Answer Set Programming for Temporal Conformance Analysis of Clinical Guidelines Execution

Matteo Spiotta[1,2], Paolo Terenziani[1], and Daniele Theseider Dupré[1](✉)

[1] DISIT, Sezione di Informatica, Università del Piemonte Orientale, Vercelli, Italy
dtd@di.unipmn.it

[2] Dipartimento di Informatica, Università di Torino, Torino, Italy

Abstract. Analyzing conformance of execution traces with clinical guidelines is not a trivial task, because guidelines are developed for physicians who should always interpret them according to their general knowledge; their application to the specific conditions of a patient or a specific context cannot always be foreseen in the guideline specification. In this paper we consider the conformance problem not only for the sequence of action execution events, but also for their timing: the guideline may include temporal constraints for the execution of actions, and its adaptation to a specific patient and context may add or modify conditions and temporal constraints for actions. We propose an approach for analyzing execution traces in Answer Set Programming with respect to a guideline and Basic Medical Knowledge, pointing out discrepancies – including temporal discrepancies – with respect to the different knowledge sources.

1 Introduction

A Clinical Guideline (CG) is "a systematically developed statement to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" [7]. CGs are developed in order to capture medical evidence and to put it into practice, and deal with "typical" classes of patients, since the CG developers cannot define all possible executions of a CG on any possible specific patient in any clinical condition. When treating "atypical" patients, physicians have to resort to their Basic Medical Knowledge (henceforth BMK; informally, in this paper, BMK includes the different forms of medical knowledge that physicians have acquired during their studies and their clinical practice).

The interplay between CG and BMK recommendations can be very complex. For instance, actions recommended by a CG could be prohibited by the BMK, or a CG could force some actions despite the BMK discourages them (see, e.g., [3]). Such a complexity significantly increases in case the temporal dimension is taken into account: indeed, (i) temporal information is an intrinsic part of most CGs and BMK, and (ii) the interplay between CGs and BMK occurs in time. Regarding issue (i), actions have pre-conditions which temporally constrain them

This research is partially supported by Compagnia di San Paolo.

D. Riaño et al. (Eds.): KR4HC/ProHealth 2015, LNAI 9485, pp. 65–79, 2015.
DOI: 10.1007/978-3-319-26585-8_5

(e.g., an action must be performed only while a precondition holds), and may be temporally constrained with each other (e.g., in case of hip fracture, surgery is recommended within 36 h after admission). Considering (ii), though there are cases in which CG and BMK recommendations are in contrast, and one of the two prevails over the other, which is then ignored, in most cases the two recommendations should be "merged" along time. Typical cases are the treatment of exceptions [9,12,13], which, depending on the situation, may be treated as soon as they occur (thus suspending CG execution), delayed after the end of the CG, or executed concurrently with it. In all cases, some of the temporal constraints (in the CG and/or in the BMK) may be violated.

Unfortunately, the proper solution for managing the interplay between CGs and BMK is usually situation- and patient-dependent, and, in general, expecting that a model could provide such solutions is not realistic. Nevertheless, computer science can support physicians in the analysis of such an interplay, considering also patient data and contextual information, providing physicians with the relevant knowledge to understand and evaluate how the patient has to be treated. This is the challenging goal of the approach in this paper. In particular, as in [14], we explore the interplay between CGs and BMK from the viewpoint of *a posteriori* conformance analysis [10], intended as the adherence of an observed CG execution trace to both the CG and BMK. We do not provide any form of evaluation of how the interplay has been managed (i.e., whether the treatment was appropriate or not): we aim at identifying, in the trace, situations in which some recommendation (either in the CG or in the BMK) has not been followed, and at providing possible justifications for situations of non-conformance. In such a way, conformant treatments (to both the CG and the BMK) are automatically identified and presented to physicians, as well as non conformant treatments and their possible justifications, based on the available knowledge.

2 Input to the Framework

At least four different types of data/knowledge sources should be considered to analyze compliance: patient data, contextual data, CG model, and BMK.

We distinguish two types of **patient data**. We consider *patient findings*, i.e., data which are usually collected in patients' EHR. In particular, we assume that all data required during patient treatment are available, and that all such pieces of information are temporally tagged. We also assume to have a complete *log* of all the clinical actions executed on the patient, in which each occurrence of actions is temporally tagged. Specifically, we assume that the starting and ending time (both in case of completion and of abort) of all actions are recorded.

We consider **contextual data** such as personnel and resources availability.

Our approach is not biased towards any specific **CG formalism**; however, we will use the GLARE formalism [1,17] as a concrete example, due to its specific attention to the temporal aspects. Indeed, we consider the possibility of distinguishing between atomic and composite actions, and of specifying (therapeutic and diagnostic) decisions. CGs specify the control flow of actions and include

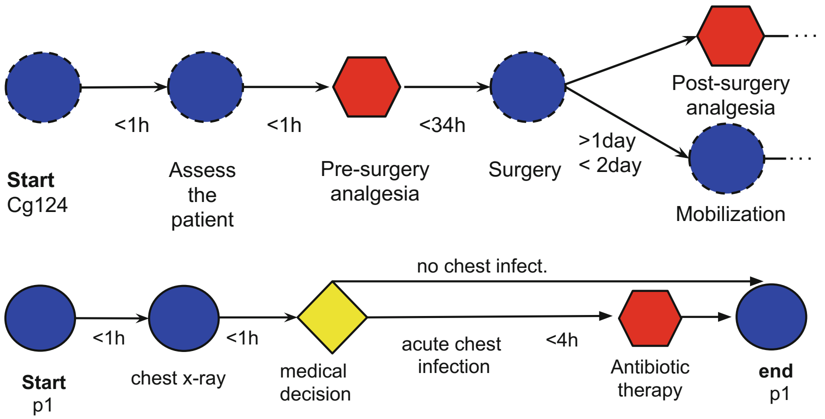

Fig. 1. Hip fracture CG (above) and chest infection plan (below). Circles are atomic actions, hexagonal nodes are composite actions, diamond nodes are decisions.

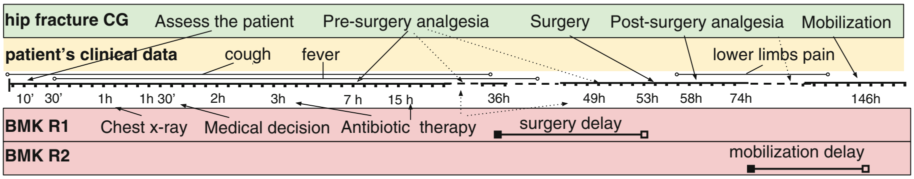

Fig. 2. Example case.

temporal constraints between them. Additionally, actions may have preconditions, and temporal constraints between the time when preconditions hold and the time when the related action must be executed can be specified. Specifically, in this paper (as in GLARE) we assume that temporal constraints may be used to impose a minimum and maximum delay between the starting/ending points of events (actions and/or preconditions in our context).

Figure 1 presents an adapted excerpt, represented in GLARE, of the guideline for hip fracture by the British National Institute for Health and Care Excellence (NICE) [6]. The CG contains information about recommended timing to ensure the effectiveness of the treatment. Since it recommends that surgery is performed "on the day of, or the day after, admission" we consider a "within 36 h" recommendation. Rehabilitation through mobilization therapy must start the day after surgery.

We assume that pieces of knowledge in the **BMK** are formed by:

- a *trigger,* i.e., conditions on the patient and context that make the piece of knowledge relevant, and either:
- a simple or composite action, which is suggested in the given conditions; or:
- knowledge (such as counterindications) suggesting to avoid or delay some action, in one of the following forms:

- *Avoid a:* states that action *a* should not be executed (we assume that such a statement is triggered by conditions that are not reversible);
- *Delay a while c:* states that action *a* should not be performed as long as *c* holds (where we expect *c* to be reversible);
- *Delay a for d:* suggests delaying action *a* for time *d*.

Knowledge involving suggested actions (**"do"** knowledge for short) is similar to exceptions in [9] and guideline-independent exceptions in [12], whose triggers are not expected to be only relevant at a specific point of a CG, but rather may occur – requiring treatment – at any point. Both "do" knowledge and the one that suggests avoiding or delaying an action (**"do not"** knowledge for short) refine rules proposed in [14], taking better into account the temporal dimension.

Then, while the CG provides treatment for general cases, the BMK may account both for additional actions and for cancellation or a different timing of actions prescribed by the CG (see Sect. 3.1).

The clinical case used to present our approach is the treatment of a patient which has been hospitalized to treat a hip fracture. At hospitalization time, the patient had also cough and high temperature. The following BMK rules R1 and R2 relevant to the case are considered:

R1. For patients with high body temperature and cough, the presence of a chest infection has to be investigated through a chest x-ray, and, if present, treated with an antibiotic therapy[1] (see Fig. 1).

R2. Mobilization has to be delayed for patients having pain in lower limbs.

Figure 2 describes an example case. Actions in the first row are directly recommended by the CG. However, several non-compliant actions (with respect to the CG) appear. Actions *chest x-ray*, *medical decision*, *antibiotic therapy* are justified by rule R1. The fact that another problem is being treated should also explain the delay of *hip surgery* beyond the 36 h recommended by the CG, even though we do not expect to have an explicit model of the condition which allows surgery to proceed. Rule R2 is triggered because the day after surgery the patient has pain in the lower limbs; it justifies the delay of *mobilization*.

3 Execution Model

In order to establish whether a trace is adherent to the guideline, the semantics of the guideline should be established. Possible states and transitions of a work action are as in Fig. 3. The control flow of the CG execution, or triggers in the BMK, may indicate that a given action has to be considered for execution (is a candidate). A candidate action could become active, i.e., actually started, or discarded

[1] Actually, the recommendations in [6] explicitly mention acute chest infection as one of the conditions to be checked and, if necessary, treated, to avoid delaying surgery too much; but they also mention, without detailing them, less common concerns which may require delaying surgery: we consider chest infection like one of these.

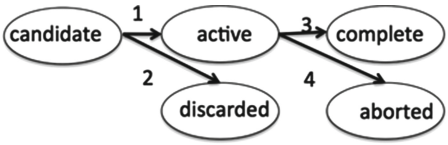

Fig. 3. States for a work action

(transitions 1 or 2); if active, it could either be completed or aborted (transitions 3 or 4). In order to define which transition could occur and when, we start discussing the case of a single source of knowledge, the CG, and, for the sake of brevity, we limit the discussion to transitions 1 and 2.

The control flow that makes a candidate imposes constraints on the time t_{act} when action a could become active. In general, such constraints are given with respect to the start or end times of previous actions in the control flow (possibly more than one action). In order to simplify the description in the following, we assume they are given with respect to the time t_{ca} when the action becomes candidate and are of the form:

$(i)\ t_{ca} + m \leq t_{act} \leq t_{ca} + n$

Then, $t_{ca} + n$ provides a deadline for starting the action in order to conform to the CG. In addition, preconditions on a impose constraints of similar form:

$(ii)\ t + m \leq t_{act} \leq t' + n$

where t and t' are the start or end time of a precondition, or, more precisely, of an episode of a precondition. Expressions $t' + n$ in constraints (ii) do not, in general, provide deadlines because the precondition could become true again[2]; e.g., a blood exam could be constrained to be executed within 1 day after the previous step (type (i)) and could require fasting for at least 8 h (type (ii)).

The conformant execution after an action becomes candidate can be characterized as follows:

1. The action should start (become active) at a time t_{act} such that all preconditions, with their temporal constraints, enable the action, if one such time exists.
2. Otherwise, when the first deadline is reached, the action is discarded.

The conditions above can be represented in first-order logic, using the following predicates:

- $preconditions(a, \bar{t})$ means that an episode for each precondition of a hold, and $\bar{t}$ are the start and end times of such episodes;

[2] In some cases there is a deadline, e.g. for a diagnostic action that should be performed within a given time after the *first* episode of a symptom. Such deadlines can be dealt with as the other ones, but, again, for simplicity, we ignore them in the following.

- $C(t_{ca}, t_{act}, \overline{t})$ represents the constraints between the time t_{ca} when the action becomes candidate, the times $\overline{t}$ of start and end of precondition episodes, and the time t_{act} when the action could become active;
- *trans* represents times of transitions and their binding for the same action instance; in particular, $trans(a, candidate, t_{ca})$ means that a becomes candidate at t_{ca}, while $trans(a, t_{ca}, active, t_{act})$ (with four arguments) means that the instance of a that became candidate at t_{ca} becomes active at t_{act}.

The correspondents of conditions (1–2) above are then:

$$\forall t_{act} t_{ca} trans(a, t_{ca}, active, t_{act}) \rightarrow \exists \overline{t}\ preconditions(a, \overline{t}) \wedge \\ trans(a, candidate, t_{ca}) \wedge C(t_{ca}, t_{act}, \overline{t}) \tag{1a}$$

$$\forall t_{ca} trans(a, candidate, t_{ca}) \wedge [\exists t_{act} \overline{t}\ preconditions(a, \overline{t}) \wedge C(t_{ca}, t_{act}, \overline{t})] \\ \rightarrow \exists ! t_{act}\ trans(a, t_{ca}, active, t_{act}) \tag{1b}$$

$$\forall t_{ca}\ trans(a, candidate, t_{ca}) \wedge [\nexists t_{act} \overline{t}\ preconditions(a, \overline{t}) \wedge C(t_{ca}, t_{act}, \overline{t})] \\ \leftrightarrow trans(a, t_{ca}, discarded, t_{ca} + n) \tag{2}$$

Formula (1a) states that if the action starts at t_{act}, it is allowed to start at such time; (1b) states that if there are times when a candidate action is allowed to start, it starts (at one of those times, due to (1a)). Formula (2) states that the action is discarded (at time $t_{ca} + n$, i.e., at the deadline) if and only if there is no allowed time to start it.

Such formulas, and similar ones for the other transitions, can be seen as a way of providing semantics for the CG formalism, and are the basis for detecting (non-)compliance of a sequence of events with the guideline; in fact, a non-compliance corresponds to the fact when one of such formulas is false, and types of non-compliance correspond to different ways such formulas can be false.

Item 2 and Formula (2) correspond to a strict interpretation of the CG recommendations. If not all conditions for starting the action can be met, there may be valid medical alternatives to discarding the action: either relaxing a precondition or a deadline, i.e., performing the action late, or even performing the action at a time when a precondition recommends not to. We therefore point out the occurrence of case 2, and for a non-conformance case where a deadline or a precondition is violated, it could be pointed out whether performing the action meeting *all* the recommendations was possible or not.

3.1 Taking BMK into Account

In the following we describe the alterations of the guideline execution that can possibly be considered justified, taking BMK into account. We identify several situations where CG and BMK recommendations could be merged, or, in case they are in contrast, one could override the other.

When a (possibly composite) action a in the *"do"* BMK knowledge is triggered, several **modalities** of execution are considered possible (similarly to [12]):

- execution of a and the CG proceeds concurrently, according to their own constraints (**concur** modality);
- a and the CG are executed concurrently, but temporal constraints are not enforced (since they are presumably given for the case where there is no concurrent treatment for another problem) (**concur_no_tc**); as special cases, a is delayed after the end of the CG execution (**after**), or a is executed first, and then the CG execution proceeds (**before**);
- the CG execution continues, and the BMK suggestion is ignored (**ignore**);
- the execution of the CG is aborted and a is executed (**abort**).

A special case occurs for the concurrent modality, in case the same action b is candidate for both CG execution and the execution of the BMK action: temporal constraints from both the CG and BMK (**cg_bmk_constr**), or from either of them (**cg_constr, bmk_constr**), may be enforced.

When an action a is executable or active and a *"do not"* BMK rule is triggered, the options are as follows, depending on what is triggered:

- *Avoid a:* the action a is discarded, if candidate, or aborted, if active (**avoid**); or the BMK rule is ignored (**ignore_avoid**);
- *Delay a while c:* either the BMK rule is ignored (**ignore_delay**), or c is used as an additional precondition for action a (**add_delay**), or it replaces preconditions for a (**delay**).
- *Delay a for d:* either the BMK rule is ignored (**ignore_delay**), or it adds the constraint $t_{now} + d \leq t_{act}$ (**add_delay**), or replaces with it the constraints on t_{act}(**delay**).

This accounts for a wide range of modifications to the set of executions allowed by a CG, taking into account knowledge, the BMK, that may be *general*, not being related to the class of problems addressed by the CG, as well as *specific*, justifying adaptation of the CG to a specific class of patients, which is not explicitly considered in the CG definition. It allows one of the knowledge sources to prevail over the other one (which is then ignored), or for its recommendations to be given temporal priority on the other ones. The result can only approximate the set of medically correct adaptations of a CG to a case; it is a way of making conformance analysis more flexible, without assuming exhaustivity of CGs.

4 Conformance Analysis

The goal of conformance analysis in the context described in the previous section is to reconstruct whether the sequence of events in the log can be interpreted as an execution of the CG, with the possible alterations that take BMK into account. If an exact reconstruction is not possible, discrepancies with the log must be pointed out. However, in case a BMK rule is triggered, the alternatives in Sect. 3.1 introduce several potential reconstructions, also because some the modalities are (logically) stronger than others. For example, in a "do" rule, the *concur*, *after* and *before* modalities are strictly stronger than *concur_no_tc*; then, if one of the former is consistent with the log, also *concur_no_tc* is. Similarly, *add_delay* is stronger than *delay* and *ignore_delay*. With different data, a weak modality may be consistent, while a stronger modality is not, i.e., it implies some discrepancies. In general, especially in case several BMK rules are triggered, this gives rise to several potential solutions, each one with zero or more discrepancies.

We introduce therefore a notion of preference among explanations, where, as a primary criterion, we prefer reconstructions with a minimum number of discrepancies with the log. Secondarily, i.e., among explanations with the same number of discrepancies, we prefer explanations that are conformant with more knowledge. In the first case mentioned above, we prefer *concur* over *after* and *before* (even though it is not logically stronger) since *concur* means executing the CG and the BMK plan respecting all their constraints; and we prefer *after* and *before* over *concur_no_tc*, since the former impose to respect the internal quantitative constraints of each plan, while the latter only imposes that the control flow of each plan is followed. The *add_delay* modality is preferred over *delay* and *ignore_delay*, since it corresponds to being conformant with more knowledge.

These preferences are not intended as medically preferred choices, but rather as a way for not presenting explanations that unnecessarily assume a deviation from prescriptions of the overall knowledge. The modality is part of the solution; this is useful especially in cases where being conformant with all knowledge is not possible, i.e., two pieces of knowledge provide contrasting prescriptions, and it is not specified which one should prevail. Consider the case where, according to a BMK rule, an action was actually delayed, overriding the CG constraints: in the interpretation of the log there is evidence of this choice, even though it was not necessarily the best medical choice. On the other hand, for a non-conformant action, in case a conformant execution was also possible, this is pointed out.

In order to perform such conformance analysis, formulae in Sect. 3 (i.e. including (1a, 1b and 2)) could be the basis for analyzing conformance of a trace with respect to a CG only. In fact, it should be detected whether some of the formulae are false for some action. E.g. (1b) is violated if a should be started, but it is not. In the negation of formula (1a) we could distinguish several cases: the action is started, but either it was not candidate, or some of its precondition is never true, or for the times when the action changes state, there are no episodes of the preconditions such that the constraints hold.

The set of formulae could be further elaborated in order to interpret traces based on the BMK as well as the CG: the actual preconditions and temporal constraints to enforce depend on the knowledge source(s) being considered, and, e.g., in (1a), the executed action may fail to be a CG candidate, but be a BMK candidate. We do not, however, describe in detail this option; rather, we describe in the following how the approach is represented in Answer Set Programming (ASP) [8], which is also based on logic; similarly to [14], this allows an ASP solver to infer whether and how the sequence of events in the log can be reconstructed according to the patient DB, the CG and the BMK, following the specifications in Sect. 3. While a SAT solver finds models of a propositional representation, an ASP solver (in particular, we used Clingo [18]) finds stable models of an ASP representation,

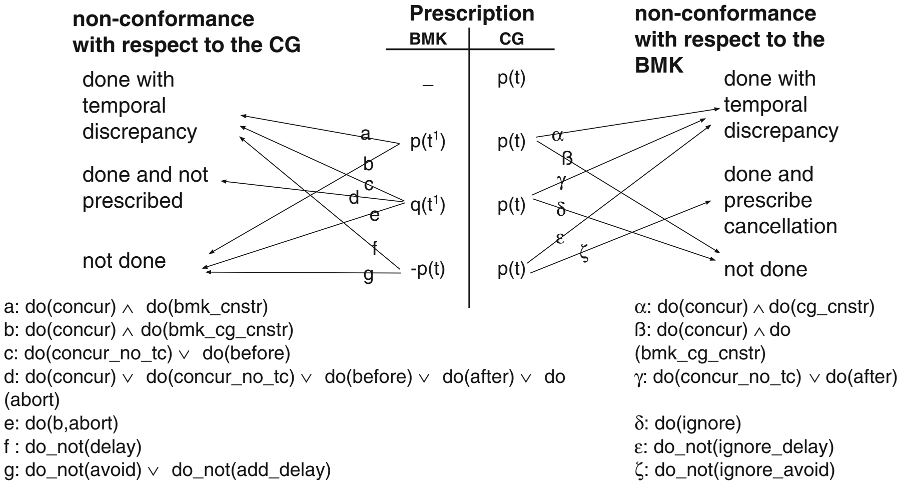
non-conformance with respect to the CG
Prescription
BMK
CG
non-conformance with respect to the BMK
done with temporal discrepancy
done and not prescribed
not done
_
p(t)
p(t^1)
p(t)
q(t^1)
p(t)
-p(t)
p(t)
a
b
c
d
e
f
g
α
ß
γ
δ
ε
ζ
done with temporal discrepancy
done and prescribe cancellation
not done
a: do(concur) ∧ do(bmk_cnstr)
b: do(concur) ∧ do(bmk_cg_cnstr)
c: do(concur_no_tc) ∨ do(before)
d: do(concur) ∨ do(concur_no_tc) ∨ do(before) ∨ do(after) ∨ do (abort)
e: do(b,abort)
f : do_not(delay)
g: do_not(avoid) ∨ do_not(add_delay)
α: do(concur) ∧ do(cg_cnstr)
ß: do(concur) ∧ do (bmk_cg_cnstr)
γ: do(concur_no_tc) ∨ do(after)
δ: do(ignore)
ε: do_not(ignore_delay)
ζ: do_not(ignore_avoid)

Fig. 4. Discrepancies and their justifications.eps

which is nonmonotonic and allows rules with variables, which are substituted in a grounding phase with a finite number of constants (in our case, for example, variables for time instants take values in a finite domain of time points). Annotation rules are used to identify cases of non-conformance with at least one source of knowledge, which include cases where formulae in Sect. 3 are false. The nonmonotonic nature of ASP is useful to model exceptions to guideline execution according to the BMK. Optimization statements in ASP are used to represent preferences.

Figure 4 summarizes an analysis which contributes to demonstrating coverage of our approach. It represents cases where a discrepancy with one knowledge source may be justified by the other source. Different lines in the table provide different cases as regards current prescriptions from the CG and the BMK ($p(t)$ means that action p is candidate to start at time t, while $-p(t)$ means that p should not be done at t). Arrows connect a line to a type of discrepancy with respect to the CG or the BMK, and labels below provide a shorthand description of what will be part of the answer set in that case. For example, if the CG and BMK propose the same action as candidate (2nd line), it may be the case (a) that the action is performed without conforming to the CG constraints (the exception runs in concurrent modality and only the bmk constraints are enforced), or (α) it is performed without conforming to the BMK constraints, or (b and β) it is not performed because all constraints are enforced, but they are never all true.

4.1 ASP Representation

The ASP model can be divided in the following main components:

- the **inputs** for the analysis: the log trace, the CG and BMK model;
- the **control flow component** which provides the set of allowed actions in each state (considering both the CG and the BMK);

- the **interaction component** which generates the different modalities of execution for the fired BMK rules (i.e., "concur","before", etc.);
- the **temporal constraints component** which determines the allowed timing of actions with respect to other actions and action preconditions;
- the **annotation rules component** which detects non compliance, and selects BMK execution scenarios which minimize unexpected behaviour.

The ASP model reconstructs for each state the behaviours allowed by the execution model. The sequence of actions in the log, together with patient and context data, makes it possible to identify the paths in the control flows (of the CG and the BMK plans) followed during execution. When a triggering condition is satisfied by context and patient data, a candidate interpretation (an answer set) is generated for each possible modality of rule execution. Preconditions and temporal constraints are evaluated taking into account the actions specification, with variations imposed by the BMK rules in the specific execution context. The annotation rules use this information to detect behaviours that differ from the expected one in each candidate interpretation.

The encoding of **inputs** is straightforward. Patient and context data are represented with ground facts (`holds(Name,Value,Ts,Te)`) binding data to intervals in which they were known to be true. The GLARE representation of CG and BMK plans is encoded with ground facts which describe the control flow, preconditions and time constraints. BMK trigger rules can be mapped to a set of ASP rules having the trigger condition as body and one of the following as head:

- `prescribe(ID,A,T)` for a BMK rule ID prescribing the composite action A at time T;
- `prescribeAvoid(ID,A,T)` for a BMK rule ID prescribing the discard/abort of an atomic action A at time T;
- `prescribeDelayWhileC(ID,C,A,T)` for a BMK rule ID prescribing at time T the delay of A until condition C becomes true.
- `prescribeDelay(ID,C,D,A,T)` for a BMK rule ID prescribing at time T the delay of A for time D given that condition C is true.

For example, the ASP encoding of R2 is:

```
prescribeDelayWhileC(r2,lower_limb_pain,mobilization,T):-
        holds(lower_limb_pain,Ts,Te),T>Ts,T<Te,candidate(mobilization,T).
```

The **control flow component** evaluates the control flow of CG and BMK plans. Similarly to [14], given the executed action and the control flow description, instances of `candidate(Src,A,Tca,T)` are inferred, which state that knowledge source `Src` (either `cg`, or the BMK rule identifier) enables the execution of `A` in `T` (the time `Tca` when it became candidate identifies an action instance).

The **temporal constraints component** reconstructs the expected timing of the actions taking in account the possible variations introduced by the BMK. A predicate `tc(Src,Tca,A,started/completed,Ts,Te)` specifies the allowed interval `[Ts,Te]` for starting or completing action `A` prescribed by knowledge source `Src` at `Tca`. In general we deal with STP constraints for actions in a group, but, for the sake of brevity, we only show the case of temporal constraints given for the starting point wrt the end of previous action in the control flow:

```
tc(Src,Tca,A,started,Tca+Min,Tca+Max):-
   wf_tc(Src,A,started,B,completed,Min,Max),
   candidate(Src,A,Tca,_),trans(B,completed,Tca),
   not exception(Src,A,Tca,deleteWFtc;changeWFtc).
```

where `wf_tc` states that `A` must be started between `Min` and `Max` instants after `B` was completed; `candidate` and `trans` detect the action `B` that enabled `A` at time `Tca`; `exception`, defined in the BMK component, could prescribe variations to the CG time constraints, or cancel the CG constraints. If no temporal variations are required, the action should be performed in the interval `Tca+Min,Tca+Max`; otherwise the BMK component (below) either cancels the constraints (`deleteWFtc`), or it suggest a different timing (`changeWFtc`). In this last case `tc` is defined in the BMK component and the interval derived from the CG specification is used to check whether being conformant with both knowledge sources is possible. For the interval determined by the constraints in force, action preconditions are checked to detect whether the action should be started or discarded.

The **BMK component** determines the execution modality of BMK rules and their consequences. Some ASP rules are shown below for the "delay while condition" case; rule 1 generates an answer set for each of the three modalities (**delay, add_delay, ignore**). Rule 2 adds the temporal constraints for the cases **delay** and **add_delay**, allowing the execution of `A` after the end of the condition (and until an upper bound for the time scale). Rule 3 blocks the enforcement of temporal constraints provided in the action description (see the temporal constraints component). For the ignore modality nothing has to be changed.

```
1: 1{do_not(Tca,ID,A,delay;add_delay;ignore_delay)}1:-
       prescribeDelayWhileC(ID,C,A,T),candidate(Src,A,Tca,T).
2: tc(cg,T,A,started,Te,M):-1{do_not(T,ID,A,add_delay;delay)}1,
       prescribeDelayWhileC(ID,C,A,T),max(M),holds(C,Ts,Te),T>=Ts,T<=Te.
3: exception(cg,A,Tca,changeWFtc):-do_not(Tca,ID,A,delay).
```

The case of BMK rules prescribing the introduction of actions is similar: if `prescribe(ID,A,T)` becomes true, an answer set is generated for each execution modality reproducing the different behaviours: e.g. the **concurrent** modality enables the prescribed action, the **abort** modality enables the action and blocks all the CG candidate actions, and so on.

The **annotation rules component** consists of rules which identify discrepancies between the log trace and the different behaviours considered in the answer sets. The discrepancies are violations of the execution model, e.g. an action which is executed and not prescribed is detected by:

```
discr(action_executed_not_candidate,A,Tact):-
    started(A,Tact), not candidate(src,A,Tca,Tact).
```

and this violates formula (1a) in Sect. 3. An action executed too late, also violating (1a) because temporal constraints are not respected, is detected by:

```
discr(late,A,Tca,Tact,TcS,TcE):-candidate(Src,A,Tca,Tact),
    started(A,Tact),tc(Src,Tca,A,started,TcS,TcE),Tact>TcE.
```

ASP optimization statements are used to select the answer set with the smallest number of discrepancies, and, secondarily, to prefer conformance with more knowledge (e.g., instances of `do_not(_,_,_,_,addDelay)` are maximized).

In the example, R2 is triggered and three answer sets are generated. Some facts for each of them are shown below; e.g., in the 2nd and 3rd one (where the CG constraint is enforced) `tc(cg,55,mobilization,started,79,103)` is made true by the temporal constraints component, given that surgery ends at time 55 and mobilization has to be started 24–48 h after it. The other instance of `tc` in the 1st and 2nd answer set is due to the BMK component. Note that the two intervals have no intersection, then the solver selects the first solution, which has no discrepancies. In case of compatible constraints, if the action was started consistently with both of them, the solution with `add_delay` would be preferred.

```
1: do_not(55,r2,mobilization,delay),
   tc(cg,55,mobilization,started,140,10000),
   candidate(cg,mobilization,55,55..146)
2: do_not(55,r2,mobilization,add_delay),
   tc(cg,55,mobilization,started,140,10000),
   tc(cg,55,mobilization,started,79,103),
   discr(late,mobilization,55,146,79,103),
   candidate(cg,mobilization,55,55..146)
3: do_not(55,r2,mobilization,ignore),
   tc(cg,55,mobilization,started,79,103),
   discr(late,mobilization,55,146,79,103),
   candidate(cg,mobilization,55..146).
```

5 Conclusions and Related Work

CGs do not include all knowledge that physicians have to take into account when treating patients, since patient states and contexts of execution cannot always be foreseen in the definition of the guideline. In this paper we propose an approach for analyzing temporal conformance of execution traces with respect to a richer form of medical knowledge, which may be used to justify deviations from a strict application of the guideline, both as regards extra actions for situations that are not foreseen (and whose treatment may alter the timing of guideline execution), and cancellation or delay of actions that are prescribed by the guideline, when there are reasons to do so. Given that we do not assume that all exceptions and interactions are modeled, interpretations provided by the approach can only be an approximation of medically correct executions. For example, in the absence of specific knowledge, the approach assumes that a temporally non-conformant execution of a guideline for treating a patient condition is always potentially justified in case another treatment, triggered by the BMK, is being executed.

Several approaches in the literature have addressed some of the issues in our proposal, or related ones. One of such issues is the verification of properties of clinical guidelines, i.e., proving that some properties hold for all executions of a guideline, in order to improve quality of guidelines. In the Protocure project, theorem proving techniques have been adopted for verification [15]: a medical protocol is modeled in the Asbru language and mapped to a specification in KIV, an interactive theorem prover. Properties are expressed in a variant of Interval Temporal Logic. Model checking techniques have been proposed in the Protocure II and GLARE projects [2,4]. Guidelines are automatically translated into a language for formal verification (SVN in Protocure II, Promela in GLARE), properties are specified in a temporal logic (ACTL and LTL respectively) and verified through model checking engines (SVN and SPIN respectively).

The integration of CGs with general medical knowledge has been considered in some case (see e.g. [16]) using such knowledge as a source of definitions of clinical terms and abstractions. In Medintel [5] different medical information sources (e.g., guidelines, reference texts, scientific literature) are used to improve decision support and the quality of care provided by general practitioners. The approach in [11] considers different forms of general medical knowledge in the context of CG verification: knowledge on the physiological mechanisms underlying the disease, and knowledge concerning good practice in treatment selection (leading to quality requirements), assuming the availability of a preference relation between treatments. Verification is used in order to check whether, given a class of patients, and considering physiological knowledge, the CG achieves a set of intentions, satisfying the quality requirements.

A limited number of approaches have dealt with verifying conformance of a trace of actions with recommendations in a CG. In [10], differences between actual actions and CG prescriptions are detected and analyzed, e.g. by comparing, for a non-compliant actions, actual findings with findings that support the action according to the CG. However, neither general medical knowledge nor quantitative time are considered in such a work. On the other hand, [3] focuses on the interaction between clinical guidelines (CGs) and the basic medical knowledge (BMK) in the light of the conformance problem. Our approach presents several similarities to it, but it is based on ASP, and, more importantly, it does not assume a fixed model for CG-BMK interactions. Indeed, different forms of interactions may be pointed out by our analysis, grounded on the different modalities we have identified. Also, the approach in this paper is an in-depth extension of the proposal in [14], addressing the temporal dimension.

Finally, our proposal is related to work about the treatment of CG *exceptions* [9,12,13], i.e., conditions that may suddenly arise, and whose treatment is not directly foreseen in the CGs. Indeed, part of the BMK we consider in our approach regards such conditions, and some of the modalities for CG-BMK interactions have already been identified in the context of exception handling.

Acknowledgement. The authors are grateful to Gianpaolo Molino for his valuable advice on the medical issues in the paper.

References

1. Anselma, L., Terenziani, P., Montani, S., Bottrighi, A.: Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. Artif. Intell. Med. **38**(2), 171–195 (2006)
2. Bäumler, S., Balser, M., Dunets, A., Reif, W., Schmitt, J.: Verification of medical guidelines by model checking – a case study. In: Valmari, A. (ed.) SPIN 2006. LNCS, vol. 3925, pp. 219–233. Springer, Heidelberg (2006)
3. Bottrighi, A., Chesani, F., Mello, P., Montali, M., Montani, S., Terenziani, P.: Conformance checking of executed clinical guidelines in presence of basic medical knowledge. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part II. LNBIP, vol. 100, pp. 200–211. Springer, Heidelberg (2012)
4. Bottrighi, A., Giordano, L., Molino, G., Montani, S., Terenziani, P., Torchio, M.: Adopting model checking techniques for clinical guidelines verification. Artif. Intell. Med. **48**(1), 1–19 (2010)
5. Brandhorst, C.J., Sent, D., Stegwee, R.A., van Dijk, B.M.A.G.: Medintel: decision support for general practitioners: a case study. In: MIE 2009, pp. 688–692 (2009)
6. British National Institute for Health and Care Excellence. Hip fracture: The management of hip fracture in adults. http://www.nice.org.uk/guidance/cg124
7. Field, M., Lohr, K. (eds.): Guidelines for Clinical Practice: From Development to Use. National Academy Press, Washington, D.C (1992). Institute of Medicine
8. Gelfond, M.: Answer sets. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation, Chap. 7. Elsevier, Amsterdam (2007)
9. Grando, A., Peleg, M., Glasspool, D.: A goal-oriented framework for specifying clinical guidelines and handling medical errors. J. Biomed. Inform. **43**(2), 287–299 (2010)
10. Groot, P., Hommersom, A., Lucas, P.J.F., Merk, R., ten Teije, A., van Harmelen, F., Serban, R.: Using model checking for critiquing based on clinical guidelines. Artif. Intell. Med. **46**(1), 19–36 (2009)
11. Hommersom, A., Groot, P., Lucas, P.J.F., Balser, M., Schmitt, J.: Verification of medical guidelines using background knowledge in task networks. IEEE Trans. Knowl. Data Eng. **19**(6), 832–846 (2007)
12. Leonardi, G., Bottrighi, A., Galliani, G., Terenziani, P., Messina, A., Della Corte, F.: Exceptions handling within GLARE clinical guideline framework. In: AMIA 2012, American Medical Informatics Association Annual Symposium (2012)
13. Quaglini, S., Stefanelli, M., Lanzola, G., Caporusso, V., Panzarasa, S.: Flexible guideline-based patient careflow systems. Artif. Intell. Med. **22**(1), 65–80 (2001)
14. Spiotta, M., Bottrighi, A., Giordano, L., Dupré, D.T.: Conformance analysis of the execution of clinical guidelines with basic medical knowledge and clinical terminology. In: Miksch, S., Riano, D., Teije, A. (eds.) KR4HC 2014. LNCS, vol. 8903, pp. 62–77. Springer, Heidelberg (2014)

15. ten Teije, A., Marcos, M., Balser, M., van Croonenborg, J., Duelli, C., van Harmelen, F., Lucas, P.J.F., Miksch, S., Reif, W., Rosenbrand, K., Seyfang, A.: Improving medical protocols by formal methods. Artif. Intell. Med. **36**(3), 193–209 (2006)
16. ten Teije, A., Miksch, S., Lucas, P. (eds.): Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, Studies in Health Technology and Informatics, vol. 139. IOS Press, Amsterdam (2008)
17. Terenziani, P., Molino, G., Torchio, M.: A modular approach for representing and executing clinical guidelines. Artif. Intell. Med. **23**(3), 249–276 (2001)
18. The Potsdam Answer Set Solving Collection. http://potassco.sourceforge.net/