MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation P. Otero et al. (Eds.) © 2022 International Medical Informatics Association (IMIA) and IOS Press. This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0). doi:10.3233/SHTI220021

Handling Time Constraints in Infection Clinical Pathways Using openEHR TP

Natalia Iglesiasa , Jose M. Juareza , Manuel Camposa,b

^aAIKE Research Group (INTICO), University of Murcia ^bMurcian Bio-Heatlh Institute (IMIB-Arrixaca)

Abstract

Clinical Pathways (CP) provide healthcare personnel with an easy-to-understand high level model of medical steps in specific patient conditions, thereby improving overall process quality in clinical practice. The emergence of new clinical-oriented standards such as openEHR Task Planning (TP) could pose a major step towards clinical process improvement, particularly in complex domains such as infection diagnosis and treatment, where time plays a critical role. In this work, we analyze the suitability of TP to successfully represent time constraints of common process patterns in infections, modelling some of the Catheter-Related Blood Stream Infection (CR-BSI) process patterns as a case study. Our research shows that TP is useful to represent time constraints of infection CPs, although minor improvements could increase its suitability not only for infection processes but for other time-related complex clinical scenarios.

Keywords:

Clinical Pathways; openEHR Task Planning

Introduction

Clinical Pathways (CPs) are institution-specific care protocols that describe all possible paths of a single patient evolution over *time* in specific clinical settings, integrating the institution's organizational roles and resources. CPs often rely on medical evidence expressed in Clinical Guidelines (CGs), that standardise abstract best practices for the diagnosis and treatment of specific medical conditions. Whereas CGs focus on "what" needs to be done, CPs focus on "when" tasks should be done, describing the *temporal* context in which clinical work needs to be carried out. A correct integration of CPs into an institution's workflow is therefore a major requirement for their successful application in real clinical settings.

CPs are usually represented using general-purpose Business [Process Management \(BPM\), defined as "](https://www.omg.org/bpmn/)*process management [strategy focus](https://www.omg.org/bpmn/)ed on the continuous improvement of business processes*[" \[1\]. BPM aims at eliminating](https://specifications.openehr.org/releases/PROC/latest/task_planning.html) process inefficiencies or reorganizing low-value futile tasks. Despite its ease of use, it shows important shortcomings in clinical domains, mainly due to the lack of adequate out-of-the-box *temporal* capabilities. In the last years, the general-purpose OMG's *Business Process* Management Notation (BPMN¹), has emerged in clinical domains as a process management standard [2], though extensions had to be implemented to handle time constraints [3] and the transition to process execution in clinical institutions is still scarce [4]. More recently, new *clinical-oriented* process representation standards are emerging that address specific needs of clinical workflows. OMG has released the *Case Management Model and Notation* (CMMN) that extends BPMN for multi-activity cases that respond to unpredictable situations, and the *Decision Model and Notation* (DMN) that enables the definition of complex business logic by users in the form of process rules. The 3 OMG standards (the "*triple crown*") can be used independently or simultaneously [5]. Further, the new HL7 service-oriented extensible standard *Fast Healthcare Interoperability Resources* (FHIR), successor of *HL7 V2, V3* and *CDA* v2, makes heterogeneous healthcare data accessible and exchangeable using a RESTful API and it has already been used in many research works [6]. Finally, the openEHR Foundation released in 2016 a new Workflow Management standard, the openEHR *Task Planning* (TP) specification², which includes a Visual Modeling Language³ backed up by a formal semantics, which ultimately would allow for automatic translation of graphical workflow models into executable models. TP allows modelling of orders or actions in the future, including *parallelism* and *events* supporting absolute and relative time references. TP is conceived as a clinical process navigator, empowering users to perform ad-hoc modifications at execution time to reflect on real-time changes. While openEHR is an active area of research⁴, the TP specification has been used to our knowledge in [7] to represent the obstetric care process in 9 Portuguese hospitals, or in [8], where a model is proposed for process-oriented traceability of CPs.

Methods

Our work is based on the hypothesis that TP provides a suitable way for representing *time* dependencies in clinical infection processes. We showcase the John Hopkins Hospital (JHH) *Catheter-Related Blood Stream Infection* (CR-BSI) [9], a [highly recurring infection and a major cause of morbidit](https://specifications.openehr.org/releases/PROC/Release-1.5.0/tp_vml.html)y and [costs in hospitals, caused by gram-positive bacteria present](https://www.zotero.org/groups/11839/openehr/items/RH8PH268/library) in Intravascular catheters, which treatment options depend on the microorganism causing the infection, the type of antibiotic and the time, frequency and dosage of its intake [10]. The type of catheter, the way it is handled by healthcare personnel, the duration of its placement, or the patient condition codetermines the risk of a hospital-acquired CR-BSI, especially in critically ill patients whose catheter is not removed [11].

³ Task Planning Visual Modelling Language (TP-VML) (openehr.org) ⁴ https://www.zotero.org/groups/11839/openehr/items/RH8PH268/library

¹ Business Process Model & Notation™ (BPMN™) | Object Management Group (omg.org)

² Task Planning (TP) Specification (openehr.org)

Our methodology consisted of 1) analysis of the suitability of the extended BPMN standard and TP, comparing key temporal features for infection modelling; 2) representation and comparison of CR-BSI process patterns using both standards; and 3) proposal of TP improvements in view of the results of the use-case scenario. The BPMN-based CR-BSI process models are taken from [12,13], and they contain extensions to address time constraints, such as task and edge duration. For modelling the corresponding CR-BSI TP workflows, we have used the open source $draw.io^5$ tool with the VML libraries⁶.

Results

A. Identification and comparison of key temporal features. According to state of the art [3,13] key generic and specific features for modelling infection CPs are identified in **Table 1**, where we review the expressivity of both extended BPMN and TP standards to cover them:

Table 1 Coverage of key features in Extended BPMN and TP –

Feature	Extended-BPMN	TP	Type
Structured workflow	SESE restrictions	Built-in	Generic
Process modularity	Call Activity	Dispatchable tasks/subplans	Generic
Events	Timer/Signal events	Task Transition/State Trigger events	Generic
Parallel execution	Gateways	Concurrency mode	Generic
Task Duration	BPMN extension	N/A	Specific
Relative time con- straints	BPMN Extension	Built in task waits	Specific
Use of resources	Minimally defined	Minimally defined	Specific
Multiple tasks	Multiplicity marker	Repeatable tasks	Specific
Delays between itera- tion	N/A	Repeat attribute "period"	Specific
Data integration	N/A, uses UML	Capture datasets / subject's proxy services	Specific
Overrides	Only Add Activi- ties/events	Built-in behaviour	Specific

As we can see, deterministic and non-deterministic events, parallel execution, task duration, relative time constraints between tasks and delays between iterations of looping tasks are key temporal features for modelling infection CPs.

B. Infection patterns. The diagnosis and treatment of infections share some common distinctive patterns, typical for diseases caused by bacterial microorganisms: 1) infections need lab tests to determine the causing pathogen; 2) they must be treated readily to avoid complications that could become lifethreatening, according to CGs empiric evidence; 3) antibiotic treatment must follow strict temporal administration rules to be effective; and 4) patients must be monitored throughout the whole process, to quickly adjust the treatment in case of problems. In our analysis we have compared the extended-BPMN CR-BSI pattern models of [12,13] with matching TP models, to empirically check the TP suitability for modelling time constraints in infection CPs.

Pattern of "Empiric Treatment" (PET): When suspicion of an infection exists, the clinical approach is to immediately start treating the patient with a wide-spectrum antibiotic following empirically gained knowledge usually laid out in CGs, while [lab tests are ordered](https://app.diagrams.net/) in parallel to identify the causing microorganism. As soon as lab tests are available, the ET is revised and adjusted if required, either by changing its dosage or duration, or replacing it with an antibiotic specific for the concrete pathogen causing the infection. PET is represented in BPMN using deterministic events that depend on the reliability of a clinical institution response times for test results (**Fig.1**), while in TP a non-deterministic approach is chosen by ending the "*ET*" antibiotic administration loop when the "*Lab Tests*" parallel task group ends, providing more flexibility and possibly shorter execution times (**Fig.2**).

Figure 1 BPMN representation of PET [13]

Figure 2 TP representation of PET

Pattern of "*Adjustment of course of therapy*" (*PACT*), showcased with the JHH CGs *S. Aureus* therapy: "*Criteria for a 14-day course of therapy: patient is clinically stable; followup blood cultures drawn 2-4 days after the initial cultures are negative for S. Aureus; the patient defervesces with 72 hours of initiation of effective anti-staphylococcal therapy. All other patients should receive 4-6 weeks of therapy based on extent of infection.*" The BPMN model of this pattern uses 3 activities at predefined points in time after starting therapy, in a purely deterministic approach (**Fig. 3**).

Figure 3 S.Aureus PACT using BPMN

Fig. 4 shows a possible corresponding TP model using deterministic events. Therapy duration is decided in "*Review Duration*", launched after tasks "*Measure Temperature*" and "*Check culture results*" are finished, to decide if therapy should continue as planned or be adjusted to a minimum of 28 days and a

maximum of 42 days, thus an extension of 14-28 additional days. We have used a parallel "*and-all-paths*" group with 2 branches: the first branch is a parallel group "*Follow-up*" representing the "*Measure Temperature*" and "*Check culture results*" time-driven tasks, executing at day 3 and 4 respectively; the second branch is a repeatable group executing the initial 14 days "*Short therapy*" task. Once "*Follow-up*" ends, "*Review duration*" is launched asynchronously through a hand-off dispatchable task, to determine if what started as "*Short therapy*" should be extended to a "*Long-therapy*" or continue as planned. When the initial 14 days "*Short therapy*" ends, if long therapy was decided in "*Review duration*", a "*Therapy extension*" repeatable task is started for additional 14-28 days. This extended therapy repeats unconditionally until the minimum number of iterations (14 d) is reached, after which it can be interrupted if patient evolution is positive, via the *Repeat-spec terminate condition*, ending "*Therapy extension*". In the most extreme scenario, *Therapy extension*" will end when the *repeat.upper* limit (28 in our case) is reached. Although this is a valid model, we had to do some maths with the days in order to faithfully represent treatment duration.

Figure 4 S.Aureus PACT using TP VML

The purely deterministic scenario drawn in this model, could be represented as shown in **Fig. 5,** where the main difference is that in case treatment duration is adjusted by the "*Adjust treatment*" task, a *Capture Dataset* (form) would collect the new treatment information, update the EHR accordingly and signal a new *repeat* "*override-condition*" through a *task transition event,* that would reset the *repeat* metadata attributes to *repeat.lower*=28 (instead of 14*)* and *repeat.upper*=42.

Figure 5 Extensions proposed for TP VML in S.Aureus PACT

Discussion

In our experiments we have been able to model all time constraints of the CR-BSI patterns using exclusively standard TP constructs. In some cases, we identified possible improvements to the TP standard to make it even more suitable for modelling time constraints in infection CPs:

- 1. The TP *concurrency modes* were useful in representing temporal dependencies of infection processes, especially the *and-all-paths*.
- 2. The BPMN "*duration*" extension, widely used in the BPMN diagrams, does not exist in TP and has been replaced by dynamic behavior through non-deterministic events triggering a *repeat terminate-condition* (e.g. "PET"). Despite not being strictly required, there could be thinkable scenarios in which *duration* is required, as intrinsic property of any piece of work that could be used as input for other TP constructs (e.g. Repeat loops, to determine the number of iterations in a treatment). Consequently, we suggest adding optional duration attributes to the TP Plan Item parent class of a task: minDuration, maxDuration, timeUnits and durationUse.
- 3. The next most used construct was the *repeat* attribute, to represent medication plans, therapy administration or monitoring of patient vital signs. The TP *repeat* construct mimics both a "*for*" programming loop till the *repeat.lower* limit is reached, and a "*while*" programming loop beyond that point. Being essential for modelling infection CPs, it could be further enhanced to allow *on-the-fly* modification of metadata attributes, with as a result a much simpler and more adaptive representation (**Fig.5**). We suggest extending the *Repeat-spec* class with new interrupt/end conditions (*start-*, *skip-* and *override*-conditions), and an improved *terminate-condition* to allow logical expressions or DLM rules. The execution logic of the proposed *Repeat-spec* class would be: if *start-condition* is true and *repeat.lower* is greater than zero, a loop is started of "1 to *repeat.lower*" iterations (the "*for*" part of the repeat loop). This minimum number of iterations is executed unconditionally, unless a *skip-condition* is met in any of the iterations. Only when iteration number "*repeat.lower + 1*" is reached, the rest of the conditions could apply, that is, *override*- and/or *terminate* condition.
- 4. Further we have made extensive use of the TP *Task Transition Event*, in order to signal task completion (e.g. "PET"), although the *state trigger event* could have been used instead, as it signals changes in environmental variables monitored by the TP engine. TP temporal relations between tasks are based on *waits* that hold task execution. To this end, a list of both deterministic (*timer, timeline, calendar*), and non-deterministic events (*task transition, manual, system or callback notification, state trigger*) to be "waited on" can be specified in the *Task-wait* class *eventslist* attribute. The *wait* of a given task stops when any of the events become true (a logical OR). We propose to add an "*event-list-relation*" attribute in the *Task-wait* class, with possible values *OR|AND*, so that a task can wait on the simultaneous occurrence of more than one event, in the fashion of a *Complex Event Processing* (CEP) system. As this might require other deep-going adaptations of the TP logic, a thorough cost/benefit analysis should be done on this specific feature. Furthermore, TP has an optional *eventrelation* attribute in the *Task-wait* class, intended to "*allow a task to be specified as commencing before, with or after the triggering event*" with 3 possible values: *before, with*

and *after*. In case of deterministic triggering events, the *event-relation* attribute could be further refined with a new "*offset*" attribute expressed in positive or negative time units, relative to the triggering event, as a "time quantifier", to add or subtract a concrete amount of time to the firing time of the *task-wait* triggering event(s), causing the *taskwait* to be delayed or advanced.

- 5. We have analyzed the time dependencies between TP tasks from a pragmatical point of view, checking if they meet the 13 Allen time interval operands [14]. We found that 3 pairs of the 6 symmetric Allen interval operands on task execution are naturally implemented in the TP standard:
	- - *Precedes/preceded by* is built upon the default behavior of the TP execution type *sequential*, that allows a task t2 to become available when the predecessor task t1 is completed, **plus** a *task-wait* in front of the successor task t2.
	- - *Meets/met by* allows a task t2 to become immediately available for execution when the predecessor task t1 is completed, provided that no *task-waits* are included (default behavior of *sequential* tasks).
	- - *Starts/started by* is achieved by modelling a parallel group with tasks t1 and t2 **plus** no *task-waits* included, so tasks t1 and t2 start naturally at the same time.

The other 3 pairs of symmetric Allen interval operands plus the "*equals*" operand impose restrictions upon the tasks' end and/or start times, always in parallel execution settings, and need more fine-grained time expressions, not always obvious in the current version of the TP standard:

- - *Overlaps/overlapped by* can be modelled using a parallel group with *task-waits* either on *t1* or *t2*. If the overlapping part of both tasks needs to be specified more accurately, this pattern should include global *timeline task-waits* to each task, either relative to the Plan start time, or ideally, relative to the first starting task (*t1* or *t2*). This last scenario would however require an extension of the *Plan-Time-Origin* enumeration class, with a new attribute *"plan-item*" to specify a list of *Plan items* to be used as relative reference for the *timeline* offset.
- - *Finished by/finishes* specifies that parallel tasks *t1* and *t2* need to end at the same time. Generally speaking, an *and-all-paths* parallel group could be used, as it ends only when all branches are finished. However, the time to completion cannot be enforced, as it depends on the longest executing branch. Thus, for a compulsory end time for *t1* and *t2*, a deterministic *end time* restriction should be applied to each task of the parallel group. We could not find a way to implement this in the current TP version, a possible solution could require an extension of the *Plan Item* class with a new attribute *end time* of type *duration* or alternatively, *task-waits* should apply not only to transition a task to the "*available*" state, but to any state, or at the very least, to the "*completed*" state.
- - *Contains/during* states that *t2* must be started after *t1* and must end before *t1* ends (or viceversa). The start of *t2* after *t1* can be implemented as in the *Overlaps/overlapped* case. The end of *t2* before *t1* is however not that obvious to implement, as *task-waits* are not an option, because both tasks are already executing by then. In the current TP version, we could not find a reasonable way to express that "*t2* must end *x* units of time before the end of tI ", as relative time

constraint. To make it feasible, *task-waits* should not only apply to transition a task to "*available*" state, but to any state, or at the very least, to the "*completed*" state, as in the *Finished by/finishes* pattern. In that case, a combination of a "*task-wait to complete*" having as triggering event the end of *t2* with a time quantifier, would allow for a fine-grained expression of this pattern.

- *Equals* states that tasks *t1* and *t2* must be executed at the exact same time. This can for example be achieved with a parallel group with no *task waits* and a specific restriction on each task duration, thus one possible solution would require the abovementioned *duration* extension.

As a summary, **Table 2** shows a list of the proposed TP improvements, which are of potential interest to other complex clinical settings.

Table 2 Summary of proposed TP extensions for infection CPs –

TP extension	Where	Reason
New duration attributes	Plan Item	Informational and support Allen
New Repeat Condition attributes	Repeat-spec	Enrich repeat conditions
New Repeat terminate-condition value	Repeat-spec	Specify a DLM rule/expression
New Repeat behaviour	TP-engine	Override repeat attributes at exec.time
New Resource attributes	Resource-	Extend Resources perspective
	Participation	
New Resource allocation logic	TP Engine	Behaviour for exclusive resources
New Resource transition event	Events	Detect a resource state transition.
New Resource state machine	TP Engine	Transition between states of resources
New "event-list-relation" attribute	Task-wait	Logical relation between events (AND)
New "offset" attribute	Task-wait	Time spacer of deterministic events
New "resume-type" value	Resume-action	Allow DLM rule
New "end-time" attribute	Plan Item	Support Allen Finishes/finished by
New "plan-item" attribute	Plan-item-	Support Allen Overlaps/overlapped by
	origin	
New "Task-wait" behaviour	TP-engine	Support Allen Contains/during

Table 3 shows a comparison of the existing features of extended-BPMN and standard TP with our proposed extended TP model, represented in the TP* column.

Table 3 – Proposal of TP improvements for each key feature

Feature	Extended BPMN	TP	$TP*$
Structured workflow definition	Extension SESE	Built-in	None
Process modularity	Call Activity	Dispatchable tasks in sync. or async. mode/SubPlans	New Resume-type in sync. dispatch
Events	Timer & Signal events	Rich set of specialized events	New Resource Transition event
Parallel execution	Gateways	Concurrency mode	None
Task Duration	BPMN Extension	N/A	New task attributes
Relative time constraints between tasks	BPMN Extension	Built-in task-waits	New plan-item, plan-item-origin attributes and TP
			engine task wait hehavior
Use of resources	Minimally defined	Minimally defined	New resource attributes and hehavior
Multiple tasks	Multiplicity marker	Repeatable tasks	New Repeat conditions and hehavior
Delays between iterations of looping tasks	N/A	Repeat attribute "period"	None
Data integration	N/A. uses UML	openEHR/ Capture datasets / subject's proxy services	Mapping between EHR records and TP metadata
Overrides at execution-time	Few exceptions (Add activities/events)	By design (Remove or add tasks, plan parameters and subject preconditions)	Override of model metadata (repeat) information

Conclusions

In this paper we have analyzed the representation of time constraints for modelling infection CPs using extended-BPMN models of common patterns of the CR-BSI infection and the TP standard notation. We were able to successfully translate the extended BPMN diagrams into the TP specification with no need for extensions, showing the out-of-the-box adequacy of TP for modelling time constraints in infection CPs. However, we have identified improvement opportunities taking advantage of the rich TP metadata model, which could increase

the suitability of the standard for modelling infection processes and potentially for other time-critical scenarios. The timerelated extensions are mainly focused on increased nondeterministic synchronization between tasks and improved repeatable constructs with the possibility of overrides affecting the metadata model. We also recognize the importance of the Resource perspective in clinical models, as resources often affect clinical decisions [15].

Future work will focus on the *openEHR Decision Language Module* (DLM) specification⁷, a new openEHR Foundation formalism for defining rules callable from TP and expressed via the *openEHR Expression Language*⁸ .

Acknowledgements

This work was partially funded by the SITSUS project (Ref:RTI2018-094832-B-I00), supported by the Spanish Ministry of Science, Innovation and Universities (MCIU),the Spanish Agency for Research (AEI) and by the European Fund for Regional Development (FEDER).

References

- [1] H. Smith and P. Fingard, Business Process Management: The Third Wave, *Meghan-Kiffer Press*.
- [2] M. Ramos-Merino, LM. Álvarez-Sabucedo, J.M. Santos-Gago, J. Sanz-Valero, A BPMN Based Notation for the Representation of Workflows in Hospital Protocols, *Journal of med. Syst.* 42(10) (2018), 181.
- [3] C. Combi, M.Gozzi, R. Posenato,R. and G. Pozzi, Conceptual modeling of flexible temporal workflows, *ACM Trans. Auton. Adapt. Syst.* 7, 2, 19 (2012), 29 pp.
- [4] [A. de Ramón Fernández, D. Ruiz Fernández and Y. Sabuco García, Business process management for optimizing clinical processes: A systematic literature review, Health Informatics Journal, 26(2) (2020), 1305– 1320.
- [5] M. Wiemuth, D. Junger, M. Leitritz, J. Neumann, T. Neumuth and O. Burgert, Application fields for the new Object Management Group (OMG) Standards Case Management Model and Notation (CMMN) and Decision Management Notation (DMN) in the perioperative field, *Int J of Comp-assist radiology and surgery* 12 (2017).
- [6] M. Baskaya, M. Yuksel, GBL Erturkmen, M. Cunningham and P. Cunningham, Health4Afrika - Implementing HL7 FHIR Based Interoperability, *Stud Health Technol Inform,* Aug 21;264 (2019), 20-24.
- [7] D.S. Alves, P.A. Maranhão, A.M. Pereira, G.M. Bacelar-Silva, T. Silva-Costa, T.W. Beale and R.J. Cruz-Correia, Can openEHR represent the clinical concepts of an obstetric-specific EHR - ObsCare software? *Stud Health Technol Inform.* Aug 21;264 (2019), 773-777.
- [8] [F. Frexia, Innovative information models to capture the](https://specifications.openehr.org/releases/PROC/latest/decision_language.html) [dynamics of clinical process](https://specifications.openehr.org/releases/PROC/latest/decision_language.html)es: introducing process-oriented traceability in medical informatics specifications as a case study, PhD thesis (2019), Università degli Studi di Cagliari.
- [9] S.E. Cosgrove and E. Avdic, Antibiotic guidelines 2013- 2014, Johns Hopkins Medicine (2013).
- [10] Malek and I. Raad, Catheter- and device-related infections in critically ill cancer patients, *Oncologic Critical Care* (2019).
- [11] J. Garnacho-Montero, T. Aldabó-Pallás, M. Palomar-Martínez, J. Vallés, B. Almirante, R. Garcés, F. Grill, M. Pujol, C. Arenas-Giménez, E. Mesalles, A. Escoresca-Ortega, M. de Cueto and C. Ortiz-Leyba, Risk factors and prognosis of catheter-related bloodstream infection in critically ill patients: a multicenter study, *Intensive Care Med*, 34(12) (2008), 2185-93.
- [12] F. Zerbato, BPMN-based design and comparison of clinical pathways and data for catheter-related bloodstream infections, MSc thesis (2015), University of Verona.
- [13] F. Zerbato, B. Oliboni, C. Combi, M. Campos and J. M. Juarez, BPMN-Based Representation and Comparison of Clinical Pathways for Catheter-Related Bloodstream Infections, *International Conference on Healthcare Informatics* (2015), 346-355.
- [14] J. F. Allen, An interval-based representation of temporal knowledge, *7th international joint conference on Artificial intelligence* 1 (1981), 221-226.
- [15] R. Braun, M. Burwitz, H. Schlieter and M. Benedict , Clinical Processes from Various Angles - Amplifying BPMN for Integrated Hospital Management (2015), 10.1109/BIBM.2015.7359794.

Address for correspondence

Corresponding author: Natalia Iglesias Email: natalia.iglesias@um.es