

Data-Aware Monitoring For Healthcare Workflows Using Formal Methods

Ji Ruan and Wendy MacCaul

Centre for Logic and Information, StFX University, Canada, {jruan, wmaccaul}@stfx.ca

Abstract. This paper presents an approach to monitor healthcare workflows using a logic-based formal method. We introduce a monitoring architecture with workflows and knowledge bases, and propose a logical language, FO-LTL-K, to express temporal and knowledge properties to be monitored. We formalize some of the norms for palliative care using the proposed logic and characterize the complexity of the model checking problem.

Keywords: Data and knowledge integration, Temporal knowledge representations, Logic-based methods, Ontologies, Complexity.

1 Introduction

A workflow [17] is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules. Healthcare workflows have been proposed to improve the efficiency of health services delivery [10, 6]. Using model checking, initial attempts have been made to verify the correctness of healthcare workflows at the design stage [7, 12]. Monitoring (which uses data generated during care and relevant background healthcare knowledge) is necessary to determine the correctness of healthcare workflow execution. In this paper we develop an approach to monitor healthcare workflows using a logic-based formal method.

Logic-based formal methods provide a solid theoretical foundation for modelling and verifying complex systems. Three kinds of logic relate to our work: First-order logic (FOL), Temporal logic (TL), and Description logic (DL). FOL [16] enables us to represent different healthcare data types and quantify over their contents. TL is a formalisation describing temporal changes. It is applied to the verification of computer programs, or more broadly, to reactive systems [11]. Healthcare data reflects the dynamics of healthcare systems, such as the changes of health status of patients and the delivery of care, therefore we must reason about temporal changes. DL [1] models individuals, concepts, and roles. It is used in Artificial Intelligence for formal reasoning on the concepts of an application domain (e.g., Medicine), and is of particular importance in providing a logical formalism for knowledge bases, also called ontologies.

This paper is organized as follows: section 2 introduces palliative care and a monitoring architecture; section 3 discusses the representation of healthcare knowledge and data; section 4 proposes a logical language, **FO-LTL-K**, to express the properties to be monitored, and uses it to formalize some palliative care norms; section 5 provides the complexity of the model checking problem; section 6 discusses related and future work.

2 Palliative Care and A Monitoring Architecture

Palliative care refers to the physical, psychological, spiritual and practical care given to patients and their families when they are dealing with the issues associated with serious illness. As patients are usually part of a family, when care is provided, the patient and family are treated as a unit. The main focus of palliative care is to ease the suffering of the patient and his or her family and to help them to cope with their difficulties.

According to [3], a palliative care workflow consists of the following essential processes: *Referral, Consultation, Intake, Therapeutic Encounter, and Discharge*. These processes can be further refined as sub-workflows. For example, a therapeutic encounter is defined as a sub-workflow with the following processes: *Assessment, Information Sharing, Decision Making, Care Planning, Care Delivery, and Confirmation*. It is depicted in Fig. 1, in which the black triangle and square represent the start and end of the workflow respectively, the arrows indicate the order of the processes to be carried out, and the double-lined squares indicate compound processes. These processes can be further refined to include a greater level of detail.

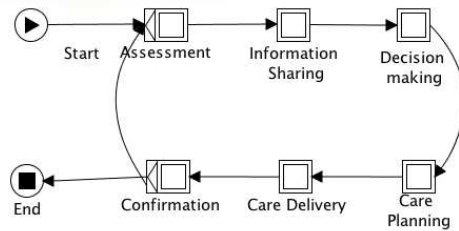


Fig. 1. A Therapeutic Encounter Workflow

Many stakeholders are involved in the processes of palliative care, e.g., Nurses, Physicians, Oncologists, Pharmacists, and Social workers. These stakeholders carry out the practical work and record various kinds of healthcare information. Their work is guided by certain requirements to ensure the quality of care and to prevent errors. In Canada, the norms of practice [3] are set by the Canadian Hospice Palliative Care Association to guide the process of palliative care. Norms are simple statements that provide a benchmark of normal or desired practice to which individuals or organizations providing palliative care can and should aspire. They are less rigid and less specific than standards, thus enabling individual organizations to develop their own standards of practice from the norms.

We propose a monitoring architecture to check whether the actual healthcare practice meets these norms. It has the following four components (Fig. 2): (1) a Workflow Management System (WfMS) which defines, creates and manages the execution of workflows through the use of a workflow engine able to interpret the process definition and interact with workflow participants; (2) a Data Management System (DMS)

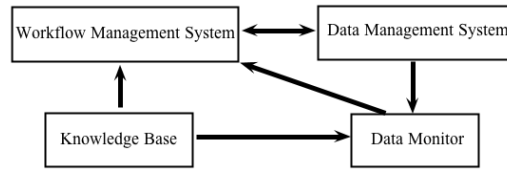


Fig. 2. A Monitoring Architecture

which stores and retrieves the information that is produced by the execution of workflows in WfMS; (3) a Knowledge Base (KB) which holds medical, organizational, and healthcare knowledge to support the execution of workflow; (4) a Data Monitor which monitors the data flow from the DMS and uses the knowledge from the KB to help healthcare stakeholders with decision-making.

3 Healthcare Knowledge and Data Representation

There are different kinds of information contained in a healthcare system. We classify them into two kinds in our architecture. The first is static, or changes infrequently over time—e.g., the knowledge of medication compatibility and the relationship of stakeholders in healthcare; the second is dynamic, hence may change frequently over time—e.g., the assessment result of pain may reduce from level 9 to level 3 due to proper treatment.

The first kind of information comes from the healthcare domain knowledge, and is maintained in the KB. There is a vast amount of knowledge related to healthcare, such as the knowledge of disease and human coping, the medication compatibility, and the responsibilities of different caregivers, etc. An ontology is a formal representation of a set of concepts within a domain and the relationships among those concepts. Extensive ontologies for the healthcare domain, such as the International Classification for Nursing Practices (ICNP) and the Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT), as well as numerous drug ontologies have been developed for the electronic exchange of clinical health information. Many ontologies are constructed using some fragment of DL. Efficient DL reasoners [15] are available, and are useful both in the development stage of the ontology (e.g., for consistency checking) and for querying the ontology. We use DL to encode knowledge related information to facilitate the use of DL reasoners.

The second kind of information is produced during the care delivery processes, and is managed in the DMS. A basic healthcare data unit is called a healthcare record, which has three required data fields: the patient's ID, a time stamp, and an electronic signature of the caregiver who provided the record. The time information is critical in healthcare since some tasks must be completed within a certain time limit, and some tasks should be repeated regularly. The granularity of time in palliative care is usually measured by hours or even days. According to the time stamp of each record, the healthcare records of a patient can be retrieved, and form a linear order. Thus, we use a linear temporal model to represent a patient's healthcare history. Apart from the three required data

```

<record>
<time>9am January 6, 2010</time>
<patientID> B9527 </patientID>
<pain>9</pain>
<weakness>6</weakness>
<medication> Med1 </medication>
<medication> Med2 </medication>
<actiontaken> Assessment; B9527</actiontaken>
<signature> Bob </signature>
</record>

```

Fig. 3. A healthcare record in XML format.

fields, each healthcare record will have other data fields, depending on what is intended to be captured at the moment. For example, it may include assessment results for pain, tiredness, depression, etc., or the medication that is prescribed, or other data. We use a standard XML format to encode and facilitate the exchange of such healthcare records.

Definition 1 (Healthcare Data Model). *A healthcare data model \mathcal{M} is a finite sequence $r_1 r_2 \dots r_n$ where r_i is a healthcare record. A healthcare record contains tags and data. Data content is included in tags in the format: $\langle \text{tag} \rangle \text{content} \langle / \text{tag} \rangle$, and content items within the same tag can be separated using ‘;’. The whole record is wrapped by a pair $\langle \text{record} \rangle \langle / \text{record} \rangle$. In particular, patient’s ID, time stamp, and caregiver’s signature are tagged with $\langle \text{patientID} \rangle$, $\langle \text{time} \rangle$, and $\langle \text{signature} \rangle$ respectively.*

An example of a healthcare record in Fig. 3 shows (a) that the patient (ID ‘B9527’) underwent an assessment (which gives pain level 9 and weakness level 6) by caregiver Bob at 9am January 6, 2010, and (b) the prescribed medication (Med1 and Med2).

4 Specifications

Specifications express properties that should hold in a system. We propose a language called **FO-LTL-K**, which is a combination of a first-order linear temporal language and a description logic language. It is designed:

- to represent and quantify over the data contents in a health record, e.g., all the medications inside the tag $\langle \text{medication} \rangle$;
- to specify real-time requirements in different intervals, e.g., an assessment must be made within 3 hours (or 3 days);
- to specify temporal changes of data, e.g., in one assessment, the pain level is 4, but in the next assessment, it is 8;
- to specify healthcare knowledge expressions, e.g., medication Med1 is not compatible with medication Med3.

Definition 2 (Syntax). *The language **FO-LTL-K** is constructed from the alphabet:*

- Predicate symbols P_1, P_2, \dots , each of which is some fixed arity;
- Function symbols F_1, F_2, \dots , each of which is some fixed arity;
- Concept symbols (also called atomic concepts) C_1, C_2, \dots ;
- Role symbols (also called atomic roles) R_1, R_2, \dots ;
- Individual variables x_1, x_2, \dots ; Individual constants c_1, c_2, \dots .

Terms t : Variables and constants are terms; $F(t_1, \dots, t_n)$ is a term if F is an n -ary function symbol and t_i is a term.

Concepts C, D : $C ::= C_i \mid \top \mid \perp \mid \neg C \mid C \sqcap D \mid \forall R.C \mid \exists R.C$, where, C_i is a concept symbol and R is a role symbol.

Formulas $\varphi ::= t_1 = t_2 \mid P(t_1, \dots, t_n) \mid u_1 : C \mid (u_1, u_2) : R \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U} \psi \mid \forall x.\varphi(x)$, where t_i is a term, x is a variable, u_i is a constant or variable, P is a predicate, C is a concept, and R is a role. We use the following abbreviations for other connectives and operators: $C \sqcup D = \neg(\neg C \sqcap \neg D)$, $\varphi \vee \psi = \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \rightarrow \psi = \neg(\varphi \wedge \neg\psi)$, $\mathbf{F}\varphi = \top \mathbf{U} \varphi$, $\mathbf{G}\varphi = \neg\mathbf{F}\neg\varphi$, $\exists x.\varphi(x) = \neg\forall x.\neg\varphi(x)$.

A variable x is a free variable if it is not in the scope of any $\forall x$ or $\exists x$. A formula is a sentence if it does not have any free variables.

Intuitively, a temporal formula $\mathbf{X}\varphi$ means “in the next record, φ holds”, $\mathbf{F}\varphi$ means “in a future record, φ holds”, and $\varphi \mathbf{U} \psi$ means “ φ holds until ψ holds”. The precise meanings will be given in Definition 6. We introduce a formal definition of knowledge bases.

Definition 3 (Knowledge Base). A knowledge base \mathcal{K} is a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$, where \mathcal{T} (the TBox) is a set of expressions of the form $C \sqsubseteq D$ ($R \sqsubseteq S$) and $C \equiv D$ ($R \equiv S$) for concepts C, D (and roles R, S), and \mathcal{A} (the ABox) is a set of expressions of the form $(c_1 : C)$ and $((c_1, c_2) : R)$ for constants c_1, c_2 , concept C and role R .

Here are a few examples to show the intuitive meaning of knowledge expressions.

- Oncologist, Nurse, Physician, Caregiver, Manager : These are concepts that express the stakeholders associated with a care unit. $(Alice : \text{Nurse})$ asserts that Alice is a nurse, and $(Bob : \text{Manager})$ asserts that Bob is a manager. $\text{Nurse} \sqsubseteq \text{Caregiver}$ expresses that every nurse is a caregiver.
- HasSupervisor: This is a role relating two individuals. For example, $((Alice, Bob) : \text{HasSupervisor})$ asserts that Alice has a supervisor named Bob.

The following introduces the model that interprets the formulas in Definition 2.

Definition 4 (First-order Temporal Structure). A first-order temporal structure \mathcal{H} is a tuple $\langle \Delta, \mathcal{I}, h_1 h_2 \dots h_n \rangle$, where Δ is a nonempty set that forms the domain, \mathcal{I} is a knowledge interpretation, and $h_1 h_2 \dots h_n$ is a finite sequence of first-order interpretations. The interpretation \mathcal{I} assigns every atomic concept C_i to a set $C_i^{\mathcal{I}} \subseteq \Delta$, every atomic role R to a set $R^{\mathcal{I}} \subseteq \Delta \times \Delta$, every constant c to an element $c^{\mathcal{I}} \in \Delta$, and every function F with n -arity to a function $F^{\mathcal{I}}$ from Δ^n to Δ , such that $F_i^{\mathcal{I}}(c_1^{\mathcal{I}}, \dots, c_n^{\mathcal{I}}) = c^{\mathcal{I}}$ whenever $F(c_1, \dots, c_n) = c$. Then \mathcal{I} is extended to the concepts as follows:

$$\begin{aligned} \top^{\mathcal{I}} &= \Delta & ; & & \perp^{\mathcal{I}} &= \emptyset \\ (\neg C)^{\mathcal{I}} &= \Delta \setminus C^{\mathcal{I}} & ; & & (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \end{aligned}$$

$$\begin{aligned}
(\forall R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.((a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}})\} \\
(\exists R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists b.((a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}})\}
\end{aligned}$$

Each first-order interpretation h_i ($1 \leq i \leq n$) assigns an n -ary predicate symbol P to a relation P^{h_i} over Δ^n .

The reader will recall that we have partitioned all information into static knowledge and dynamic data. Therefore the knowledge interpretation \mathcal{I} interprets the fragment of the language that does not change over time, i.e., concepts, roles, constants and functions; and the first-order interpretation h_i interprets the rest of the language.

The following defines a relation ' \models ' between a first-order temporal structure and a knowledge base, which is sufficient to ensure that the two are consistent.

Definition 5 (Knowledge Base Satisfaction). Given a first-order temporal structure $\mathcal{H} = \langle \Delta, \mathcal{I}, h_1 h_2 \cdots h_n \rangle$ and a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we say that \mathcal{H} satisfies \mathcal{K} , written as $\mathcal{H} \models \mathcal{K}$, if the following hold:

- for all $C \sqsubseteq D$ in \mathcal{T} , $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and for all $C \equiv D$ in \mathcal{T} , $C^{\mathcal{I}} = D^{\mathcal{I}}$.
- for all $R \sqsubseteq S$ in \mathcal{T} , $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$, and for all $R \equiv S$ in \mathcal{T} , $R^{\mathcal{I}} = S^{\mathcal{I}}$.
- for all $t : C$ and $(t_1, t_2) : R$ in \mathcal{A} , $t^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(t_1^{\mathcal{I}}, t_2^{\mathcal{I}}) : R^{\mathcal{I}}$

A first-order temporal structure $\mathcal{H} = \langle \Delta, \mathcal{I}, h_1 h_2 \cdots h_n \rangle$ can be derived as follows.

1. The Δ consists of a finite set of strings that is used to specify the data contents;
2. The interpretation \mathcal{I} is specified according to the setting of care, so that $\mathcal{H} \models \mathcal{K}$; this makes sure that the interpretation is consistent with the knowledge base;
3. The sequence of first-order interpretation $h_1 h_2 \cdots h_n$ is derived from a healthcare data model (Definition 1). Suppose h_i is associated with the i -th record m_i in a corresponding healthcare data model \mathcal{M} . Thus the h_i s act as the states in a temporal system. Each predicate P that corresponds to a tag occurring in m_i is assigned to the set of data inside the tag, written as P^{h_i} , otherwise it is assigned to an empty set. For example, Fig. 3 shows a healthcare record m ; suppose m is associated with h , the tag `<medication>` is associated with a unary predicate P_1 , and no tag is associated with unary predicate P_2 , then $P_1^h = \{Med1, Med2\}$, $P_2^h = \{\}$.

Definition 6 (Semantics). Given a first-order temporal structure $\mathcal{H} = \langle \Delta, \mathcal{I}, h_1 h_2 \cdots h_n \rangle$, and a knowledge base \mathcal{K} that is satisfied by \mathcal{H} , the sentences in **FO-LTL-K** are interpreted as follows.

- $\mathcal{H}, h_i \models t_1 = t_2$ iff $t_1^{h_i} = t_2^{h_i}$;
- $\mathcal{H}, h_i \models P(t_1, \dots, t_n)$ iff $(t_1^{h_i}, \dots, t_n^{h_i}) \in P^{h_i}$;
- $\mathcal{H}, h_i \models \neg \varphi$ iff $\mathcal{H}, h_i \not\models \varphi$;
- $\mathcal{H}, h_i \models \varphi \wedge \psi$ iff $\mathcal{H}, h_i \models \varphi$ and $\mathcal{H}, h_i \models \psi$;
- $\mathcal{H}, h_i \models \mathbf{X}\varphi$ iff there exists h_{i+1} such that $\mathcal{H}, h_{i+1} \models \varphi$;
- $\mathcal{H}, h_i \models \varphi \mathbf{U} \psi$ iff there exists $j \geq i$ such that for all $i \leq k < j$, $\mathcal{H}, h_k \models \varphi$ and $\mathcal{H}, h_j \models \psi$;
- $\mathcal{H}, h_i \models c_1 : C$ iff $c_1^{\mathcal{I}} \in C^{\mathcal{I}}$;
- $\mathcal{H}, h_i \models (c_1, c_2) : R$ iff $(c_1^{\mathcal{I}}, c_2^{\mathcal{I}}) \in R^{\mathcal{I}}$;
- $\mathcal{H}, h_i \models \forall x. \varphi(x)$ iff for all $a \in \Delta$, $\mathcal{H}, h_i \models \varphi[a/x]$.

We now specify some of the norms in [3] using **FO-LTL-K**. Recall that the interpretation of norms depends on a local setting, each of which should have a reasonable time frame for actions to be completed. Therefore we consulted the caregivers in the Guysborough Antigonish Strait Health Authority (GASHA) for their interpretation of these norms.

- **Norm 1:** When language is a barrier, translators who understand the medical concepts and terminology facilitate information sharing.

This norm guides the second stage (*Information Sharing*) of a therapeutic encounter (see Fig. 1). When a patient is admitted into a palliative care program, some basic information, including the languages that he or she can speak, is entered into his or her healthcare record. This is captured in a unary predicate *PatientLanguage*. Suppose the official language of the palliative care program is kept in its knowledge base under the concept **OfficialLanguage**¹. Our specification of this norm takes time into consideration, requiring an action to take place within a constant time c , e.g., ‘3 hours’. It is represented as follows:

$$\begin{aligned} \forall t_1. (\neg \exists x. (PatientLanguage(x) \wedge x : OfficialLanguage) \wedge Time(t_1) \\ \rightarrow \mathbf{X}(\exists t_2. (Time(t_2) \wedge (t_2 - t_1) \leq c \wedge ActionTaken(d, p)))) \end{aligned}$$

where, t_1 refers to the time stamp of a ‘current’ record, t_2 refers to the time stamp of the ‘next’ record due to the temporal operator **X**, the binary predicate *ActionTaken*(x, y) denotes “action x is taken to y ”, constant d denotes an action “Find a translator”, constant p denotes a patient, and the predicate *Time* is interpreted as a singleton set of the unique time stamp associated to a record².

- **Norm 2:** The patient family’s understanding of the shared information is assessed regularly.

There are at least 7 norms requiring regular assessments. The gap between two assessments depends on numerous factors, such as the severity of the issues, the availability of resources, etc. In palliative care, patients’ families often play a very important role as many patients choose to receive care at home. Suppose the knowledge base already has the following role definition: $IsFamilyOf \equiv IsFatherOf \sqcup IsMotherOf \sqcup IsChildOf$, we can express “ x is a family member of y ” implicitly using $(x, y) : IsFamilyOf$ instead of using $(x, y) : IsFatherOf \sqcup IsMotherOf \sqcup IsChildOf$. This norm can be specified as “the next assessment should be made within a time duration c ”, and is represented as:

$$\begin{aligned} \forall t_1 \forall x. (Time(t_1) \wedge ((x, p) : IsFamilyOf) \rightarrow \mathbf{F}(\exists t_2. (Time(t_2) \\ \wedge (t_2 - t_1) \leq c \wedge (ActionTaken(e, x)))) \end{aligned}$$

where the temporal operator **F** refers to a ‘future’ record, constant e denotes “assessment of the understanding of shared information”, and constant p denotes a patient.

¹ In comparison, the predicate *PatientLanguage* is interpreted over healthcare records because different patients may speak different languages.

² If $Time^h = \{c\}$, where c is the content of the single time stamp in the record associated with h , then $\mathcal{H}, h \models \forall x. (Time(x) \rightarrow x = c) \wedge \exists x. Time(x)$.

5 Complexity

A data monitor constructs a first-order temporal structure from a healthcare data model and a knowledge base, then updates itself when a new healthcare record is received. It checks or monitors the properties specified in **FO-LTL-K**, and generates alerts when necessary. The core of our monitoring method is the model checking problem $\mathcal{H}, h_i \models \varphi$ in **FO-LTL-K**. We examine the computational complexity of this problem.

Computational complexity theory studies the amount of computational resources needed, such as time and storage, to solve a problem. It is well-known that the complexity of the model checking problem (also called the satisfaction checking problem) of first-order logic is PSPACE-complete [18]. We show that the model checking problem of **FO-LTL-K** has the same complexity by reducing it to a model checking problem of first-order logic, using a syntactic translation TRS and a semantic translation TRSM. The main idea is to, (1) let TRS associate a natural number i with a formula φ in **FO-LTL-K** so the temporal information is embedded in $\text{TRS}(\varphi, i)$; (2) let TRSM transform the $h_1 h_2 \cdots h_n$ into a single h by augmenting each h_i with a time parameter i semantically. We show this in detail.

Syntactic Translation TRS. The syntactic translation TRS recursively maps a **FO-LTL-K** expression and a natural number t to a first-order expression as follows:

- For each constant c and variable x , $\text{TRS}(c, t) := c$, $\text{TRS}(x, t) := x$; for each n-arity function and term t_i $\text{TRS}(F(t_1, \dots, t_n, t)) := F(\text{TRS}(t_1), \dots, \text{TRS}(t_n))$; for each concept symbol C_i , $\text{TRS}(C_i, t) := C_i$; for each role symbol R , $\text{TRS}(R, t) := R$;
- For each concept C, D and role R , $\text{TRS}(\neg C, t) := \neg \text{TRS}(C, t)$, $\text{TRS}(C \sqcap D, t) := \text{TRS}(C, t) \wedge \text{TRS}(D, t)$, $\text{TRS}(\forall R.C, t) := \forall y(R(x, y) \rightarrow \text{TRS}(C, t)(y))$, $\text{TRS}(\exists R.C, t) := \exists y(R(x, y) \wedge \text{TRS}(C, t)(y))$;
- For each predicate P and terms t_i , $\text{TRS}(P(t_1, \dots, t_n), t) := P(t_1, \dots, t_n, t)$; for each formula with concepts or roles: $\text{TRS}(u_1 : C, t) := \text{TRS}(C, t)(u_1)$, $\text{TRS}((u_1, u_2) : R, t) := R(u_1, u_2)$; for each formula $\forall x.\varphi$, $\text{TRS}(\forall x.\varphi, t) := \forall x.\text{TRS}(\varphi, t)$;
- For each formula with temporal modalities:
 - $\text{TRS}(\mathbf{X}(\varphi), t) := \exists t'((t' = t + 1) \wedge \text{TRS}(\varphi, t'))$;
 - $\text{TRS}(\varphi \mathbf{U} \psi, t) := \exists t'((t \leq t') \wedge \text{TRS}(\psi, t') \wedge \forall t''((t \leq t'') \wedge (t'' < t') \rightarrow \text{TRS}(\varphi, t'')))$;

Semantic Translation TRSM. The semantic translation TRSM maps a first-order temporal structure $\mathcal{H} = \langle \Delta, \mathcal{I}, h_1 \cdots h_n \rangle$ to a first-order structure $\text{TRSM}(\mathcal{H}) = \langle \Delta', h \rangle$:

- The domain $\Delta' = \Delta \cup \{1, \dots, n\}$;
- The interpretation h coincides with \mathcal{I} on the constants and function symbols. For the new predicates
 - The concept and role symbols are associated with unary predicates and binary predicates, and h uses the interpretation from \mathcal{I} , e.g., $t \in C_1^{\mathcal{I}}$ iff $t \in C_1^h$.
 - Suppose P_1 is an n-arity predicate in the **FO-LTL-K** and P_2 is the corresponding (n+1)-arity predicate in the first-order language that is being mapped to using TRS; we require $(t_1, t_2, \dots, t_n, k) \in P_2^h$ iff $(t_1, t_2, \dots, t_n) \in P_1^{h_k}$ in \mathcal{H} .

We can show the following by an induction on the structure of φ .

Theorem 1. *Given a first-order temporal structure \mathcal{H} and a **FO-LTL-K** sentence φ , we have: $\mathcal{H}, h_i \models \varphi$ iff $\text{TRSM}(\mathcal{H}, h_i) \models \text{TRS}(\varphi, i)$*

Since both translation functions TRS and TRSM are linear, and the model checking problem for first-order logic is PSPACE-complete [18], we have the following:

Theorem 2. *Given a first-order temporal structure \mathcal{H} and a **FO-LTL-K** sentence φ , the model checking problem: $\mathcal{H}, h_i \models \varphi$, is PSPACE-complete.*

This result is not surprising, but at least it shows that the model checking problem of this logic is not computationally too expensive (e.g., exponential). Based on Definition 6, it is straightforward to develop an algorithm for model checking, that is, to determine if $\mathcal{H}, h_i \models \varphi$. For details of proofs and algorithms, see [13], where the application of DL reasoners to enhance the efficiency of the model checking is also discussed.

6 Related & Future Work

A number of frameworks (e.g., Asbru, EON, GLIF, GUIDE) [8] are available for the computer interpretation of clinical guidelines (also called medical guidelines). Clinical guidelines aim at guiding decisions and establishing criteria for diagnosis, management, and treatment of specific medical problems. They are based on an examination of current evidence within the paradigm of evidence-based medicine. Palliative care norms, as we have discussed, are more general, and are meant to be interpreted relative to the context of particular care settings. Model checking techniques have been used to verify the correctness of a variety of guidelines and to monitor their executions [2, 4]. While we also apply model checking techniques, our focus is different from these approaches, as our logical language is designed both to specify the norms and to take the advantages of the existing knowledge bases.

In [9], the authors provided an agent-based alarm management system for a palliative care unit. They showed how intelligent agents can continuously monitor the evolution of the health status of palliative patients using two kinds of alarms: basic alarms (e.g., *(Hunger < 3)* and *Extreme weakness : Dangerous weakness*), and evolution alarms (e.g., *Number of evaluations: 2. Δ Weakness > 2 : Fast weakness increase*). We can specify these alarms in our language. Moreover, our data model and language are richer for temporal and knowledge-based reasoning.

In [5], the authors proposed a tableau-based algorithm for the runtime monitoring of workflow constraints. Sample properties taken from runtime monitoring scenarios were expressed using a first-order linear temporal logic. The main difference with our work is that knowledge bases are an integral part of our architecture, and we allow description logic expressions to facilitate the use of existing knowledge bases.

For further research, we plan to implement data-aware monitoring for palliative care and explore the use of autonomous agents [14] in monitoring.

Acknowledgment. We thank the Natural Sciences and Engineering Research Council of Canada, the Atlantic Canada Opportunities Agency, and GASHA for their support. We also thank the anonymous reviewers for their comments.

References

- [1] F. Baader, D. Calvanese, D. L. McGuinness, and D. N. and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge University Press, Cambridge, UK, 2003.
- [2] A. Bottrighi, L. Giordano, G. Molino, S. Montani, P. Terenziani, and M. Torchio. Adopting model checking techniques for clinical guidelines verification. *Artificial intelligence in medicine*, 48:1–19, 2010.
- [3] F. Ferris et.al. *A Model to Guide Hospice Palliative Care: Based on National Principles and Norms of Practice*. Canadian Hospice Palliative Care Association, 2002.
- [4] P. Groot, A. Hommersom, P. J. Lucas, R.-J. Merk, A. ten Teije, F. van Harmelen, and R. Serban. Using model checking for critiquing based on clinical guidelines. *Artificial intelligence in medicine*, 46:19–36, 2009.
- [5] S. Halle and R. Villemaire. Runtime monitoring of message-based workflows with data. In *Proceedings of the 2008 12th International IEEE Enterprise Distributed Object Computing Conference*, pages 63–72, USA, 2008. IEEE Computer Society.
- [6] K. Miller and W. MacCaull. Toward careflow management systems. *Journal of Emerging Technologies in Web Intelligence Special Issue, E-health: Towards System Interoperability through Process Integration and Performance Management*, 1(2):137–145, 2009.
- [7] K. Miller and W. MacCaull. Verification of careflow management systems with timed *BDI_{CTL}* logic. In *3rd International Workshop on Process-oriented Information Systems in Healthcare (ProHealth 09)*, 2009. Ulm, Germany.
- [8] Mor Peleg et. al. Comparing computer-interpretable guideline models: A case-study approach. *JAMIA*, 10:2003, 2002.
- [9] A. Moreno, D. Riano, and A. Valls. Agent-based alarm management in a palliative care unit. In *3rd Workshop on Agents Applied in Healthcare*, IJCAI, Edinburgh, 2004.
- [10] S. Panzarasa, S. Maddè, S. Quaglini, C. Pistarini, and M. Stefanelli. Evidence-based care-flow management systems: the case of post-stroke rehabilitation. *J. of Biomedical Informatics*, 35(2):123–139, 2002.
- [11] A. Pnueli. The temporal logic of programs. In *Proceedings of the Eighteenth IEEE Symposium on the Foundations of Computer Science*, pages 46–57, 1977.
- [12] F. Rabbi, H. Wang, and W. MacCaull. YAWL2DVE: An automatic translator for workflow verification. In *The 4th IEEE International Conference on Secure Software Integration and Reliability Improvement (SSIRI 2010)*, pages 53–59, 2010.
- [13] J. Ruan and W. MacCaull. Data-aware monitoring for healthcare workflows using formal methods, 2010. StFX CLI Technical Report, CLI-TR 01-2010.
- [14] J. Ruan, W. MacCaull, and H. Jewers. Enhancing patient-centered palliative care with collaborative agents. In *The Second International Workshop on Collaborative Agents REsearch and Development (CARE 2010)*, Toronto, Canada, 2010.
- [15] U. Sattler. A List of Description Logic Reasoners. <http://www.cs.man.ac.uk/~sattler/reasoners.html>. Last accessed May 07, 2010.
- [16] R. M. Smullyan. *First-order logic*. Dover Publications Inc., 1995.
- [17] W. van der Aalst and K. van Hee, editors. *Workflow Management: Models, Methods, and Systems*. The MIT Press, 2004.
- [18] M. Y. Vardi. The complexity of relational query languages (extended abstract). In *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 137–146, New York, NY, USA, 1982. ACM.