Conformance Checking of Executed Clinical Guidelines in presence of Basic Medical Knowledge

Alessio Bottrighi[1], **Federico Chesani[2]**, Paola Mello[2],
Marco Montali[3], Stefania Montani[1], Paolo Terenziani[1]

[1]DI, University of Piemonte Orientale

[2]DEIS, University of Bologna

[3]KRDB, Free University of Bozen-Bolzano

Outline

- Clinical Guidelines (CGs)
  - Short intro
  - CGs and *ideal* vs. *real* world
  - CGs and compliance
- A couple of big questions
- The interplay between CGs and medical knowledge
  - Activity lifecycle as a connection point between CG and BMK
- CG and BMK in EC
  - Characterization of conformance and evaluation with the Event Calculus

Clinical Guidelines

From the MeSH dictionary:

"… work consisting of a set of directions or principles to assist the health care practitioners with patient care decisions about appropriate diagnostic, therapeutic, or other clinical procedures for specific clinical circumstances."

- One of the main goals is to capture medical evidence and to put it into practice
  - By suggesting best practices, actions, procedures
  - Based on medical evidence … CGs are updated frequently

Clinical Guidelines

CGs are a mix of several different informations and suggestions:

- Description of the class of patients the CG applies to
- Medical protocols and procedures to follow (more structured)
  - E.g.: *"(when dealing with heart stroke)… actions "Electrocardiographic study", "Echocardiographic study", and "Coronary Angiography" should be executed in sequential order".*
- General instructions to apply in any situation, and/or to face expected/unexpected events (more unstructured)
  - E.g.: *"In a patient affected by unstable angina and advanced predialytic renal failure, coronary angiography remains mandatory, even if the contrast media administration may cause a further final deterioration of the renal functions, leading the patient to dialysis."*
- Organizational and management aspects

Clinical Guidelines

CGs adoption ensures:

- Standardization of the health processes within large health organisations
- Quality of the health processes
- Through the CG updates, spread of new knowledge among health professionals
- Possibly, costs evaluation, prediction and management

Ideal world

CGs are developed by applying evidence-based medicine to *large* classes of *abstract* patients.

Assumptions:

- Ideal patients
  - statistically relevant
  - with only the disease targeted by the CG
- Ideal physicians
- Ideal resources
  - All the resources needed for applying the CG are available … almost infinite resources!!!

Real World

- Context and patients are not ideal
  - Resources may be missing
  - Each single patient has its own story, condition, preferences
  - → Unforeseen situations are common
- → CGs are routinely adapted on a per patient basis, using the Basic Medical Knowledge (BMK)
- Physicians are not ideal
  - → they need (computerized) support

Computerized Clinical Guidelines

Many software tools are available for dealing with CGs, supporting:

- Formal languages for CGs definition and elicitation
- CG execution, application to patients, and logging
- Integration with the structure's databases (EPR)

However…

- No standards about the CG definition language
- Mainly, support for the procedural aspects only (workflows and BPM techniques)
  - support for other knowledge types, such as if-then rules and BMK, is missing

CGs and Compliance

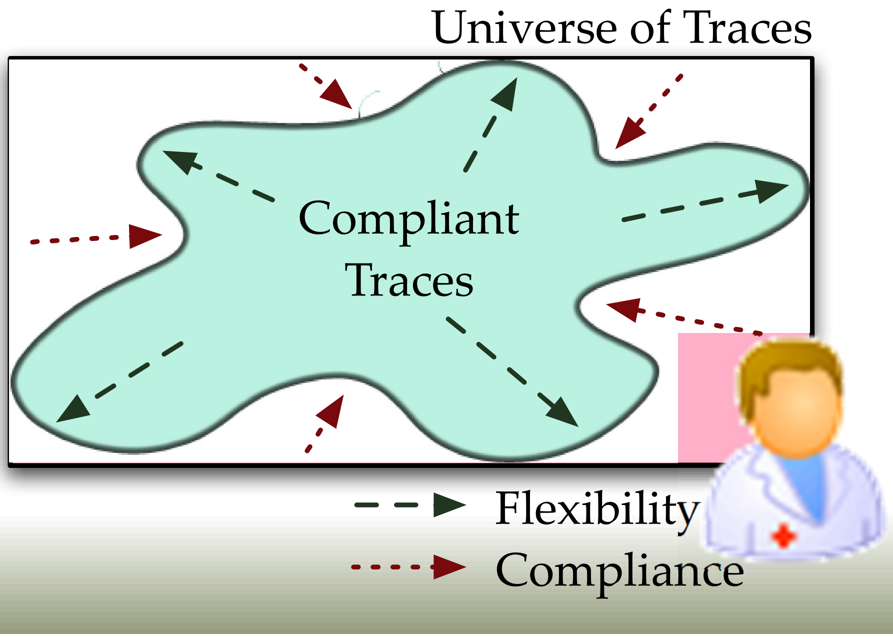
Moreover, given the log (a.k.a., execution trace) of the actions applied to a patient, how to evaluate if the traces are compliant to the CG? Keeping in mind that:

- CGs are about ideals, but are applied in real worlds
- Dynamic events and situation evolution heavily impact on the CG execution
- BMK is always heavily exploited during CG execution

A very important disclaimer: it is not a matter of evaluate the physician behavior. Rather, the focus is on the prescribed vs. real executions of the CG!

Compliance vs Flexibility

- **Compliance** The act of conforming as requested by the CG
- **Flexibility** The ability of accommodating and promptly adapting to change and unforeseen situations



Big Question (One)

How do BMK and CGs interact?

- **Compliance problem**: adherence of a CG execution trace to the *CG+BMK* model

Big Question (Two)

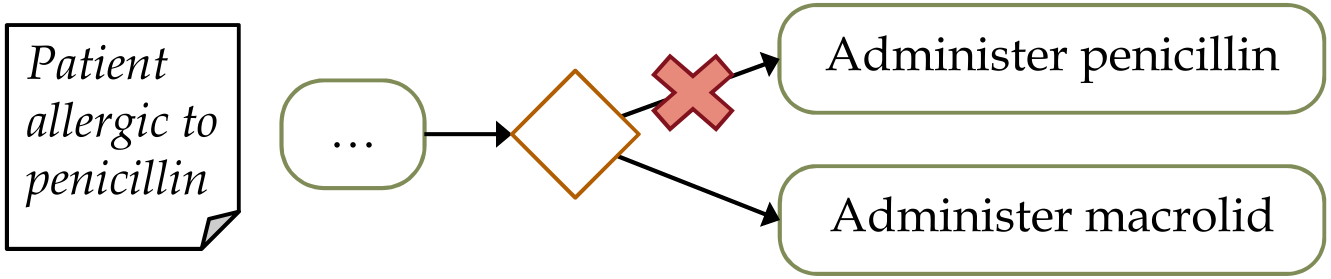
How to represent CG, BMK, and CG+BMK?

- And also: how to reason upon, and evaluate the compliance?

Choice enforcement

CG	BMK
Patients suffering from bacterial pneumonia caused by agents sensible to penicillin and to macrolid, must be treated one of them	Don’t administer drugs to an allergic patient.

- BMK reinforces the CG helping in the discrimination among possible alternatives

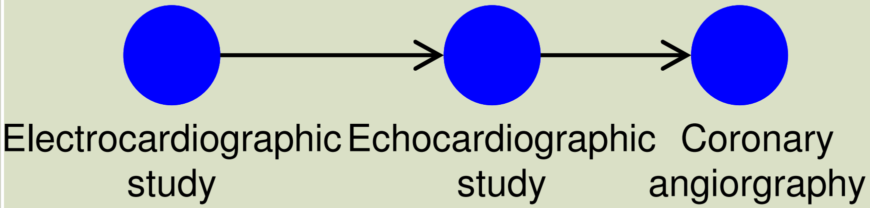


Openness

CG	BMK
	Calcemia and glycemia are routinely performed for all patients admitted to the internal medicine ward of Italian hospitals.

- BMK introduces further activities that can/must be executed alongside the ones of the CG
- The CG cannot be interpreted as a *closed* specification
  - Closed = everything not explicitly mentioned is forbidden

Exceptions

CG *in GLARE [Terenziani et al.]*	**BMK**
 Electrocardiographic study Echocardiographic study Coronary angiorgraphy	Threats to patient's life must be addressed immediately.

- (sometimes) CG's prescriptions = standard behavior
- BMK may introduce high-priority prescriptions used to deal with exceptional situations
  - They override the CG

Mandatory behaviors

CG	BMK
In a patient affected by unstable angina and advanced predialytic renal failure, coronary angiography remains mandatory, even if the contrast media administration may cause a further final deterioration of the renal functions, leading the patient to dialysis.	Don't administer treatments to the patient when they are likely to be dangerous

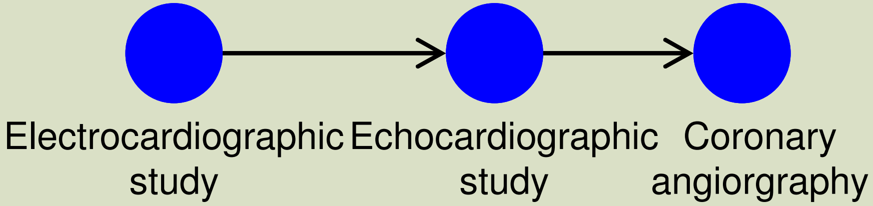
- (sometimes) BMK = default situation
- CG introduces mandatory activities in order to handle special cases
  - They override the BMK

CG+BMK: Lessons learnt 1/2

- BMK is used to fill the gap between the ideal world targeted by CGs and the real world
- The interplay between CG and BMK is complex
  - It is likely the case they *seem* to contradict each other
- Contradiction is only apparent
  - CG actions should not be interpreted as "must do" actions
  - Both CG and BMK are "defeasible"
    - Parts of the CG are amended by the BMK
    - Parts of the BMK are overridden by the CG

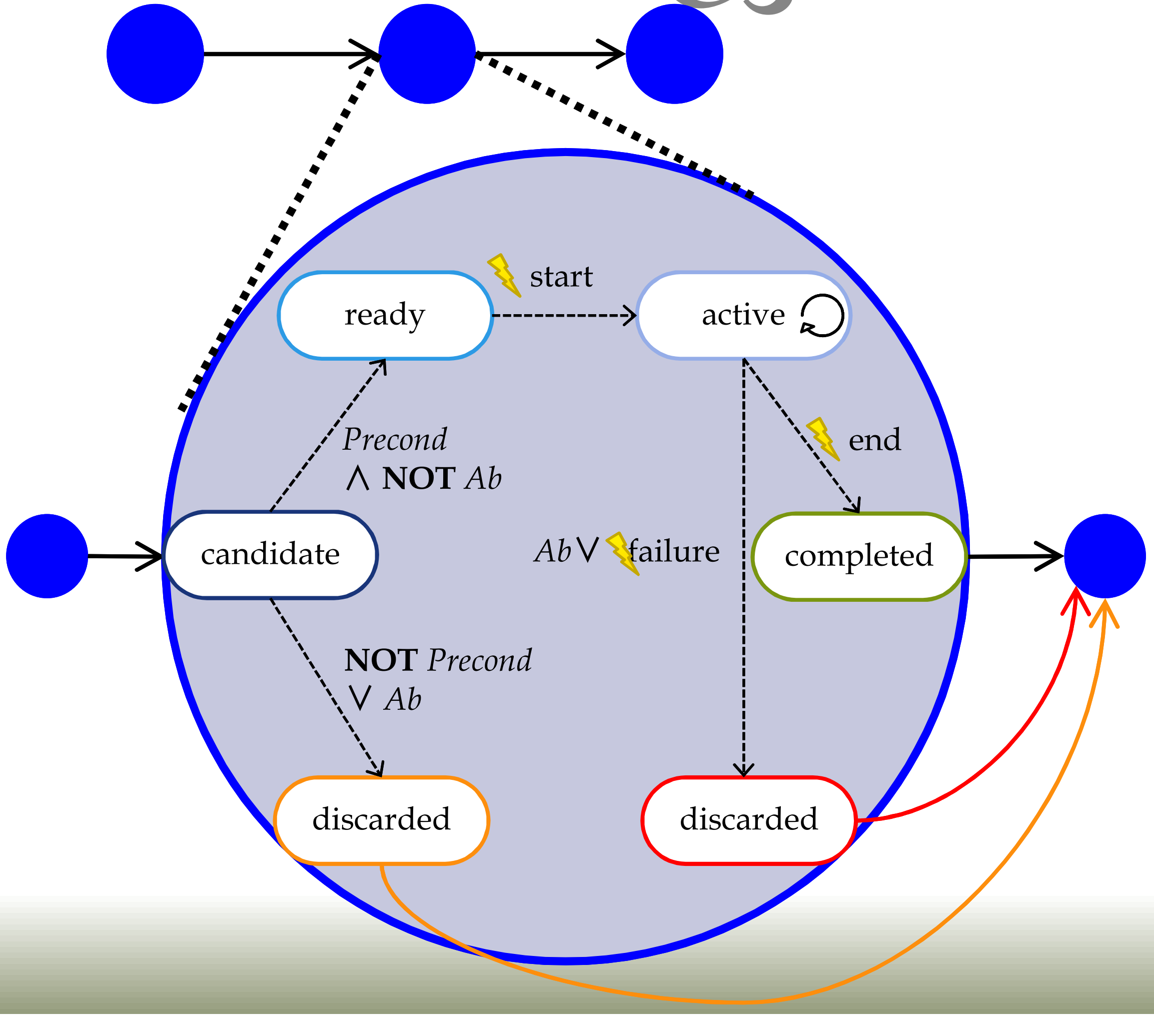
CG+BMK: Lessons learnt 2/2

- Both BMK and CG may involve declarative and procedural knowledge
- **Procedural knowledge** fixes the sequencing of actions to be done
- **Declarative knowledge** captures constraints and properties to be satisfied, without saying "how"

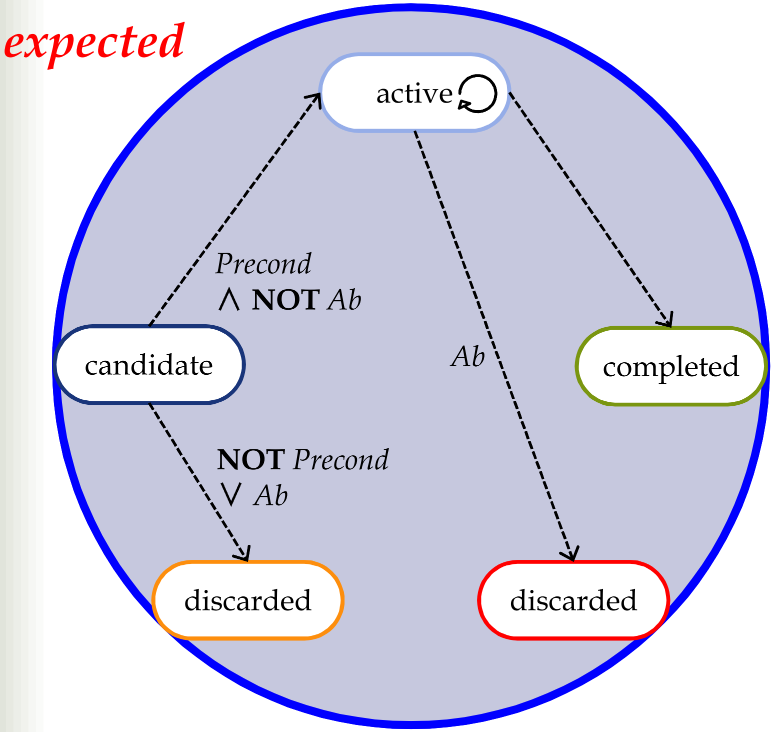
CG *in GLARE [Terenziani et al.]*	BMK
 Electrocardiographic study → Echocardiographic study → Coronary angiorgraphy	Threats to patient's life must be addressed immediately. An hearth failure is a life threat. Diuretic therapy is a possible immediate response for acute heart failure.

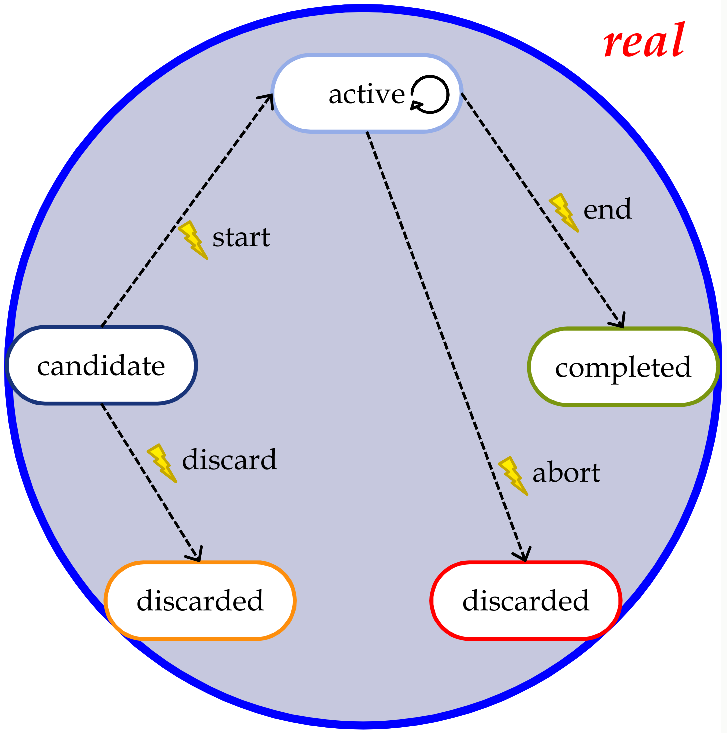
Our answer to question one: extending the

Activity Lifecycle



Expected vs. *Real* activity life cycle





- Ready and candidate states collapsed
- Expected life cycle → triggered by logical conditions
- Real life cycle → triggered by event occurrences
- **Compliance: detect and show deviations**

About question two:
Reasoning - Requirements

- Ability to reason upon events, time and data
    - Events characterizing the life cycle of activities
        - E.g.: *"Glucose test completed at time …"*
    - Events used to collect information about patient and context
        - E.g.: *"At time …, the patient had an heart failure."*
- Ability to deal with declarative and procedural knowledge
    - Rules
    - States/milestones

Our answer to question two:

Event Calculus

[Kowalski and Sergot, 1986]

- A logic-based framework for reasoning upon events and their effects
- Composed of
  - EC ontology: a set of special predicates to represent how events manipulate fluents
    - Fluent = property that changes over time → states!
  - A logic-based formalization capturing the semantics of the EC ontology
- Can be axiomatized using logic programming with NAF
  - Prolog!

Our answer to question two:

Event Calculus

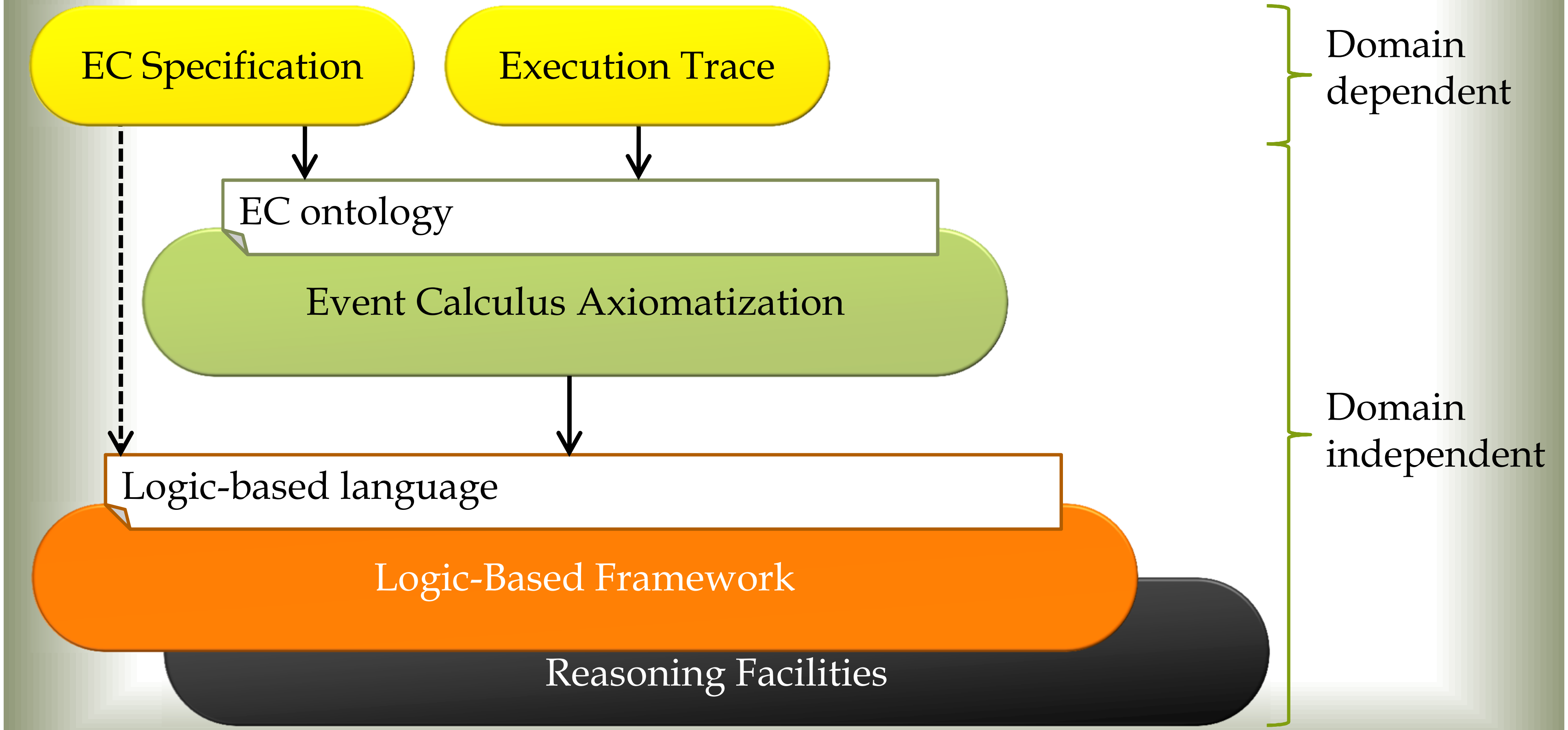
[Kowalski and Sergot, 1986]

- At CILC2009 (Ferrara) we introduced the *Reactive Event Calculus*, an implementation of EC that overcomes some limitations of classical logic-based EC implementations

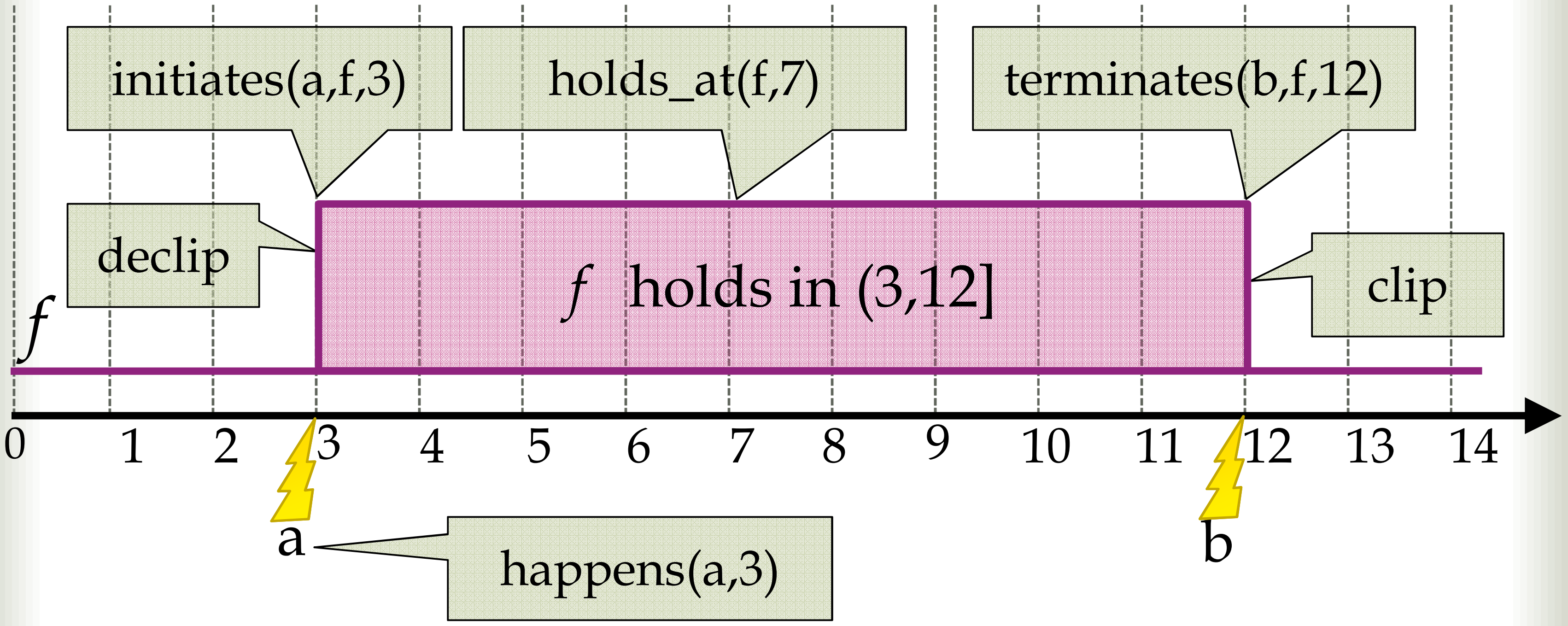
REC allows to:

- Represent the procedural and workflow-related aspects of a CG/BMK
  - See the work [Cicekli and Cicekli, 2006] about representing generic workflows with EC
- Represent the declarative knowledge, thanks to the underlying Prolog knowledge base
- Represent rules and reactive behaviors, thanks to fluents

EC Framework



The Simple EC Ontology



Approach

- Events
  - Life cycle events: **`exec(event(start, A))`**
  - Patient-related events: **`heart_failure`**, **`glucose(91)`**
- Status Fluents
  - **`status(nextCGcandidate, As)`** indicates the next activity (set of activities) according to the CG
    - Control-flow dimension
  - **`status(A, S)`** indicates the current state during the execution of A
    - Activity life cycle
  - **`status(cg,nc)`** indicates the presence of a deviation
- Other EC + Prolog rules to capture BMK
- Facts to describe the structure of the CG and the preconditions of each activity
- Deviations explicitly captured by modeling the notions of “next expected action” directly as a fluent

Capturing BMK

- Obviously, only relevant portions of the BMK can be considered

- *Threats to patient's life must be addressed immediately*
  - Occurrence of a life threat gives raise to an abnormality situation
  - The abnormality situation disappears only if a proper treatment is started

```
initiates(exec(E),abnormality(E),T):- life_threat(E).
terminates(exec(event(start,A)),abnormality(E),T):-
                                           treatment(E,A).
```

- *An hearth failure is a life threat*
- *Diuretic therapy is a possible immediate response for acute heart failure*

```
life_threat(hearth_failure).
treatment(hearth_failure,diuretic_therapy).
```

Capturing Deviations

CG next activity is *B*	Operator starts *A*

```
initiates(exec(event(start,A)),status(cg,nc),T):-
                holds_at(status(nextCGcandidate,B),T), A ≠ B.
```

CG *A* must be discarded	Operator starts *A*

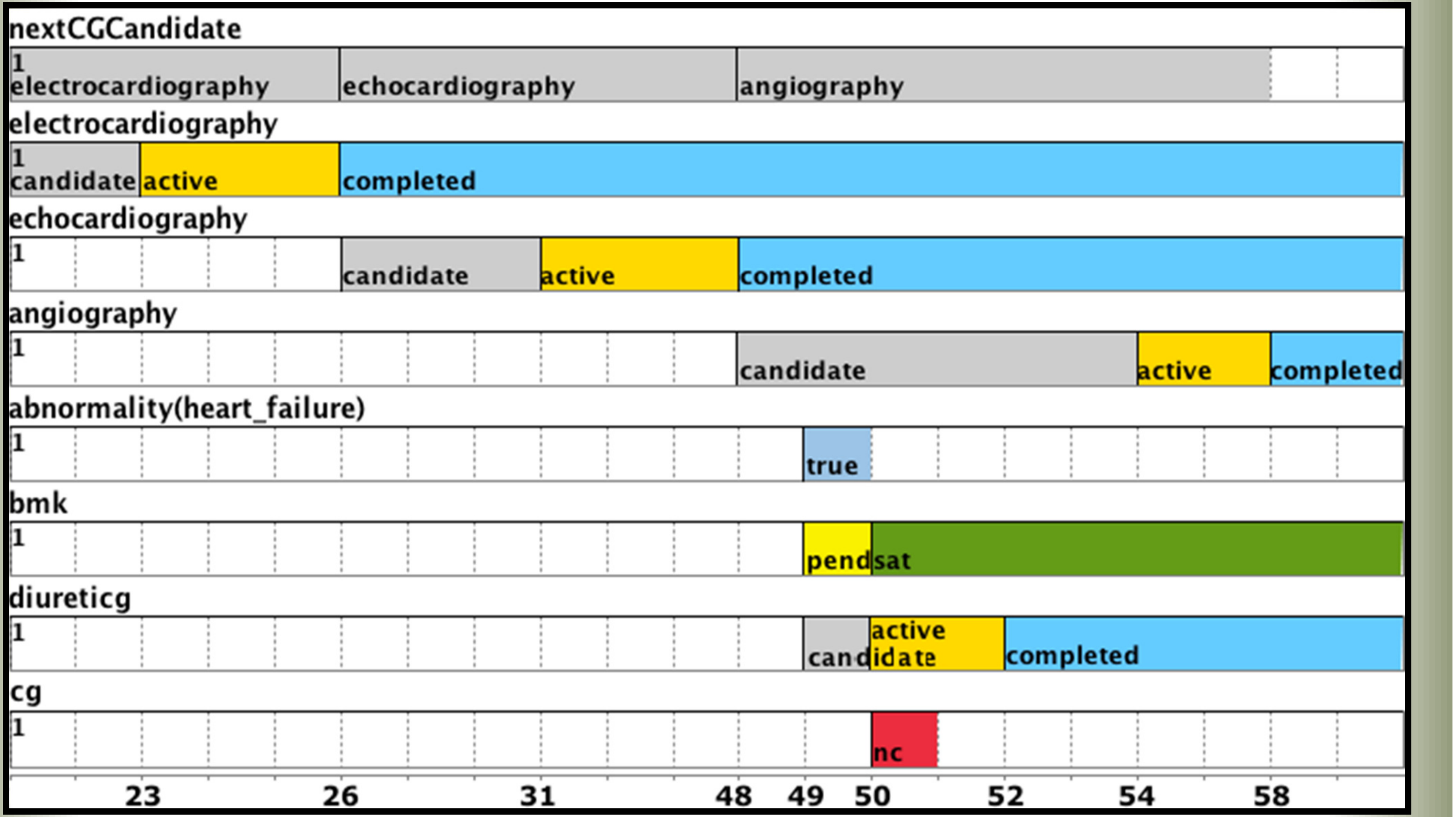
```
initiates(exec(event(start,A)),status(cg,nc),T):-
          holds_at(status(A,candidate),T),¬preconditions(A,T).
initiates(exec(event(start,A)),status(cg,nc),T):-
   holds_at(status(A,candidate),T),holds_at(abnormality(_),T).
```

CG *A* must be started	Operator discards *A*

```
initiates(exec(event(discard,A)),status(cg,nc),T):-
         holds_at(status(A,candidate),T),
         preconditions(A,T),¬holds_at(abnormality(_),T).
```

- Easy to include further cases by adding rules

REC http://www.inf.unibz.it/~montali/tools.html



hearth failure

Conclusions

- Healthcare professionals use BMK to put CGs into practice
- Accommodating the BMK in CG modeling and execution has a lot of implications
  - Specialized activity life cycle
  - Complex interplay between BMK and CG
  - Conformance as a tool for highlighting deviations
- Event Calculus is a suitable framework to formalize CG + BMK

Ongoing/Future Work

- Completing the EC-based formalization of GLARE
- Studying the combination with more declarative approaches such as DECLARE
  - EC-based formalization of DECLARE constraints already available (see the MOBUCON tool)
- Applying REC *during* the execution for clinical operational decision support
- Investigating what happens if we only have “start” and “end” events
  - How to infer which activities have been discarded
  - How and where to “reconnect” the real execution with the CG model after a deviation

Thank you!