

Conformance Analysis of the Execution of Clinical Guidelines with Basic Medical Knowledge and Clinical Terminology

Matteo Spiotta^{1,2}, Alessio Bottrighi¹, Laura Giordano¹,
and Daniele Theseider Dupré¹ (✉)

¹ DISIT, Sezione di Informatica,

Università del Piemonte Orientale, Alessandria, Italy

{alessio,dtd}@di.unipmn.it

² Dipartimento di Informatica, Università di Torino, Turin, Italy

Abstract. Clinical Guidelines (CGs) are developed for specifying the “best” clinical procedures for specific clinical circumstances. However, a CG is executed on a specific patient, with her peculiarities, and in a specific context, with its limitations and constraints. Physicians have to use Basic Medical Knowledge (BMK) in order to adapt the general CG to each specific case, even if the interplay between CGs and the BMK can be very complex, and the BMK should rely on medical terminological knowledge. In this paper, we focus on *a posteriori* analysis of conformance, intended as the adherence of an observed execution trace to CG and BMK knowledge. A CG description in the GLARE language is mapped to Answer Set Programming (ASP); the BMK and conformance rules are also represented in ASP. The BMK relies on the SNOMED CT terminology and additional (post-coordinated) concepts. Conformance analysis is performed in Answer Set Programming and identifies non-adherence situations to the CG and/or BMK, pointing out, in particular, discrepancies from one knowledge source that could be justified by another source, and discrepancies that cannot be justified.

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” [1]. The CGs are developed in order to capture medical evidence and to put it into practice, and deal with general classes of patients, since the CG developers (typically expert committees) cannot define all possible executions of a CG on any possible specific patient in any possible clinical condition. CG developers make some implicit assumptions:

1. the CG is applied to an *ideal patient*, i.e., patients have just the single disease considered in the CG (thus excluding the concurrent application of more than one CG), and are statistically relevant (they model the typical patient affected by the given disease), not presenting rare peculiarities or side-effects;

This research is partially supported by Compagnia di San Paolo.

2. the CG is applied in an *ideal context*, i.e., in the context of execution, all necessary resources are available;
3. *ideal physicians* are executing the CG, i.e., physicians whose knowledge always allow them to properly apply the CGs to specific patients.

On the other hand, when a CG is applied to a specific patient, the patient and/or the context may not be ideal. The physicians indeed exploit Basic Medical Knowledge (BMK) to adapt the CG to the specific case at hand. The interplay between these two types of knowledge can be very complex, e.g., actions recommended by a CG could be prohibited by the BMK, or a CG could force some actions despite the BMK discourages them. Thus the physicians' judgment is very important in order to correctly execute a given CG in a specific case, as observed by the Infectious Diseases Society of America in its *Guide to Development of Practice Guidelines* [2]: "Practice guidelines, however, are never a substitute for clinical judgment. Clinical discretion is of the utmost importance in the application of a guideline to individual patients, because no guideline can ever be specific enough to be applied in all situations."

The issue of studying the interplay between the knowledge in CGs and BMK is relatively new in the literature. Several approaches have focused either on CGs or BMK in isolation, or have considered the BMK only as a source of information, such as definitions of clinical terms and abstractions [3]. Only recently some approaches (e.g., [4,5]) have considered that CGs cannot be interpreted and executed in "isolation", since CGs correspond to just a part of the medical knowledge that physicians have to take into account when treating patients. In this paper, we explore the interaction between CGs and BMK from the viewpoint of conformance analysis, intended as the adherence of an observed CG execution trace to both types of knowledge. Observe that both CG knowledge and BMK can be defeated (for a more detailed discussion see [4]), and it is, in general, the physician's responsibility to assess whether a trace can be deemed as conformant.

Our goal is to support the physicians in the conformance analysis task, providing them as much information as possible to make this task easier. The approach is based on GLARE ([6] and Sect. 2) to represent CGs, and on SNOMED CT ([7] and Sect. 3) for medical terminology; our general framework is described in Sect. 4 and its representation in Answer Set Programming in Sect. 5. In particular, we provide a set of rules defining, on the one hand, discrepancies from one source of knowledge that are, at least potentially, justified by another source; on the other hand, discrepancies that are not justified.

The BMK uses terms from SNOMED CT, and additional *post-coordinated* concepts, i.e., in the meta-terminology of medical ontologies, concepts defined or constrained in terms of the ones provided in advance. One such concept C can be used in a BMK rule to state, for example, that execution of an action which is not the CG currently being executed, or the fact that an action prescribed by the CG is not executed, is (potentially) justified if the patient, other than the problem being dealt with by the CG, has a problem in the class C .

2 The GLARE Representation Formalism

In this section, we highlight some of the main features of the GLARE representation formalism (a detailed description is provided in [6]). GLARE distinguishes between *atomic* and *composite* actions. Atomic actions are elementary steps in a CG, in the sense that they do not need a further de-composition into sub-actions to be executed. Composite actions are instead composed by other (atomic or composite) actions. GLARE provides four different types of atomic actions:

- *work actions*, i.e., actions to be executed at a given point of the CG;
- *decision actions*, used to model the selection among alternative paths in a CG. GLARE provides *diagnostic* decisions, used to make explicit the identification of the disease the patient is suffering from, among a set of possible diseases, compatible with her findings. Such a decision is based on patient’s parameters. GLARE also provides *therapeutic* decisions, used to represent the choice between therapeutic paths in a CG, based on a pre-defined set of parameters: effectiveness, cost, side effects, compliance and duration;
- *query actions* model requests of information (typically patients’ parameters), that can be obtained from the outside world (e.g. physicians, databases, patients visits or interviews). CG execution cannot proceed until such information has been obtained;
- *conclusion actions* represent the explicit output of a decision process.

Actions in a CG are connected through control relations. Such relations establish which actions might be executed next, in which order. GLARE introduces four different types of control relations: *sequence*, *concurrency*, *alternative* and *repetition*. The sequence relation explicitly establishes which is the next action to be executed; the alternative relation describes which alternative paths stem from a decision action, the concurrency relation between two actions states that they can be executed in any order, or also in parallel and the repetition relation, states that an action has to be repeated several times (i.e. the number of repetitions can be fixed a priori, or, alternatively, it can be asserted that the action must be repeated until a certain exit condition becomes true).

3 SNOMED CT

The Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) is a standardized healthcare terminology. It is developed and distributed by the International Health Terminology Standards Development Organization (IHTSDO). SNOMED CT was created with the aim of improving data quality and patient safety, facilitating semantic interoperability by capturing clinical data in a standardized, unambiguous and granular manner. It is used in more than 50 countries around the world, as the foundation for electronic health records and other applications [8]. SNOMED CT is distributed in its official release format RF2 with a parser to generate an OWL 2 EL version of the terminological knowledge. The number of concepts, descriptions, and relationships varies with every release.

SNOMED CT contains more than 300,000 concepts and consists of several independent hierarchies ranging from Disease, Drug, Living organism, Procedure, to more general concepts as Physical Object and Physical Force.

ELK [9] is a Description Logic reasoner developed to provide high performance reasoning support for OWL 2 EL, whose underlying logic is the low-complexity description logic \mathcal{EL}^{++} ; see, e.g., [10] for a discussion on the expressiveness needed for the medical domain. In [11], ELK is evaluated to be the fastest reasoner in loading and classifying SNOMED CT as well as other ontologies.

4 Conformance Analysis Framework

A main goal of the framework presented in this paper is to exploit reusability of knowledge, in several ways:

- A model of the CG in Answer Set Programming (ASP, [12]) is derived automatically from the description of the CG in GLARE, and can be used for conformance analysis, as in this paper, i.e., analyzing if and how a single execution deviates from the CG, as well as for verifying properties of the CG, that should hold for all executions, e.g., using the approach in [13] as model checker in the loosely coupled framework in [14].
- A common repository of Basic Medical Knowledge (BMK) can be used, in the framework in this paper, with models of different CGs.
- The terminology, based on SNOMED CT, provides the link for triggering BMK rules for a specific CG and its execution on a specific patient.

Figure 1 presents the general framework. The main entities, to be input to an ASP solver, are ASP representations of the log, the CG model, BMK rules and the set of compliance annotation rules. BMK rules use subsumption of concepts in the terminology that make it possible to interpret the current situation as a case of application of the rule; in the current framework, subsumptions that may be relevant for a given log are queried in advance to ELK. The framework evaluates discrepancies of the log (actual execution) wrt executions suggested by the CG, considering the possible “variations” suggested by the BMK.

The log contains the data recorded during guideline execution. It includes data specific to the individual patient, such as medical records (from the Electronic Patient Record, EPR) and the actions performed on the patient; it also includes data related to the context (e.g., hospital) in which the CG is performed, such as availability of equipment and personnel.

The ASP model of the CG encodes all the admitted treatment paths provided by the CG. Tools such as GLARE provide a formal representation of CGs, which can be translated to ASP. In this framework, information on when an action is executed is used both to verify whether it is justified by the CG, and to justify execution of subsequent actions in the CG. Both the control flow perspective and the data perspective of the GLARE CG specification is encoded in the CG ASP model. In the current version, quantitative time constraints present in GLARE are not supported.

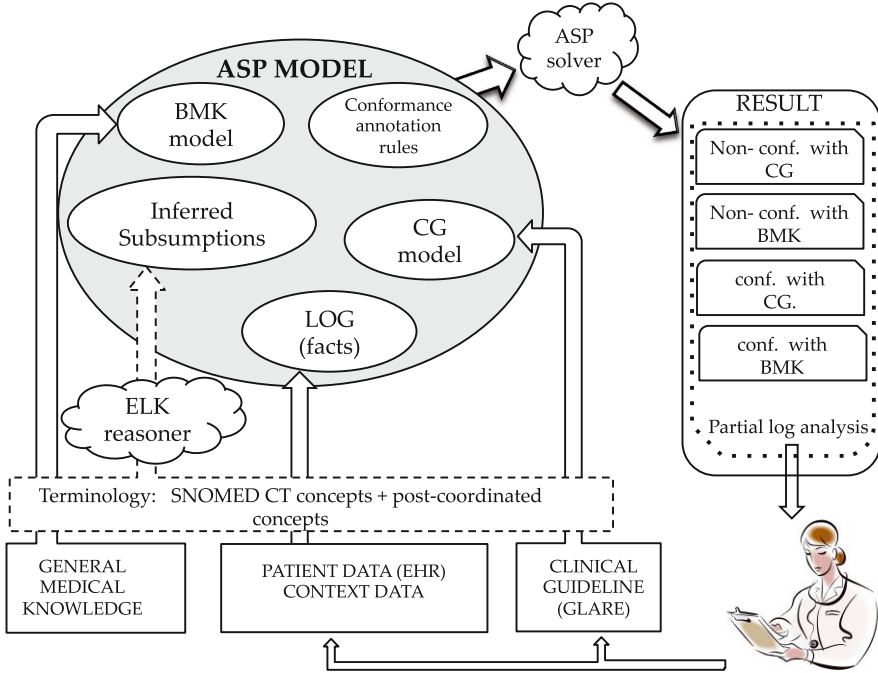


Fig. 1. General framework

To better evaluate the interplay of BMK and CG we take into account the action execution model in Fig. 2, similar to the one in [4]. At a given point in the execution of a CG on a specific patient, the control flow of the CG or rules in the BMK indicate that a given action has to be executed (is a candidate). A candidate action is discarded if its preconditions (modeled in the CG) are false; or it may be discarded because of conditions that are not explicitly modeled in the CG, but are, hopefully, modeled in the BMK as reasons for discarding it. Decision and conclusion actions are instantaneous. Work actions and query actions, once started, can either be completed or aborted. An action is aborted if a failure occurs during its execution, or it may be aborted because some condition arises; again, we expect that some of such reasons for aborting are modeled in the BMK.

Once an action of the CG is discarded or aborted, in general we cannot infer the correct way to continue the execution of the CG. In some cases the physician would continue the execution skipping the uncompleted action (e.g., for an action having minor impact on the treatment), in other cases she would restart the execution from some point further away (e.g., a previous decision point or the end of the partial plan), or the entire CG should be interrupted (e.g., in case the action is essential for the treatment). We do not assume that this information is modeled, therefore we suggest that the analyst should point out where in the CG and in the log the analysis can be restarted.

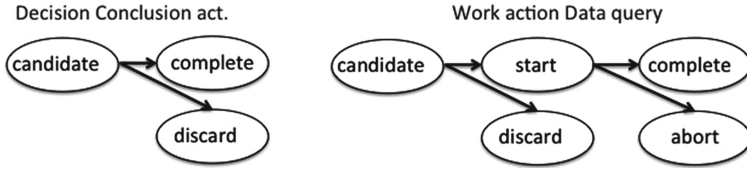


Fig. 2. Action model

The annotation compliance rules are the keystone of the entire framework. They define the output of the analysis, and are triggered by discrepancies, starting from the actions recorded in the log and the expected actions derived from the CG and BMK. Two different classes of discrepancies are provided:

- Discrepancies of the log with a knowledge source (KS; either the CG or a BMK rule) which are “supported” by another source.
- Discrepancies of the log with a KS that are not supported by another one.

While the second class represents incorrect behavior (wrt the considered KSs), the first one represents a case of (at least, potential) conflict between knowledge sources. Which one should prevail cannot be stated in general [4], and providing knowledge for stating this for all cases is, in general, too costly. Therefore we provide the information in the log, which can be filtered further by the analyst.

We assume completeness and correctness of the Log. Completeness with respect to actions means that for all actions taken, the following is recorded:

- start, discard, abort, complete and failure reason (human and/or technical problem which caused incorrect completion of an action);
- the outcome of completed decision actions.

Completeness with respect to (patient or context) data means that the log contains record of data which have driven the control flow (CF) and data which could force the physician to change the normal execution applying BMK rules. Correctness means that only verified information is recorded, no conflicting data can be stored (e.g., an action is first discarded and then completed).

We expect (see [4]) that the BMK provides pieces of knowledge such as:

- Actions of a given type, or specific actions, are contraindicated for patients in a given temporary or permanent condition; e.g. an invasive exam (suitable to get more information on the problem treated by the CG) is contraindicated for patients also suffering from a problem in a given class C ;
- the execution of a CG may (have to) be suspended if a more urgent problem (e.g., a life threat) arises, and the latter one should be treated. Whether the execution actually *has* to be suspended depends, in general, on whether the current actions being executed are compatible with the treatment of the more urgent problem. Specific knowledge in this respect may be available or not. We intend that the other problem (e.g., a heart failure) is not part of

the class of problems dealt with by the CG. The source of knowledge for its treatment should, in principle, come from another CG; however, in this paper we do not address the problem of interaction of multiple CGs and we assume to have available, when analyzing logs for the execution of a CG, the set of possible treatments for other problems.

- Actions of a given type (e.g., routine exams) can be performed even if not part of the CG.

5 ASP Representation and Conformance Rules

In this section we describe the ASP representation including the one for the Log, the CG model, the BMK model, their relation with SNOMED CT, and the annotation rules.

5.1 Log Representation

In the ASP representation of the Log, context and patient data, action states and decision outcomes are encoded as follows:

- `holds(var(name,value),timeStamp)` represents the fact (from the EPR) that a patient or context datum `name` has value `value` at time `timeStamp`;
- `holds(problem,timeStamp)` represents the fact (from the EPR) that `problem` holds for the patient at time `timeStamp`;
- `action(actID,actState,timeStamp)` represents the fact that for action `actID` there is a transition to state `actState` (discard, started, aborted, completed) at time `timeStamp`;
- `decision(actID,actIDoutcome,timeStamp)` represents the fact that at time `timeStamp`, the result of the decision action `actID`, performed by the physician, is to perform action `actIDoutcome`.

We reconstruct the timeline for the framework with the predicate *next*:

```
next(S,SN):-state(S),state(SN),SN>S, not stateinbetween(S,SN).
stateinbetween(S,S2):-state(S),state(S2),state(S3),S<S3,S3<S2.
```

A predicate *state(S)* is true for all timestamps *S*; *next(S,SN)* is true for all the pairs of timestamps with no states in between. The rules below propagate data values up to the next change:

```
holds(var(N,V),SN):- holds(var(N,V),S), next(S,SN),
                    not holds(var(N,V1),SN), V!=V1.
```

The occurrence of a situation described by a term in the terminology (i.e., SNOMED CT extended with post-coordinated concepts) should be matched with information describing the situation at a given timestamp.

In medical reasoning it is not obvious how information that constitutes a problem to be treated should be separated from other information relative to the patient and context; the latter may be relevant or not for solving the problem,

but we cannot expect to describe it together with the problem in a single medical term. More generally, part of the information on the patient should be aggregated into a set of problems which need treatment. Such a description has a dynamics: e.g., new information on a problem may become available, or the problem itself may change (e.g. it may become worse, or improve with treatment, or change from acute to chronic); or, more generally, the aggregation might change (a single problem should be split into two problems, or vice versa). However, modeling such evolution is not the purpose of this work. What we need is:

- conformance reasoning, which we intend to represent in ASP;
- terminological inference, which should classify, in the terminological knowledge base, a problem the patient has, or an action occurring in the CG or in a BMK rule; to this purpose, we use ELK, which is the obvious choice for SNOMED CT, and, as far as post-coordinated concepts are concerned, covers a significant subset of \mathcal{EL}^{++} which, like the basic \mathcal{EL} , has low complexity, in particular, subsumption in \mathcal{EL}^{++} is polynomial [15].

This is obtained as follows:

- a problem the patient has at a given snapshot t is supposed to be represented (as in [8]) as an atomic concept B occurring in the terminological knowledge base (in case it is a non-atomic concept C , a new name B is introduced and $B \equiv C$ is added to the terminological knowledge base);
- a constant \mathfrak{b} is used to represent B in ASP;
- $\text{holds}(\mathfrak{b}, \mathfrak{t})$ is used to state that problem \mathfrak{b} holds for the patient at time \mathfrak{t} ;
- as we shall see in Sect. 5.3, BMK rules contain atoms of the form $\text{is_a}(\mathfrak{u}, \mathfrak{u}')$, where $\mathfrak{u}, \mathfrak{u}'$ are either ASP constants corresponding to atomic concepts, or ASP variables; for ground instances $\text{is_a}(\mathfrak{b}, \mathfrak{c})$, the subsumption $B \sqsubseteq C$ is checked in ELK¹;
- persistence of the problem description is given by:

$$\text{holds}(\mathfrak{C}, \text{SN}) :- \text{holds}(\mathfrak{C}, \mathfrak{S}), \text{next}(\mathfrak{S}, \text{SN}), \text{not } \text{-holds}(\mathfrak{C}, \text{SN}).$$

- an explicit statement $\text{-holds}(\mathfrak{b}, \mathfrak{t}')$ (where “-” represents explicit negation) is included (to block persistence) for the time \mathfrak{t}' when \mathfrak{b} is known to no longer be the description of a problem the patient has. This information is supposed to be given in the EPR when the problem description (in case new information has become available) or the problem itself has changed. In that case a statement $\text{holds}(\mathfrak{b}', \mathfrak{t}')$ should also be provided for the new concept \mathfrak{b}' describing the current situation. Notice that this is correct, in particular, for the case where there is no evidence that the problem has changed, but new information has been acquired. In fact, a BMK rule may state that a discrepancy from a CG prescription is potentially justified at some time \mathfrak{t}

¹ This can be seen as an extension of ASP where, in the grounding of ASP rules, variables in such is_a atoms are instantiated in all possible ways with atomic concepts occurring in the ASP model; the semantics of the grounded program can be taken to be a special case of the weak answer set semantics in [16] where DL queries are restricted to be concept inclusions only.

if there is a problem which is a case of some problem class c , i.e., it would contain the condition: $\text{holds}(P,S), \text{isa}(P,c)$. Information on the problem that only becomes available at time $t' > t$, should not be used to infer that the problem is a case of c in order to justify an action (or the absence of an action) at time t . This does not occur, because $\text{holds}(b,t)$, $\text{holds}(b',t')$, and $\neg\text{holds}(b,t')$ would be in the ASP representation of the log, where $\text{isa}(b',c)$ can be inferred from terminological knowledge while $\text{isa}(b,c)$ cannot.

5.2 CG Model

The CG model is not reported in full detail. The main CG component is the control flow (CF), we encode it in ASP similarly to the approach in [13]. The CF model defines $\text{candidate}(A,S)$ for actions A and states S . There are atomic actions and composed actions. Predicates $\text{end}(\text{Type},A,S)$ and $\text{start}(\text{Type},A,S)$ (where Type is either `group`, `plan` or `atomic`) are defined to reconstruct the execution interval of composite actions from the ones of atomic actions, which are registered in the log. The definition of end relates the end of atomic actions to the end of composite actions and control structures; e.g., a set of actions in a group is considered ended in S only if all the sub-processes are ended in S .

Every candidate action a (atomic/composite) can be executed, and once it ends, it enables its successors (in the control flow) $a1$ in the *next* time state by means of the predicate $\text{candidate}(A,SN)$:

- (a) $\text{candidate}(A1,SN) :- \text{succ}(A,A1), \text{not } \text{excp}(A1,S), \text{end}(_,A,S), \text{next}(S,SN)$.
- (b) $\text{candidate}(A1,SN) :- \text{decision}(A,A1,S), \text{end}(\text{action},A,S), \text{next}(S,SN)$.
- (c) $\text{candidate}(A1,SN) :- \text{end}(_,A,S), \text{next}(S,SN), \text{reExecute}(A,S)$.

In rule (a), $A1$ is candidate in SN if A ended in S and $A1$ is the *successor* of A in the flow. The predicate $\text{excp}(A,S)$, blocks execution after actions that terminated with errors or for other reasons (e.g. a completed data query without data) should not lead to the next action in the flow. Rule (b) corresponds to *decisions*: the outcome of the decision task, as registered in the log, enables the proper successor action. Rule (c) encodes *repetition*. All actions (atomic/composite), if specified by the CG, can be re-executed: $\text{reExecute}(\text{ID},\text{state})$ is true if the action ends and the exit condition on data is false. Other CG specifications mapped in ASP are the list of data requested by a data query action, parameters to evaluate therapeutic decision, exit conditions for repetitions and preconditions of action.

5.3 BMK Rules and Clinical Terminology

The BMK model consists of a set of rules which prescribe or allow the introduction or cancellation of an action, based on conditions on the patient and contextual data. Such *conditions* are defined in other rules, also making use of the terminology. BMK rules have the following forms:

```

prescribe(id,A,normal/urgent,S):- condition(S).
allow(id,A,S):- condition(S).
prescribeCanc(id,A,S):- condition(S).
allowCanc(id,A,S):- condition(S).

```

The *id* is used to point out in the analysis the set of rules that has generated or justified a discrepancy. Multiple instances of *prescribe/allow* with the same *id* and different actions encode the request to execute one of a set of possible alternative actions; (*normal/urgent*) encodes the urgency to execute the action: in the *normal* case, there is no constraint on the order of execution wrt other actions, while, in the *urgent* case, it must be the first action to be executed once the condition is true. The difference between *prescribe* and *allow* is that:

- for *prescribe*, a discrepancy is reported both in case the action is executed (as it was not an action in the CG) and in case it is not executed (it is a discrepancy wrt the prescription in this rule, possibly due to the fact that the CG overrides BMK in this case)
- for *allow*, a discrepancy is reported only when the action is executed

and similarly for *prescribeCanc* and *allowCanc*. In fact, the four predicates are related to action events as follows, to define *candidate*, *discard* and *abort* which are used in the annotation rules to point out discrepancies:

```

candidate(A,S,ID):-prescribe(ID,A,normal,S),not running(A,S).
candidate(A,S,ID):-prescribe(ID,A,urgent,S),not running(A,S).
urgent(ID,A,S):-prescribe(ID,A,urgent,S),not running(A,S).
candidate(A,S,ID):-allow(ID,A,S),action(A,started,S).
discard(A,S,ID):-prescribeCanc(ID,A,S),candidate(A,S).
abort(A,S,ID):-prescribeCanc(ID,A,S),running(A,S).
discard(A,S,ID):-allowCanc(ID,A,S),action(A,discarded,S),candidate(A,S).
abort(A,S,ID):-allowCanc(ID,A,S),running(A,S),action(A,aborted,S).

```

The *conditions* in rules also make use of concepts defined in terminological knowledge. In particular, we use atoms of the form: *is_a(u,u')*, where *u,u'* are either ASP constants corresponding to atomic concepts, or ASP variables.

For all pairs *c,d* of constants, which occur in the log or BMK rules, and correspond to atomic concepts, before performing conformance analysis, ELK is queried to check whether $C \sqsubseteq D$, where *C* and *D* are the atomic concepts corresponding to *c* and *d*. In that case, *is_a(c,d)* is added to the ASP representation.

In the following we provide the representation for some of the rules in [4].

BMK: Calcemia and glycemia measurements are routinely performed in all patients admitted to the internal medicine ward of Italian hospitals, regardless of the disease.

SNOMED CT contains the concept name *blood calcium measurement* to represent calcemia measurement, and the concept name *blood glucose measurement* which subsumes 42 kind of glycemia tests. We add a concept *routine action* including such classes:

blood calcium measurement \sqcup *blood glucose measurement* \sqcup ... \sqsubseteq *routine action*

and the rule:

```
allow(r1,A,S):-holds(var(admitted_Int_med,true),S),is_a(A,routine_act).
```

where *routine.act* corresponds to the concept *routine action*.

BMK: Contrast media administration for coronary angiography may cause a further final deterioration of the renal functions in patients affected by unstable advanced predialytic renal failure. Assuming that the latter is suitably defined, using the terms already available in SNOMED CT, as a concept corresponding to the ASP constant `adv_predial_renal_failure`, the rule can be represented as:

```
prescribeCanc(r2,A,S):- holds(D,S),is_a(D,adv_predial_renal_failure).
```

BMK: The execution of any CG may be suspended, if a problem threatening the patients life suddenly arises. Such a problem has to be treated first. One such problem is acute heart failure; an immediate response for it could be a Diuretic Therapy.

SNOMED CT provides a severity property for diseases; although there is no pre-coordinated concept using it, we assume that for diseases that are considered to be life threatening for the purpose of the above rule, their severity is provided (e.g., acute heart failure is stated to have a life-threatening severity), the concept *LifeThreat* is added with the following constraint:

$$LifeThreat \equiv Disease \sqcap \exists Severity. LifeThreateningSeverity$$

The following BMK rule models the fact that if there is a life threat \mathfrak{D} , then an action \mathfrak{Act} which is a special case of a treatment \mathfrak{T} , suitable for a superclass $\mathfrak{D1}$ of \mathfrak{D} , is justified:

```
prescribe(r3,Act,urgent,S):-holds(D,S), is_a(D,lifeThreat)
    treatment(D1,T),is_a(D,D1),is_a(Act,T)
```

and in the same situation an action which is not a (special case of) treatment for (a superclass of) \mathfrak{D} should be cancelled:

```
prescribeCanc(r3,T,S):-holds(D,S), is_a(D,lifeThreat),
    running(T,S),not is_a(T,T1):treatment(D1,T1):is_a(D1,D).
```

The `treatment` relation is modeled explicitly in ASP:

```
treatment(heartFailure,diureticTherapy).
treatment(heartFailure,betaBlockerTherapy).
treatment(heartFailure,inotropeTherapy) [...]
```

to mean that all diuretic therapies, beta blocker therapies, ..., are (in principle) suitable in case of heart failure. Of course, in case more specific information is available (i.e., that a type of therapy is only suitable for some specific class of heart failure), it should be provided. However, the BMK rule above is quite general: it is not intended to prescribe a specific appropriate treatment for a specific case (neither a CG is supposed to do so), but to provide a general justification for a deviation from the execution of a CG.

Notice moreover that properties in SNOMED CT are intended in \mathcal{EL} as existential restrictions. If the property of *DiureticTherapy*, that consists in being a treatment for *HeartFailure*, were added as:

$$DiureticTherapy \sqsubseteq \exists IsTreatmentFor.HeartFailure,$$

it would only mean that for any *DiureticTherapy* there is some (individual) *HeartFailure* for which it is a treatment. Moreover, for any superclass C of *HeartFailure*, $DiureticTherapy \sqsubseteq \exists IsTreatmentFor.C$ can be inferred, and of course we do not want to interpret this as “*DiureticTherapy* is a treatment for C ”.

5.4 Conformance Annotation Rules

In a state t , relatively to action a , the following discrepancies, potentially justified by a knowledge source (KS, either the CG or a BMK rule), are defined:

- A1 **A discrepancy with a KS s , justified by a BMK rule r** , if a is recorded as discarded in t , rule r prescribes discarding a , and s suggests a as candidate.
- A2 **A discrepancy with a BMK rule r , justified by a KS s** , if a is recorded as started in t , rule r prescribes discarding a , and s suggests a as candidate.
- A3 **A discrepancy with a KS s , justified by BMK rule r** , if a is recorded as aborted in t , rule r prescribes aborting a and, until t , action a was running as suggested by s .
- A4 **A discrepancy with a BMK rule r justified by the KS s** , if a is recorded as completed in t , rule r prescribes aborting a and, until t , action a was running as suggested by s .
- A5 **A discrepancy with a BMK rule r justified by the KS s** , if in a state in t a rule r prescribes with urgency one or more actions and a different action, prescribed by s , is recorded as started.

In a state t , relatively to action a , the following discrepancies not justified by other knowledge sources are output:

- B1 **A discrepancy with the KS s** , if a is recorded as discarded in t , s suggests a as candidate, no BMK rule r justifies discarding a and preconditions of a are satisfied at t .
- B2 **A discrepancy with the KS s** , if a is recorded as started in t , s suggests a as candidate and preconditions of a are falsified at t .
- B3 **A discrepancy with all KSs**, if a is recorded as started in t , there is no source s which suggests a as candidate.
- B4 **A discrepancy with the KS s** , if a is recorded as aborted, no BMK rule r justify aborting a , no failure is recorded for a and, until t , action a was running as suggested by s .
- B5 **A discrepancy with the CG**, if a was candidate by the CG at t and, after t , a is not recorded as started nor discarded.
- B6 **A discrepancy with a BMK rule r** , if in an interval $[t_0, t]$ a rule r suggests a , and possibly alternative actions, as candidates; the actions are not candidate at $t+1$, and at $t+1$ no action suggested by r is executed or discarded.

The encoding of the above rules in ASP is relatively straightforward. E.g., for A1 we have the two clauses below; the first one is for discrepancies with respect to the CG, the second one for a discrepancy wrt a BMK rule, as recorded in the first argument of `discrepancy`:

```
discrepancy(cg, ID, A, S) :- action(A, discarded, S), discard(A, S, ID),
    not precondFalse(A, S), candidate(A, S).
discrepancy(ID2, ID, A, S) :- action(A, discarded, S), discard(A, S, ID),
    not precondFalse(A, S), candidate(A, S, ID2).
```

An ASP solver such as Clingo [17] computes an answer set of the overall ASP model (see Fig. 1); the set of instances of the *discrepancy* predicate in the answer set contains the information necessary to produce a user-friendly result.

6 Framework Execution Example

Let us consider an example of execution of a fragment of CG for acute myocardial infarction associated to a BMK containing the rules presented in Sect. 5.3. The fragment of CG (Fig. 3) consists of three activities in sequence, Electrocardiographic study (ECG), Echocardiographic study (ES) and Coronary Angiography (CA).

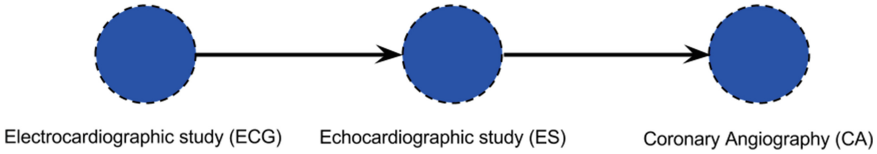


Fig. 3. Fragment of CG for acute myocardial infarction

Consider now the partial log, shown in Fig. 4, of a specific execution of the CG. The log contains trace of actions that have been executed (started and completed): ECG, Glucose measurement blood test strip (BMTS) and ES; trace of the discarded action CA; and patient data: the patient was admitted to the internal medicine ward, the patient had an acute heart failure.

In this example the execution of ECG and ES is compliant with the model of the CG, but discarding CA is not, as the action has no preconditions which can fail. From the BMK perspective, executing BMTS and discarding CA is correct while the lack of a treatment for the heart failure is a violation of the expected behavior. The first rule introduced in Sect. 5.3 (r1), triggered by the admission in the internal medicine ward, allows the execution of any action subsumed by the (post-coordinated) concept “routine action”. By the definition of this concept given before, the execution of BMTS is allowed since the action is subsumed by “glucose blood measurement” in SNOMED-CT. From time 17 the acute heart failure diagnosis triggers rule r3 in Sect. 5.3. This rule prescribes the cancellation of any action which is not a treatment for the life threatening problem, causing the annotation of the discrepancy A1 at time 19. Rule r3 also prescribes the execution with urgency of a reparative treatment, but no action is recorded as started, and the discrepancy B6 is reported at the end of the log.

7 Related Work and Conclusions

We presented a framework for analyzing conformance of execution traces for patient treatment with Clinical Guidelines relying on Basic Medical Knowledge and Clinical Terminology.

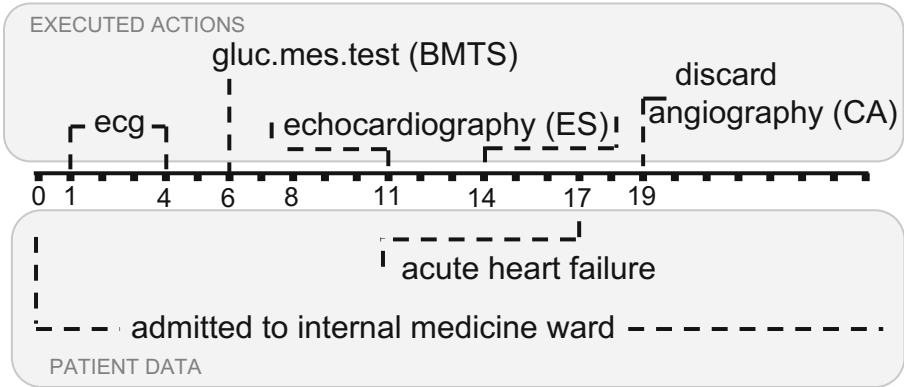


Fig. 4. Execution trace

The approach, as presented in the paper, is specific to healthcare processes, but a similar one can be used for comparing actual execution traces with process models in other organizations, i.e., for *business* processes; also in other contexts, in fact, there might be “ideal” process models, which make sense as a reference, but do not define all conceivable process adaptations in all situations.

Conformance analysis work in the process model area, e.g. [18–20], is mainly devoted to measuring the adherence of a model with execution traces, in order to refine a model, rather than to analyze, as in our approach, the correctness of an execution with respect to a model.

Our approach builds on the work in [4], which is mainly devoted to studying the interaction between CGs and BMK from the viewpoint of the conformance problem. In this paper, we proposed a general framework architecture, which allow us to integrate existing ontologies (SNOMED CT, in particular) as knowledge sources and to exploit terminological inference in conformance analysis. Moreover, in [4] the authors identify only non-adherence situations to the CG and/or BMK, while we propose a finer classification where we point out possible justifications, when one knowledge source “supports” a situation of non-adherence to another source.

The issue of compliance with clinical guidelines is discussed in [21] taking into account a wide range of reasons for non-adherence to guidelines, from “human factors” regarding both patients and physicians, who do not necessarily accept guidelines especially because they tend to be too rigid, to reasons that are considered in this paper, e.g., patient characteristics that make the guideline inappropriate for her. The author points out that even in a system that supports execution of guidelines, analyzing non-compliance at execution time might be inappropriate because treating patients may be more urgent than documenting the actions being performed, or because relevant data may not yet be stored in the system; patient discharge might be a more appropriate time when physicians can document non-compliance. In the present paper, we propose using

annotation rules in an off-line analysis of logs; however, they could also be used to support annotation at discharge time.

Müller et al. [22] developed the AGENTWORK system for adaptation of (healthcare) workflows to handle exceptional situations; Event/Condition/Action rules are used to model such situations, given that explicitly modeling them in the workflow would greatly reduce its readability. Our approach is not devoted to provide support at run time, and in our case rules allow or suggest deviations from the guideline, without being prescriptive.

As regards CGs and medical knowledge, the approach does not take into account the general problem of interaction of multiple CGs, where general medical knowledge should of course play a role. A recent approach in this respect is presented in [23]; it is based on an ontology of intentions (goals that actions should achieve, e.g., decreasing blood pressure), drugs (whose administration can achieve the intentions) and interactions of such intentions and drugs. The approach also relies on SNOMED-CT. Such a model of interactions can of course be used to justify discrepancies from a CG execution, if not already used to support decision at action execution time for patients affected by multiple diseases.

References

1. Field, M.J., Lohr, K.N. (eds.): *Guidelines for Clinical Practice: From Development to Use*. National Academy Press, Institute of Medicine, Washington, D.C. (1992)
2. Kish, M.A.: Guide to development of practice guidelines. *Clin. Infect. Dis.* **32**(6), 851–854 (2001)
3. 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)
4. 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)
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: Adlassnig, K.-P., Blobel, B., Mantas, J., Masic, I. (eds.) *MIE, Studies in Health Technology and Informatics*, vol. 150, pp. 688–692. IOS Press, Amsterdam (2009)
6. Terenziani, P., Molino, G., Torchio, M.: A modular approach for representing and executing clinical guidelines. *Artif. Intell. Med.* **23**(3), 249–276 (2001)
7. International Health Terminology Standards Development Organization: SNOMED CT. <http://www.ihtsdo.org/snomed-ct/>
8. Lee, D., Cornet, R., Lau, F., de Keizer, N.: A survey of SNOMED CT implementations. *J. Biomed. Inform.* **46**(1), 87–96 (2013)
9. ELK. <https://code.google.com/p/elk-reasoner/>
10. Rector, A.L.: Medical informatics. In: Baader, F., Calvanese, D., McGuinness, L., Nardi, D., Patel-Schneider, P.F. (eds.) *Description Logic Handbook* (2007)
11. Kazakov, Y., Krötzsch, M., Simancik, F.: ELK reasoner: architecture and evaluation. In: Horrocks, I., Yatskevich, M., Jiménez-Ruiz, E. (eds.) *Proceeding of International Workshop on OWL Reasoner Evaluation (ORE-2012)*, CEUR 858 (2012)

12. Gelfond, M.: Answer Sets. Handbook of Knowledge Representation, chapter 7. Elsevier, New York (2007)
13. Giordano, L., Martelli, A., Spiotta, M., Theseider Dupré, D.: Business process verification with constraint temporal answer set programming. *Theory Pract. Logic Program.* **13**(4–5), 563–578 (2013)
14. 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)
15. Baader, F., Brandt, S., Lutz, C.: Pushing the \mathcal{EL} envelope. In: *IJCAI*, pp. 364–369 (2005)
16. Eiter, T., Lanni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. *Artif. Intell.* **172**(12–13), 1495–1539 (2008)
17. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Schneider, M.: Potassco: the Potsdam answer set solving collection. *AI Commun.* **24**(2), 105–124 (2011)
18. Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.P.: Towards robust conformance checking. In: Muehlen, M., Su, J. (eds.) *BPM 2010 Workshops. LNBIP*, vol. 66, pp. 122–133. Springer, Heidelberg (2011)
19. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **33**(1), 64–95 (2008)
20. Munoz-Gama, J., Carmona, J.: Enhancing precision in process conformance: stability, confidence and severity. In: *Proceeding of CIDM*, pp. 184–191 (2011)
21. Quaglini, S.: Compliance with clinical practice guidelines. In: Ten Teije, P.L.A., Miksch, S. (eds.) *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*. IOS Press, Amsterdam (2008)
22. Müller, R., Greiner, U., Rahm, E.: AgentWork: a workflow system supporting rule-based workflow adaptation. *Data Knowl. Eng.* **51**(2), 223–256 (2004)
23. Piovesan, L., Molino, G., Terenziani, P.: An ontological knowledge and multiple abstraction level decision support system in healthcare. *Decis. Anal.* **1**(8), 1–24 (2014)