MedPath

Trajano, Iago Avelino; Ferreira Filho, João Bosco; de Carvalho Sousa, Flávio Rubens; Litchfield, Ian; Weber, Philip

DOI:
10.1016/j.ijmedinf.2020.104328

MedPath:A process-based modeling language for designing care pathways

Iago Avelino Trajano, João Bosco Ferreira Filho[1]

[1]Federal University of Ceará

April 9, 2019

Abstract

In medical environments, care pathways represent a protocol that details the essential and most common conducts for the specific treatment to be followed by medical professionals during patient treatment. Care pathways have existed for a few decades now, but although formalization has become more frequent, their execution is still very intuitive and can sometimes be very complex and involve many steps and variables. Besides that, mechanisms for execution analysis and auditing are still not common practices among hospital environments. This study proposes a process-based approach to modeling care pathways by designing a Domain Specific Language capable of building and reading care pathways that can also be integrated with other software solution for pathway execution analysis and auditing.

Introduction

Medical care pathways mainly concern human life and patient welfare, thus being extremely sensitive in time and accuracy, especially in hospital environments where the patient's condition typically requires immediate care. The process takes into account several variables other than the patient's currently displayed symptoms, e.g. possible medication allergies and previous medical history, in order to avoid surprises and further complications in their medical condition.

Hospitals maintain systems called Health Information Systems (HIS) [1] for patient record management that aid in the promptness of the care provided for them and keep track of every procedure performed in the treatment so as to oversee the treatment process more rigorously. Nonetheless, most HIS are focused on organizing, executing and registering patient treatment, leaving the care pathway application almost entirely to medical professional's knowledge and Clinical Practice Guidelines released by local councils [2].

While executing the care pathway is crucial in the treatment process, keeping track of what happens during the treatment is also very important for observing and adjusting common practices in order to improve the quality of the service.

This study proposes a model-based definition for clinical pathways, applying the concepts of Model-Driven Engineering [3] to provide a Domain Specific Language [4] (DSL) specialized in the medical conduct capable of modeling care pathways.

Problem Statement

Patient treatment processes are extremely sensitive, as they involve costly and sensitive procedures, as well as the patient's health and welfare, which makes it important for them to be as thorough, fast and accurate as possible to avoid medical errors and extra costs for the hospital and patients. Auditing is an useful tool to identify possible deficiencies in medical conduct and perform progressive improvements whenever necessary. It is, however, difficult to audit and enforce the medical conduct without proper documentation and some level of standardization to the process. Care pathways are aimed at organizing, structuring and standardizing medical treatment in order to solve these problems.

Our strategy for making the pathway-building process more organized and efficient is to delimit the scope of the problem and develop a representation based on well-defined elements of the domain called a Domain Specific Language [4]. Building a care pathway is simpler and more organized when the individual can describe the actions by picking from a pool of well-defined elements that have been previously laid out.

This definition of the pathway-building problem as a process-based model can be achieved with the aid of many representation technologies, such as the Universal Modeling Language (UML) [5] as suggested in [6] and proposed in [7], [8] and [9] or Business Process Model and Notation (BPMN) [10] as proposed in [11] and [12].

We asked ourselves a question: *Why should we implement a DSL rather than just formalize a model using one of the aforementioned notations?* A DSL allows for care pathways to be expressed through a "*language*" with a higher level of abstraction of the care pathway domain, allowing for domain experts to create and modify care pathways directly when properly trained[4]. They also enhance the application's reliability and optimizability [13], maintainability and portability as well as conservation and reuse of the the knowledge as the language embodies the domain knowledge [14]. The use of a DSL can also reduce clutter from the very verbose process-based notations (e.g. concepts of events, activities and gateways) when designing a pathway, making it simpler for professionals whose expertise scope does not involve such concepts and notations.

Unlike simpler notations, the DSL is a concrete model that can be implemented as the support structure for software aimed at modeling and executing care pathways.

This study proposes a methodology to organize and optimize the patient care process based on the concept of care pathways and the paradigms of Model-Driven Engineering (MDE). We introduce an artifact called a *metamodel* which may be used to model care pathways (metamodel instances) as well as aid in their execution in computerized systems and supply datasets that may later be mined for statistics and auditing.

Requirement Analysis

Care pathways are essentially business process models with specific knowledge of the medical field that describe the progress of a patient's treatment throughout their stay in medical environments. [15]. In order to build an efficient DSL for the design and execution of care pathways, the following set of requirements has been derived from a thorough analysis of the possible scenarios when performing care pathways in clinical environments.

R1*: Represent the basic process concepts that are commonly used in care pathways (such as step, transition, decision, flow and variables).*

The language must contain elements that enable the definition of the overall structure of a care pathway, from the creation of isolated conducts (*steps*) to the establishment of communications between each of those conducts (*sequences*), defining of a sense of flow between the steps from the beginning to the end of the treatment.

R2*: Contain the most common medical concepts used in care pathways, as well as be able to accommodate any kind of medical conduct through the use its elements.*

The language must be able to express the most common medical conducts performed during patient care. For example, the creation of on-site treatment processes for administering medication, requesting examinations, patient admission for hospital treatment, etc. Administrative functions must also be taken into account, such as scheduling hospital and clinical visits and referring patients to specialists if necessary.

R3: *Establish relationships between the steps and give the pathway tools to interpret decision-making.*

The language must represent the connections between each step of the process, creating a notion of flow between the elements, so that it can construct a path that leads from beginning to end of the treatment.

Materials and Methods

This study proposes a solution based on two core concepts: **care pathways** and **Model-Driven Engineering (MDE)**. We will use the next two subsections to delve a little deeper into these concepts for a better understanding of the proposed solution.

Integrated Care Pathways

Integrated care pathways are care plans that details all the essential steps for treating patients for a specific clinical problem [16, 17]. Their main objectives are to to, 1) establish a more accurate and generalized treatment for a specific condition and, 2) improve0 the speed and quality of the treatment. They also allow the medical team to keep record of the steps, which ensure more efficient auditing of their structure and execution, improves the identification and reporting of medical errors and misconducts [18] and boosts safety organization processes in hospital environments.

The concept of care pathways emerged in industrial processes during the 1980s in the United States and was largely used ever since. By the late 1990s, care pathways had been incorporated to over 80% of US hospitals [19] and several other countries, including over 23 European countries [20], becoming a concrete reality in medical practices to the present day.

The implementation of clinical care pathways has been proven to increase treatment efficiency [21, 22], reduce hospital costs and patient length of stay [23] and improve teamwork among the medical professionals that execute them [24, 25].

Model-Driven Engineering

Model-Driven Engineering (MDE) is a software development methodology proposed in the 2000s by the Object Management Group (OMG) [26] that represents a broader vision of the concept of Model-Driven Architecture (MDA). A model-driven engineering process consists on developing domain models that encapsulate all aspects of a specific problem, creating an abstract representation of that problem. A Domain Specific Language (DSL) is a programming or executable language that offers functionalities aimed at a specific problem[4, 27].

The core idea of MDE is that it considers models first-class entities and every artifact in a software as a model element [3]. The metamodel is the product of mapping the problem's artifacts into entities. It consists of a graph-like structure that represents most (or all) of the aspects of the problem's definition. Once the metamodel is complete, it may be used to generate instances called models which, in this particular case, represent care pathways.

Related Work

There are currently some published studies on defining the scope of a care pathway into different notations such as UML [7, 8, 9], BPMN [11, 12] and other modeling languages that can map a problem into process-based structures.

In [7] we see an UML-modeling framework aligned with Petri Nets [28] for simulation of the system's workflow and evaluation of its efficiency. The study by [8] proposes a web framework based on UML modeling with OWL-S ontologies for care pathway modeling. As for the BPMN studies, [11] proposes the manual modeling of two care pathways based on BPMN by a multidisciplinary team of computer scientists and medical professionals. The study focused on whether or not it was possible to model care pathways using BPMN notations, while [12] provides a tool for modeling care pathways as a BPMN extension and an ontology that can be used as guidelines for care pathway modeling as BPMNs.

The aim of the present study, however, is building an understandable DSL that enables modeling, execution and integration of the modeled care pathways. It aims to allow professionals without a deep understanding of process-based knowledge to model and execute care pathways based on their field expertise. It also allows for a smoother integration of different software solutions that employ the proposed DSL.

MedPath

We have designed MedPath with the support of Java Eclipse Modeling Framework (EMF) [29]. The coding and design of the DSL[1] was carried out on the Obeo Designer framework [30], which provides an environment to configure numerous representations of viewpoints, allowing for graphical modeling and concrete syntax creation[31].

The abstract model was developed using Obeo Designer's interface for Ecore project modeling. It provides a set of tools and visual aid for creating elements with a drag-and-drop toolbox and pop-up boxes for element configuration [31], which allows for a smoother, non-textual modeling process. The code for all the classes drawn and defined in the metamodel can be automatically generated by Obeo.

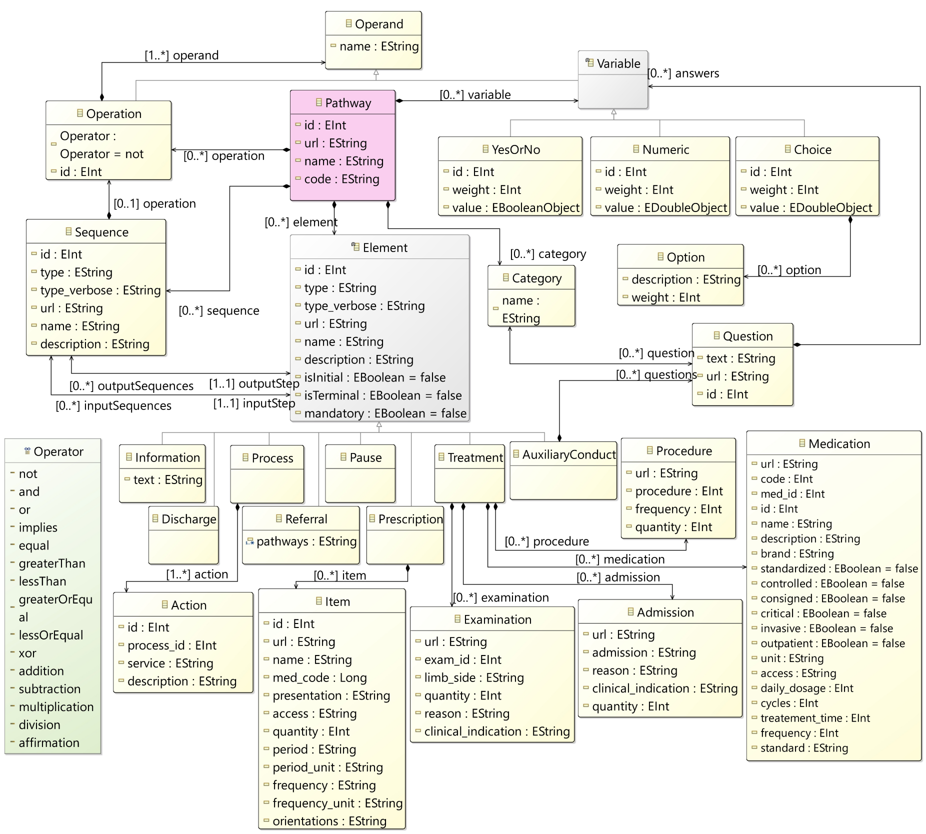


Figure 1: The general structure of a pathway metamodel.

[1]The complete DSL modeling project code can be found at: https://github.com/CarePathwayModeler/pathwayMetamodel.git

Abstract Syntax

The complete structure of MedPath's abstract syntax can be found in Figure 1. The language defines the care pathway as a set of Elements called a ***Pathway***. Elements represent either medical conduct actions or auxiliary actions necessary to advance with the pathway. The types *Element*, *Sequence*, *Operation*, *Variable* and *Category* stem directly from a Pathway element.

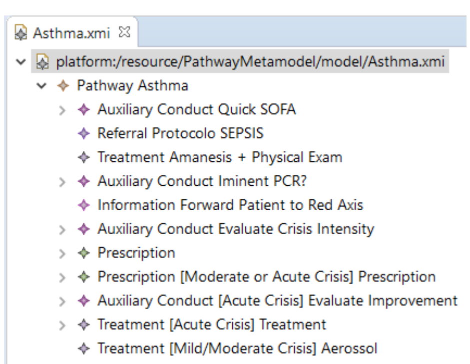
The Element supertype contains the basic information necessary to conduct-defining elements, namely Information, Discharge, Pause, Referral, Process, Prescription, Treatment and Auxiliary Conduct. Each of those contains variables that store data that in such a way that they represent medical actions taken during the treatment process. As seen in Figure 1, a Prescription element is a set of Medication items, the Treatment element represents a set of Medications, Procedures, Examinations and medical Admissions. Auxiliary Conducts are sets of Questions which, in turn, contain Variables that store their answers.

Category, Sequence, Operation and Variable elements represent auxiliary mechanisms that ensure the pathway flows correctly. Category elements are associated with Questions from Auxiliary Conducts. Sequences are associated with Elements and Operations. Variables store values and can either stem from operations or directly from the Pathway element.

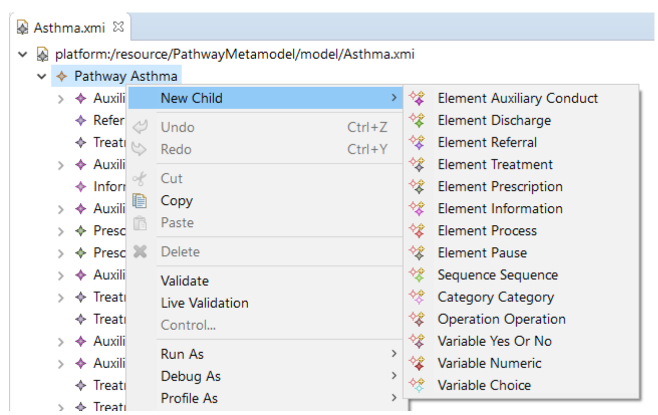
The interactions between these elements will be further explored in the *Semantics* section of this paper.

Concrete Syntax

The concrete syntax for MedPath can be defined in two different ways: textually or visually. The default concrete syntax generated by Obeo Designer is mostly textual, with some basic visual aid. It allows a user to create care pathways either through Java commands by using the rules defined by the metamodel (feeding the metamodel to a Java project and using it as a custom library) or by using Obeo's default interface that is still mostly textual (Figures 2a and 2b).



(a)



(b)

Figure 2: The

We have developed a tool for the visual representation of MedPath's concrete syntax[2] with the aid of an open source software project called Sirius [32, 33], which supports the creation of graphical modeling workbenches for projects. This tool can be used to create new care pathways with the aid of visual elements (rather than a textual approach) [34], as well as to generate visual representations for previously modeled care pathways.

[2]The concrete syntax designer model can be found at: https://github.com/CarePathwayModeler/carePathwayDesigner.git

The Sirius designer project allows us to read an instance of any DSL and configure a *Designer* element that displays the metamodel instance based on a previously defined set of rules

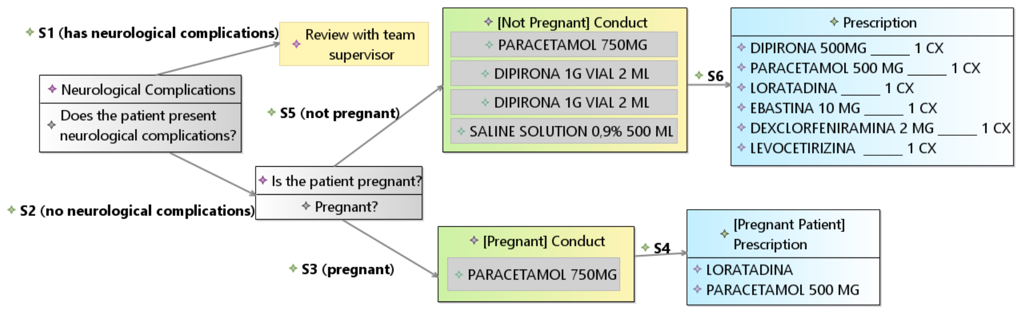


Figure 3: The visual representation of a care pathway

Figure 3 displays an example of a pathway for patients with Zika Virus modeled with MedPath and translated into a viewable structure by the concrete syntax. The treatment consists in verifying if the patient displays neurological complications and/or is a pregnant female and then defininf which treatment that patient must receive.

Semantics

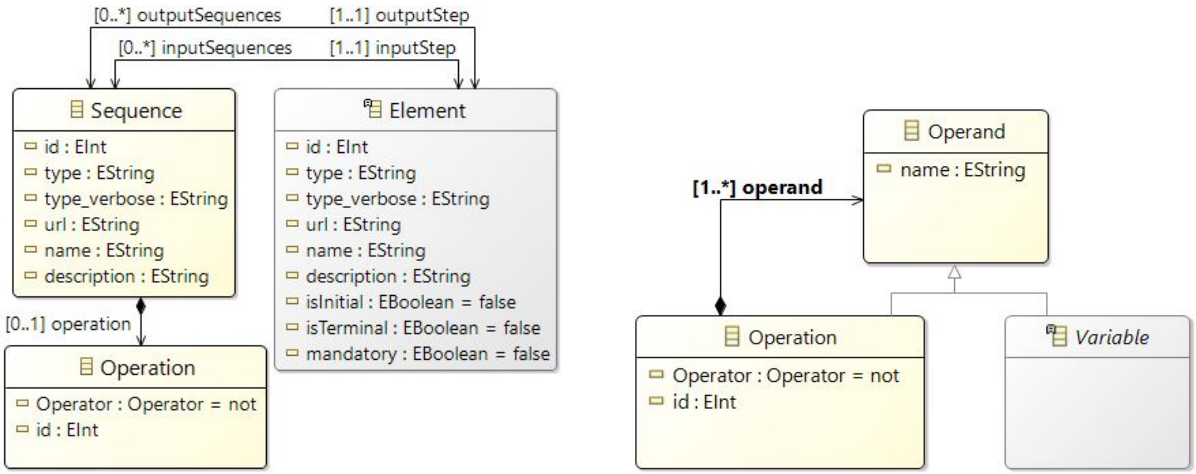
The elements of our DSL can be divided into two categories: the ones that represent common medical conducts (treatments, prescriptions, examinations, etc) and auxiliary elements that are necessary to organize and control pathway flow, store and evaluate information, navigate between pathways and other vital background functions.

The care pathway can be represented as a graph-like structure where the nodes describe steps of the treatment and the edges (which we will call *Sequences*) represent transitions between two different steps.

A *Pathway* element represents the care pathway as a whole. It is the first item to be created in a care pathway and consists of a collection of all the different items that can be found in a pathway (refer back to Figure 1).

The super-type *Element* contains basic attributes that are common to most medical conduct elements. All the elements that represent types of medical conduct inherit its attributes. The elements that stem off of this super-type can be described as follows:

- **Auxiliary Conduct**: this element consists of a set of questions (Figure 1) that must be asked to the patient during the treatment. Each question is defined in another auxiliary element called *Question* which contains the text of the question and a *Variable* element for storing the answer.
- **Information**: this is a simple informative step that may contain instructions for the patients, guidelines for the doctors (such as handing out informative brochures) (see Figure 1) or any other relevant information that has to be delivered during the treatment.
- **Treatment**: this element is the step that defines on site treatment for patients that require immediate care. A treatment may contain a set of other elements that constitute the medical conduct for *in loco* treatment, (Figure 5) such as:
  - *Medications*: prescribe medication to be administered during hospital treatment;
  - *Procedures* and *Examinations*: prescribe medical procedures and/or examinations to be performed on the patient in the hospital;



(a) The Sequence element and its related elements

(b) The relationships of the Operation, Operand and Variable elements.

Figure 4: The Sequence and Operation elements.

  - *Admissions*: refer patient for local hospitalization should the caring professional deem it necessary;

- **Prescription**: this is the element used for prescribing medication for the patient to take at home. A prescription contains a collection of another core element called *Medication* in which the medication is properly specified, along with the guidelines on how and when to take it.

- **Discharge**: the element that defines that a patient can be formally discharged and released from inpatient treatment.

- **Process**: a support element for executing auxiliary actions such as scheduling return appointments, classifying patient risk and referring the patient to specific medical specialties. These actions are registered in **Action** elements that are contained within a Process element.

- **Referral**: an element that directs the user towards another care pathway that may be better suited for the patient's condition;

A pathway also contains elements that execute auxiliary functions like storing patient condition data, composing the logic operations that help control the action flow in the pathway, transitioning the action flow from one step to another among other functions. The language contains the following auxiliary elements:

- The **Sequence** (Figure 4a) element is one of the most important among the auxiliary elements, as it is the element that links two steps together. A sequence can be unconditional, meaning the flow will always take that path once it comes to the output step, or contain what we call a *flow condition*, which consists of a logic expression that defines whether or not the action flow will take that path. An element may be connected to more than one input/output sequences. A sequence, however, can only be linked to two steps: the output step (the one from which the sequence originates) and the input step (the one which the sequence points to).

  - The **Variable** element (see Figure 1 for Variable's general structure along with its sub-elements) is present in two different contexts in this metamodel. Globally, contained in the pathway itself, or locally, as a child to Question elements. They can contain either Logic or Numeric values, and are used to either structure the expressions contained in Operation elements or save the data provided by the patient in the Auxiliary Conduct elements.

  - The flow condition of the sequence is expressed by an **Operation** element. It defines logical or

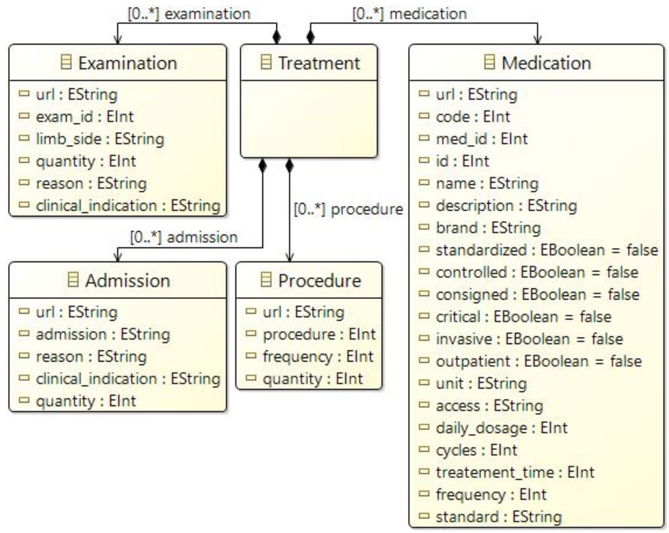


Figure 5: The Treatment element and its sub-elements.

arithmetic operations to aid the sequences in defining which path to take, and may be one of the most complex auxiliary elements in the DSL. Operations contains an operator (Figure 4b) that defines what the operation does, and a list of *Operands* that can be either Variables or other Operations (which allows for the use of nested operations for the evaluation of more complex equations). Just like variables, Operations can be global or local.

Category: element used to define labels for the Question elements. It is a non-essential auxiliary element used to group and organize questions in Auxiliary Conducts.

Evaluation

As this study proposes a model to be used in medical environments (both hospital and clinical), the evaluation strategies employed to measure its efficiency are focused on the employment of the model in practical use and the satisfaction obtained in its employment for care pathway development.

The Environment

The care pathway modeling effort was carried out by a team of 6 medical professionals and 4 modeling users. The medical professionals are responsible for defining and revising the care pathways and the modelers handle the modeling process of the care pathways with MedPath, acting as an interface between the medical professionals and the language itself (effectively, the end users of this language).

MedPath has been used to model care pathways that are in use in 39 medical units in Brazil and an estimated 1.680.000 medical services have been launched using the language since October 2017. The number of individual care pathway executions may be larger than 1.68 million, as one service may trigger more than one pathway throughout its duration. A total of 85 care pathways[3] have been modeled with MedPath and are currently employed in the patient care process.

We have established simple metrics for care pathway complexity assessment as displayed on Table 1.

[3] The pathways can be found at https://github.com/CarePathwayModeler/CarePathwayRepository.git

Table 1: An estimation of the level of complexity of care pathways.

Complexity	Number of Elements (Conducts)	Estimated Modeling Time
Simple	10	2 to 4 hours
Medium	16	5 to 10 hours
Complex	20	10 to 12 hours
Very Complex	25+	12 hours+

Table 2 contains some of the modeled care pathways, along with their number of elements, sequences and variables and types of steps, to help establish a broader view of the most complex to the least complex care pathways modeled so far.

Table 2: The modeled care pathways element statistics

Name	#elements	#sequences	#variables
Low Back Pain (hospital)	45	56	45
Diabetes - DM2	39	59	24
Abdominal Pain	34	33	22
Chest Pain	34	38	24
PSHD	30	30	23
Diarrhea	28	45	21
...	...	...	...
Syphilis	8	10	3
Diabetes - UTI	6	5	1
UTI - Pyelonephritis	5	4	3
Exposed Fracture	4	4	2

Table 3 contains other specific information on these pathways, such as the number of prescription and on-site treatment medications, examinations, procedures and referral elements to help approximate the average complexity and size of care pathways. Some of the modeled pathways have over 100 registered medications, examinations and/or procedures, as well referrals to up to 10 other possible pathways. The full table containing the information gathered from all the 85 modeled care pathways can be found here.

Table 3: The modeled care pathways element statistics

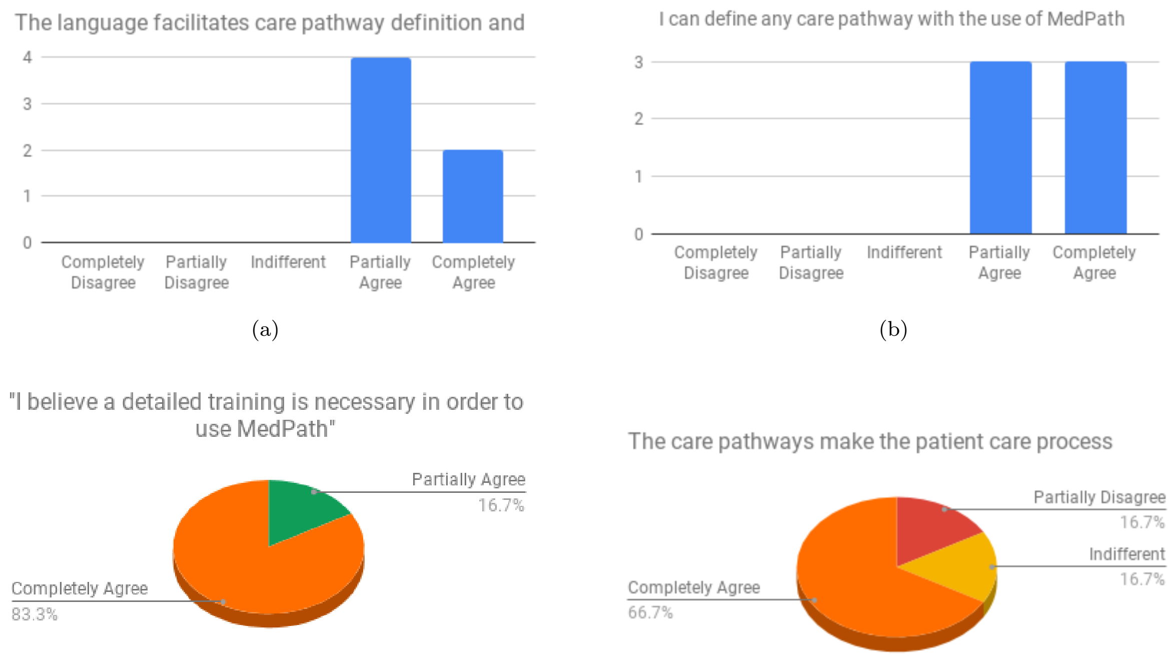
Name	#Treatments	#Prescriptions	#Presc. Meds	#Tmt. Meds	#Exams	#Procedures	#Referrals
Sepsis Treatment	10	0	0	65	154	0	0
Low Back Pain (hospital)	26	4	25	114	29	0	1
Abdominal Pain	17	2	4	2	152	0	3
Diarrhea	14	4	26	0	72	0	0
Chest Pain	13	1	2	0	38	0	1
PSHD	9	0	0	32	18	15	0
URTI	5	5	32	1	5	0	2
...	...	...	...	...	...	...	...
Cephalea (hospital)	5	2	9	26	1	0	0
Febrile Neutropenia	8	1	0	12	13	0	0
Premature Amniorrhexis	8	1	0	13	6	0	0
Toxoplasmosis	4	4	7	0	5	0	0

Additionally, a survey[4] consisting of 10 questions was applied among the professionals responsible for care

[4]The complete survey and its statistic analysis charts can be found at https://docs.google.com/spreadsheets/d/1Pmym2scpS7E7ZHBv1gvHyN4vHdWHmYzLF68DF6xYfZk/edit?usp=sharing

pathway modeling to evaluate the language's efficiency in pathway modeling and the user satisfaction regarding its employment.

The sample was composed by 6 users from different areas of expertise: 3 medical professional, 2 computer scientists and 1 production engineer. Overall, the users are satisfied with MedPath's ability to model care pathways as well as standardize medical procedures. On a Likert scale of 1 to 5, ranging from *Completely Disagree* to *Completely Agree*, all of the users believe that MedPath improves the process of development of care pathways (Figure 6a) and can prevent errors in the medical conduct. They also all agree that as it is, the language can reasonably model any care pathway (Figure 6b), although some feedback was received regarding potential improvements on the elements to accommodate for a better representation of some practices.



(a) (b) (c) (d)

Figure 6: The charts corresponding to some of the questions in the survey.

The users also believe that a the language requires training in order to be properly used (Figure 6c) and some users (33.4%) believe that it does not bring much improvement regarding medical care time (Figure 6d). It was also brought to our attention that the use of care pathways may focus the attention of the medical professional on the computer screen rather than on the patient, which might not be an ideal scenario.

Results and Discussion

The MedPath language is capable of modeling any number of medical pathways. The development of this model took a team of both medical professionals and computer scientists to understand the problem, formalize it, define its scope and develop a language capable of modelling care pathways.

The language is in continued use in a commercial application, with over 1.6 million executions in real-life scenarios, which in and of itself attests to its applicability. Improvements have been suggested and implemented in the DSL as part of the language enhancement loop (the final step in what we called Care Pathway Engineering).

This is a vast and new area of research and improvements to this model can still be done to adapt it to any number of scenarios. In order to model efficient care pathways, a thorough analysis of the process must be carried out prior to the action of modeling. With our DSL, this process becomes easier thanks to the simplification of the notations used in the creation of pathways, as well as the possibility of integration between software solutions that employ this DSL.

For future work, we plan on doing a more thorough study on user satisfaction regarding the use of the DSL, analyzing pathways to predict and prevent path irregularities and analyzing pathway execution data to suggest and perform improvements on care pathways and the modeling process itself.

References

[1] Reinhold Haux. Health information systems – past, present, future. *International journal of medical informatics*, 75 3-4:268–81, 2006.

[2] Martin Burwitz, Hannes Schlieter, and Werner Esswein. Modeling clinical pathways - design and application of a domain-specific modeling language. In *Wirtschaftsinformatik*, 2013.

[3] Jean Bézivin. *Model Driven Engineering: An Emerging Technical Space*, pages 36–64. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[4] Arie van Deursen, Paul Klint, and Joost Visser. Domain-specific languages: An annotated bibliography. *SIGPLAN Not.*, 35(6):26–36, June 2000.

[5] Marlon Dumas and Arthur H. M. ter Hofstede. Uml activity diagrams as a workflow specification language. In *Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools*, «UML» '01, pages 76–90, London, UK, UK, 2001. Springer-Verlag.

[6] Tom Crocker, Owen Johnson, and Stephen F. King. Towards a formalisation of care pathways to embody good practice in healthcare, 09 2007.

[7] Vincent Augusto and Xiaolan Xie. A modeling and simulation framework for health care systems. 44:30 – 46, 01 2014.

[8] Ali Daniyal and Syed Sibte Raza Abidi. Semantic web-based modeling of clinical pathways using the uml activity diagrams and owl-s. In *Proceedings of the 2009 AIME International Conference on Knowledge Representation for Health-Care: Data, Processes and Guidelines*, KR4HC'09, pages 88–99, Berlin, Heidelberg, 2010. Springer-Verlag.

[9] Padraig O'Leary, John Noll, and Ita Richardson. A resource flow approach to modelling care pathways workflow in healthcare. 08 2013.

[10] Object Management Group (OMG). Business process model and notation (bpmn), version 2.0, 2011.

[11] Hubert Scheuerlein, Falk Rauchfuss, Yves Dittmar, Rüdiger Molle, Torsten Lehmann, Nicole Pienkos, and Utz Settmacher. New methods for clinical pathways-business process modeling notation (bpmn) and tangible business process modeling (t.bpm). 397:755–61, 02 2012.

[12] Richard Braun, Hannes Schlieter, Martin Burwitz, and Werner Esswein. Bpmn4cp: Design and implementation of a bpmnextension for clinical pathways, 11 2015.

[13] Emin Gün Sirer and Brian N. Bershad. Using production grammars in software testing. *SIGPLAN Not.*, 35(1):1–13, December 1999.

[14] Arie van Deursen and Paul Klint. Little languages: Little maintenance. *Journal of Software Maintenance*, 10(2):75–92, March 1998.

[15] Jonas Poelmans, Guido Dedene, Gerda Verheyden, Herman Mussele, Stijn Viaene, and Edward Peters. Combining business process and data discovery techniques for analyzing and improving integrated care pathways. pages 505–517, 06 2010.

[16] Harry Campbell, Rona Hotchkiss, Nicola Bradshaw, and Mary Porteous. Integrated care pathways. *BMJ*, 316(7125):133–137, 1998.

[17] Tourginy Mathieu Couturier Tousignan Corbin Delli-Colli Raiche Dubuc, Bonin. Development of integrated care pathways: toward a care management system to meet the needs of frail and disabled community-dwelling older people. page e017, 2013.

[18] Vogus TJSutcliffe KM. The impact of safety organizing, trusted leadership, and care pathways on reported medication errors in hospital nursing units. *Med Care*, 45:997–1002, 2007.

[19] Kris Vanhaecht, Massimiliano Panella, Ruben van Zelm, and Walter Sermeus. What about care pathways? 01 2011.

[20] Kris Vanhaecht, Marcus Bollmann, Kathy Bower, Clare Gallagher, Andrea Gardini, Jen Guezo, Uwe jansen, Rashad Massoud, Karen Moody, Walter Sermeus, Ruben Van Zelm, Claire Whittle, Anne-Marie Yazbeck, Karen Zander, and Massimiliano Panella. Prevalence and use of clinical pathways in 23 countries – an international survey by the european pathway association. *Journal of integrated Care Pathways*, 10(1):28–34, 2006.

[21] M. Panella, S. Marchisio, and F. Di Stanislao. Reducing clinical variations with clinical pathways: do pathways work? *International Journal for Quality in Health Care*, 15(6):509–521, 2003.

[22] Fumihiro Shoji, Tokujiro Yano, Akira Haro, Tsukihisa Yoshida, Kensaku Ito, Yosuke Morodomi, Yoshifumi Wakata, and Yoshihiko Maehara. Assessing a clinical pathway to improve the quality of care in pulmonary resections. *Surgery today*, 41(6):787–90, Jun 2011.

[23] K J Zehr, P B Dawson, S C Yang, and R F Heitmiller. Standardized clinical care pathways for major thoracic cases reduce hospital costs. *The Annals of thoracic surgery*, 66(3):914–9, Sep 1998.

[24] Svin Deneckere, Martin Euwema, Pieter Van Herck, Cathy Lodewijckx, Massimiliano Panella, Walter Sermeus, and Kris Vanhaecht. Care pathways lead to better teamwork: results of a systematic review. *Social science medicine (1982)*, 75(2):264–8, Jul 2012.

[25] Svin Deneckere, Martin C. Euwema, Cathy Lodewijckx, Massimiliano Panella, Timothy Mutsvari, Walter Sermeus, and Kris Vanhaecht. Better interprofessional teamwork, higher level of organized care, and lower risk of burnout in acute health care teams using care pathways: a cluster randomized controlled trial. *Medical care*, 51 1:99–107, 2013.

[26] Object Management Group (OMG) Soley R. Model-driven architecture, 2000.

[27] Arie van Deursen and Paul Klint. Domain-specific language design requires feature descriptions. pages 1 – 17, 2002.

[28] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[29] David Steinberg, Frank Budinsky, and Marcelo Paternostro. Merks, emf: Eclipse modeling framework 2.0, 2009.

[30] Obeo Designer key features. https://www.obeodesigner.com/en/product/key-features. Accessed: 2017-03-15.

[31] Amine El Kouhen, Cedric Dumoulin, Sébastien Gérard, and Pierre Boulet. Evaluation of Modeling Tools Adaptation. Technical report, January 2012.

[32] V. Viyović, M. Maksimović, and B. Perisić. Sirius: A rapid development of dsm graphical editor. In *IEEE 18th International Conference on Intelligent Engineering Systems INES 2014*, pages 233–238, July 2014.

[33] Sirius getting started. https://www.eclipse.org/sirius/getstarted.html. Accessed: 2017-03-15.

[34] Thierry Biard, Alexandre Le Mauff, Michel Bigand, and Jean-Pierre Bourey. Separation of decision modeling from business process modeling using new "decision model and notation" (dmn) for automating operational decision-making. In Luis M. Camarinha-Matos, Frédérick Bénaben, and Willy Picard, editors, *Risks and Resilience of Collaborative Networks*, pages 489–496, Cham, 2015. Springer International Publishing.