Electronic Health Record (EHR) systems in hospitals, aim to store all different types of patient data, i.e. information about hospital visits, as well as detailed patient records, for instance examination results, symptoms, diagnoses that are made and medication prescriptions [1]. In recent years, Clinical Decision Support Systems (CDSS) are being adopted to extract clinical knowledge from EHR in order to stimulate the behavior of healthcare professionals and promote the collaboration between them, improving the decision-making process [2].

All patient data and other types of information are essential to providing insights to support the decisions made by healthcare professionals. However, knowledge-based systems face some challenges regarding the storage and management of knowledge, as with the large amount of data, it becomes difficult to build a common interpretable structure for different types of systems. This problem translates into the lack of interoperability, that is, the ability of heterogeneous systems to communicate with each other [3].

* Corresponding author. .
E-mail address: abelha@di.uminho.pt

10.1016/j.procs.2023.12.229

To overcome the semantic interoperability problem, the openEHR standard aims to provide specifications that guarantee a standardized structure for EHR [4]. Following a set of components, the Decision Language (DL) is a recent documentation under development that presents a proper format to express rules and guidelines to represent knowledge to clinical decision support [5].

Previous studies allowed us to investigate the DL specification in more detail and test its use through the construction of a logical decision model. Therefore, this article presents a tool designed based on the structure of Decision Logical Modules (DLM), for the construction and management of clinical rules and guidelines. In addition, knowledge representation techniques for rule-based systems were also studied [6], adopting different types of inference as production rules, decision tables, and scores.

This article is structured in four sections. First, an introduction is made. Section two describes standardized languages for decision support. The third section presents the decision logic tool developed, focusing on its features and architecture. Finally, a conclusion is described.

2. Standardized Decision Language

2.1. Arden Syntax

One of the first languages to be built for standards-based CDSSs was Arden Syntax, which emerged in 1992 with the adoption of the American Society for Testing and Materials (ASTM) standard. It now no longer belongs to ASTM, but to the Health Level Seven International (HL7) standard [7]. The use of this language is intended to help healthcare professionals represent clinical knowledge. As such, knowledge representation is organized through groups of modules referred to as Medical Logic Modules (MLMs) [8]. The modules must follow the structure of the language in order to be able to process all the clinical knowledge, as well as the logic present in the system. By following this structure, it is possible to separate the clinical knowledge from the logic, for example by using variables, and consequently promote the transparency of the MLM [9].

2.2. Guildeline Interchange Format

The InterMed Collaboratory organization developed a project on a language for representing guidelines called Guideline Interchange Format (GLIF). This project involved members of Harvard, Columbia and Standford Universities, where they were biomedical informaticists. Collaboration is the main objective of this organization, so they seek to promote the sharing of a structure for representing guidelines in the different software that healthcare institutions contain, which allows them to manipulate, analyse and process guidelines [10].

GLIF is currently in its third version. Compared to previous versions, it has been more structured in terms of its representation, with the implementation of object-oriented expressions and query languages. Flow control has been improved through better class management, i.e. revision and addition of classes that were missing in the previous version. With regard to the integration of GLIF3 with clinical information systems, a layered model was built so that patient data and clinical knowledge could be visualized more clearly when processing guidelines. In this version, open standards were implemented as frameworks to develop syntax for the guidelines and the HL7 standard was used to integrate medical terminology into their structures [11].

2.3. Decision Language

Initially, openEHR created a decision language to support clinical guidelines called Guideline Definition Language (GDL). In order to add more functionalities, a second version of GDL was also developed, aiming to improve the previous version. GDL2 is already implemented in some health information systems to support clinical decisions [12]. Of late, a need was identified to support workflows and simplify the syntax of the language, and, in consequence, the decision language (DL) specification was presented. The DL proposes a sectioned structure capable of building Decision Logic Modules (DLM) for the representation of clinical guidelines and rules. In this sense, DLM is used in EHR systems, as a way of assisting plans, guidelines and data entered through rules. The DLM language contains several attributes [5]:

- **Reference data**: attributes related to medical knowledge;
- **Input variables**: patient data;
- **Conditions**: patient data and have a Boolean value;
- **Rules**: set of conditions that give rise to one or more actions;
- **Rule set**: the formation of several rules with different conditions and actions;
- **Results**: outputs resulting from the execution of the rules.

DL aims to build decision logic by using functions to be perceived as rules, conditions, decision tables, scores, and whatever else is necessary. Functions are the crux of this language, as this format makes it possible to translate clinical rules in a way that is intuitive and easy for healthcare professionals to understand. In addition to the DLM, the DL is also used in the Subject Data Proxy (SDP), in which case it is not related to logical decisions, but involves the formulation of variables that involve states or events, and these are consumed by the DLM [13]. The choice of decision language was based on the criterion of a language that supported decisions related to workflows. In addition, it was also important that the decision language delved into other types of decision models so that it wasn't just restricted to creating rules. All these criteria contribute to the choice of the DLM language.

3. Decision Logic Tool

3.1. Architecture

Based on the need to achieve a flexible and interoperable decision support mechanism, a decision logic tool is proposed based on the structure of the openEHR decision language, allowing the creation of clinical guidelines and rules. Figure 3 demonstrates the architecture designed for the proposed system, which comprises a tool for creating logical decision modules and an inference engine.

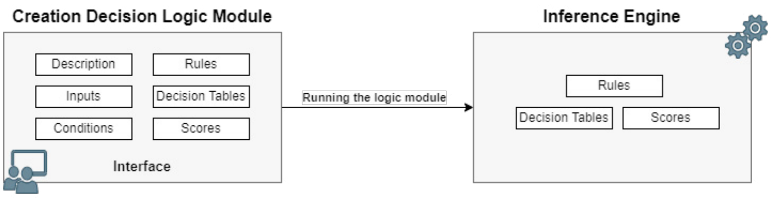


Fig. 1. Architecture.

The creation of the decision logic modules is a web interface developed to be simple and user-friendly, ensuring ease of interaction. The solution includes predefined sections, namely ”Description”, ”Inputs”, ”Conditions”, ”Rules”, ”Decision tables” and ”Scores”. These sections were defined based on the structure specified in the openEHR decision language specification and also by needs identified in the context of use. When the decision module is created, it will be used by the inference engine that will process the rules defined in response to decisions requested by other services.

3.2. Decision creation flow

To represent and visualize the entire process adopted in the tool, figure 2 shows a detailed flow diagram with a sequence of actions. The purpose of this diagram is to distill and generalize the diverse features of the tool, fostering a more profound and nuanced understanding and enhancing overall comprehension of its capabilities.

The process initializes by filling in the fields in the ”Description” section, which include details about the model (type, status, purpose, dates of its creation), as well as information about its author. The second step is to create the input variables. These variables can into different types, as ”input”, ”rule”, ”field”, ”subject proxy”, and ”calculation”. The diverse variable types enable the tool to create both complex variables, like rule variables with specific conditions, and simpler ones that only require the input of a name. After that, there are two options: create more input variables again or proceed to the next stage of the flow. Once the input variables have been defined, the conditions can be

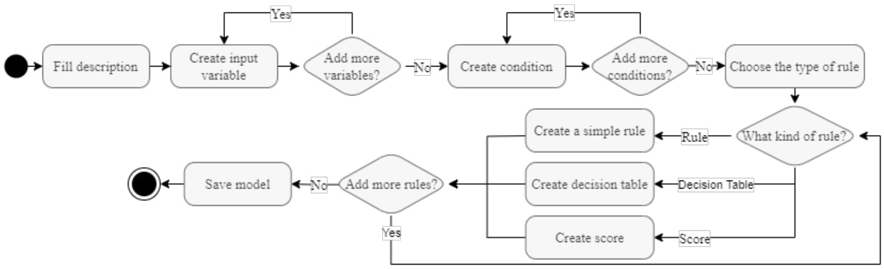


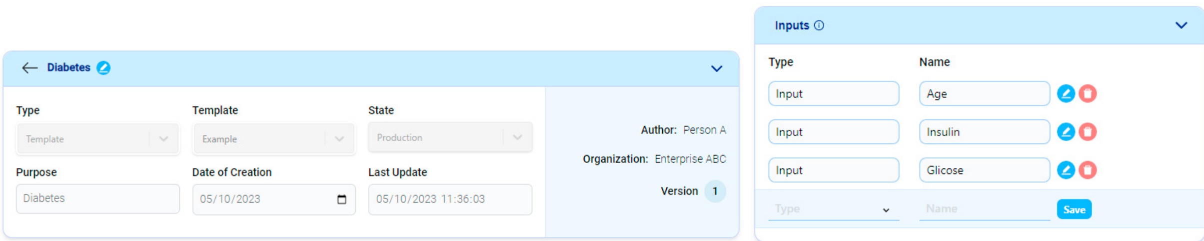
Fig. 2. Flow Diagram.

created, as these require a input variable to be joined with an operator and a value. The definition of the conditions in the model are important for the next functionalities.

Once that step has been completed, the process allows for the creation of more conditions or the selection of the type of rule to be created. There are three decision logic models: simple rule statements, decision table, and score. Each model has a different way of knowledge representation, leading to a clear distinction in their respective workflows. Simple rule statements operates on IF-THEN logic, decision table involves a set of rules with varied outputs, and score combines criteria with assigned points, and totals them to produce an output. The process can be expanded by incorporating additional rules or proceed with model validation. After having all the fields created, the tool system is responsible for validating the model, checking whether there are errors in the data entered or missing. Once validated, the model is stored in a standardized form and can be readily integrated and interoperable with the inference engine and other services.

3.3. Implementation

In this section, the tool will be explored through clinical examples about diabetes. It should be noted that these are fictitious examples and were not applied in a real context. All the essential fields about the module are easily identified and understood regarding to its theme and creation details. This tool is also responsible for the management of decision modules in terms of status and versioning. Figure 3 visually represents the initial components of the tool, namely the 'Description' and 'Input' sections.



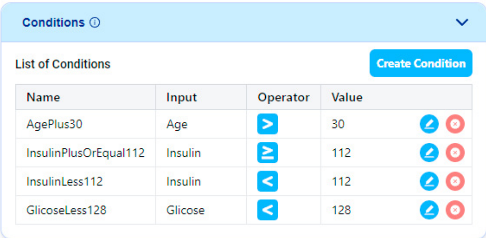
(a) Description Section. (b) Input Section.

Fig. 3. Example of tool Description and Input sections.

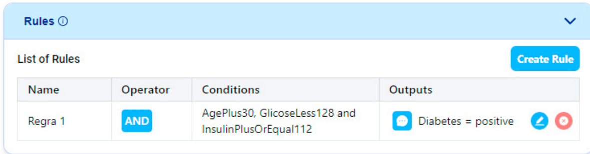
In the description section (figure 3a), it is necessary to indicate the type of module to be created, which can be a template or workflow, and, in turn, associate the previously created template or task plan. The clinical purpose of the decision module must also be indicated. The creation date, update date, status, version, and author information will be automatically recorded by the system. Figure 3b presents the input section that allows the creation of variables to be used in accordance with the rules. Variables have a name, a type, and a possible value that may or may not be associated.

The input variables will provide a starting point for creating the conditions. Therefore, a condition is structured by combining a variable, an operator, and a specified value, identified by an intuitive name. The conditions created are represented in a list, as shown in figure 4a, which can be edited or deleted. The main effort was to associate the operators with symbols for better understanding, i.e., ”equal” or ”not equal”, contributing to the logical connectives that define the relationships between these variables and values.

After defining conditions, the next section is responsible for creating decision rules related to the inference model, which may be production rules, decision tables or scores. Figure 4b) represents the list of rules that are composed of one or more conditions, having a logical operator (AND or ANY) to relate the link between them. If the conditions are truthful, an output is triggered, which can be of the event or message type.



(a) Conditions Section.

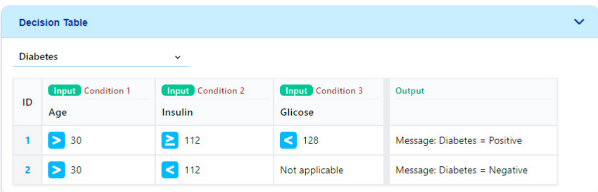


(b) Rules Section.

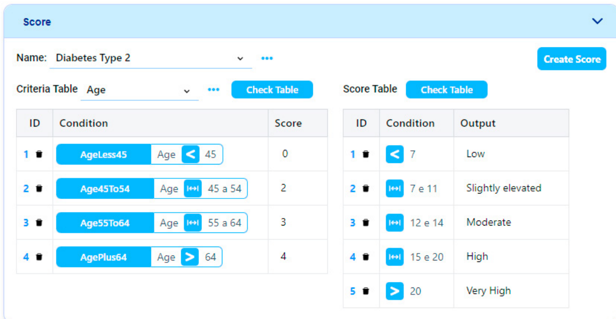
Fig. 4. Example of tool Conditions and Rules sections.

In the case of decision table (Figure 5a), the goal was to create a decision table that was 100% personalized, i.e. the entire process of creating the set of rules, as well as the associated outputs, are all selected in the table itself. There is a division in the table where it is clear that there are conditions on the left and the outputs on the right. Columns can be related with a variable or a rule type.

Scores are a point-oriented decision model. This model contains two types of tables, criteria and scales (Figure 5b). Each criteria table is linked to a variable and its conditions are associated with a value point. In the end, all the points are added up and the result can be seen in the scales table, which contains various conditions in relation to the final score. In the tool is possible to create various criteria tables and then access them via a dropdown. Figure **??** presents a table of criteria for the type 2 diabetes risk assessment score for the variable “Age”, however, there are other variables associated with the criteria tables for this case, such as body mass index, waist circumference, among others.



(a) Decision Table Section.



(b) Score Section.

Fig. 5. Example of tool Decision Table and Score sections.

4. Conclusion

This short paper presents a decision tool developed to create decision rules and guidelines for healthcare. The knowledge representation was based on the openEHR decision language specification that provides a standardized structure in the decision support.

One of the benefits offered by the proposed tool is the autonomy and flexibility it provides for healthcare professionals. This capability allows them to automatically represent rules or guidelines related to forms or clinical pathways, without relying on an Information Technology (IT) expert to make modifications in the EHR system itself. Besides, the usage of the tool will also allow healthcare professionals to reduce time and erros on the decision-making process, as well as promoting a collaborative environment where healthcare professionals can reuse guidelines that have been created by others. One of the challenges faced during the course of this study was the incomplete nature of the DL specification. Consequently, the authors had to address this issue by supplementing the lacking information based on our perspective and requirements. In doing so, we perceive this accomplishment as a substantial contribution to this research.

As a part of our forthcoming efforts, we intend to implement the tool in real contexts conducted in various case studies to verify its effectiveness and guarantee its suitability for all environments. Moreover, we plan to integrate decision trees as a visualization approach, allowing the transformation of decision models into a tree structure.

Acknowledgements

This research was funded by Fundação para a Ciência e Tecnologia, within the Project Scope: UIDB/00319/2020. Francini Hak thanks the Fundação para a Tecnologia (FCT) for the grant 2021.06230.BD.

References

[1] Lino Murali, G. Gopakumar, Daleesha M. Viswanathan, and Prema Nedungadi. “Towards electronic health record-based medical knowledge graph construction, completion, and applications: A literature study”. In: *Journal of Biomedical Informatics* 143 (2023), p. 104403. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2023.104403. URL: https://www.sciencedirect.com/science/article/pii/S1532046423001247.

[2] Robert Rudin, Mark Friedberg, Paul Shekelle, Neel Shah, and David Bates. “Getting Value From Electronic Health Records: Research Needed to Improve Practice”. In: *Annals of internal medicine* 172 (June 2020), S130–S136. DOI: 10.7326/M19-0878.

[3] Reed T. Sutton, David Pincock, Daniel C. Baumgart, Daniel C. Sadowski, Richard N. Fedorak, and Karen I. Kroeker. “An overview of clinical decision support systems: benefits, risks, and strategies for success”. In: *npj Digital Medicine* 3 (1 2020), p. 17. ISSN: 2398-6352. DOI: 10.1038/s41746-020-0221-y. URL: https://doi.org/10.1038/s41746-020-0221-y.

[4] Cecilia Mascia, Paolo Uva, Simone Leo, and Gianluigi Zanetti. “OpenEHR modeling for genomics in clinical practice”. In: *International Journal of Medical Informatics* 120 (2018), pp. 147–156. ISSN: 1386-5056. DOI: https://doi.org/10.1016/j.ijmedinf.2018.10.007. URL: https://www.sciencedirect.com/science/article/pii/S1386505618303502.

[5] Thomas Beale. *Decision Language specification*. 2021. URL: https://specifications.openehr.org/releases/PROC/Release-1.6.0/decision_language.html.

[6] Beatriz Silva, Francini Hak, Tiago Guimaraes, Maria Manuel, and Manuel Filipe Santos. “Rule-based System for Effective Clinical Decision Support”. In: *Procedia Computer Science* 220 (Jan. 2023), pp. 880–885. ISSN: 1877-0509. DOI: 10.1016/J.PROCS.2023.03.119.

[7] Klaus Peter Adlassnig, Peter Haug, and Robert A. Jenders. “Arden Syntax: Then, now, and in the future”. In: *Artificial Intelligence in Medicine* 92 (Nov. 2018), pp. 1–6. ISSN: 0933-3657. DOI: 10.1016/J.ARTMED.2018.09.001.

[8] Howard R. Strasberg, Bryn Rhodes, Guilherme Del Fiol, Robert A. Jenders, Peter J. Haug, and Kensaku Kawamoto. “Contemporary clinical decision support standards using Health Level Seven International Fast Healthcare Interoperability Resources”. In: *Journal of the American Medical Informatics Association : JAMIA* 28 (8 Aug. 2021), p. 1796. ISSN: 1527974X. DOI: 10.1093/JAMIA/OCAB070. URL: /pmc/articles/PMC8324242/%20/pmc/articles/PMC8324242/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8324242/.

[9] Matthias Samwald, Karsten Fehre, Jeroen de Bruin, and Klaus Peter Adlassnig. “The Arden Syntax standard for clinical decision support: Experiences and directions”. In: *Journal of Biomedical Informatics* 45 (4 Aug. 2012), pp. 711–718. ISSN: 1532-0464. DOI: 10.1016/J.JBI.2012.02.001.

[10] Lucila Ohno-Machado, Nilesh Jain, D. Sc, Samson Tu, Diane Oliver, Robert Greenes, Edward Shortliffe, and G. Barnett. “The GuideLine Interchange Format: A Model for Representing Guidelines”. In: *Journal of the American Medical Informatics Association* 5 (Mar. 1998).

[11] Aziz A. Boxwala, Mor Peleg, Samson Tu, Omolola Ogunyemi, Qing T. Zeng, Dongwen Wang, Vimla L. Patel, Robert A. Greenes, and Edward H. Shortliffe. “GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines”. In: *Journal of Biomedical Informatics* 37 (3 June 2004), pp. 147–161. ISSN: 1532-0464. DOI: 10.1016/J.JBI.2004.04.002.

[12] Thomas Beale. *Guideline Definition Language v2 (GDL2)*. 2019. URL: https://specifications.openehr.org/releases/CDS/Release-2.0.0/GDL2.html.

[13] Thomas Beale. *Decision Language (DL) and Model*. 2020. URL: https://specifications-test.openehr.org/releases/PROC/Release-1.5.0/decision_language.html.