SPIN Model Checking for the Verification of Clinical Guidelines

Paolo Terenziani, Laura Giordano, Alessio Bottrighi, Stefania Montani, Loredana Donzella

DI, Univ. Piemonte Orientale, Via Bellini 25, Alessandria, Italy

{laura,terenz,stefania, alessio}@mfn.unipmn.it

Index

- Aim of paper
- GLARE system
- SPIN and PROMELA
- New Architecture
- Mapping GLARE formalism in PROMELA
- The verification task
- Conclusion

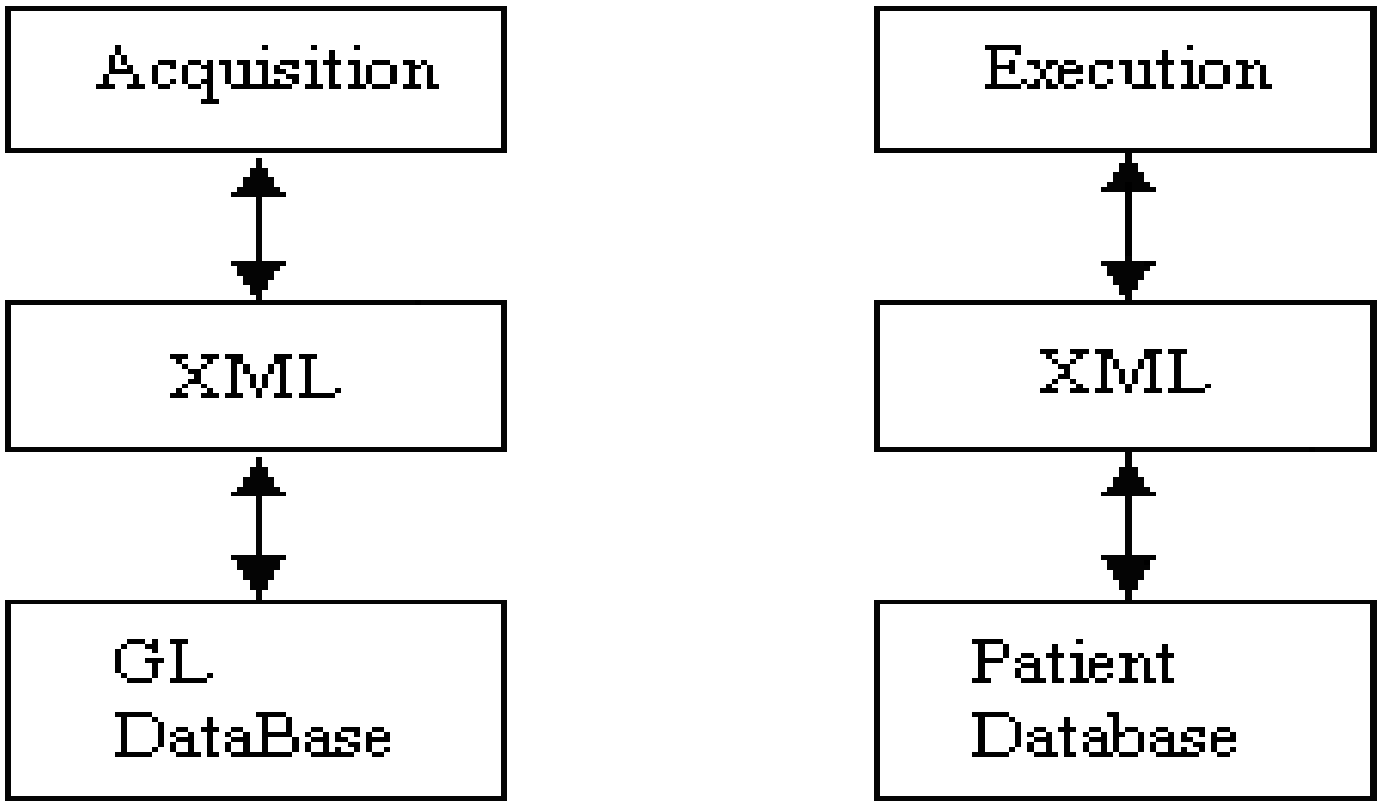
Aim of the paper

- **Model-checking** techniques are successfully used for protocol verification
- The paper aims at showing how *LTL model checking* techniques can be used in the verification of clinical guidelines
- We extend the Glare system with a verification component using the model checker SPIN

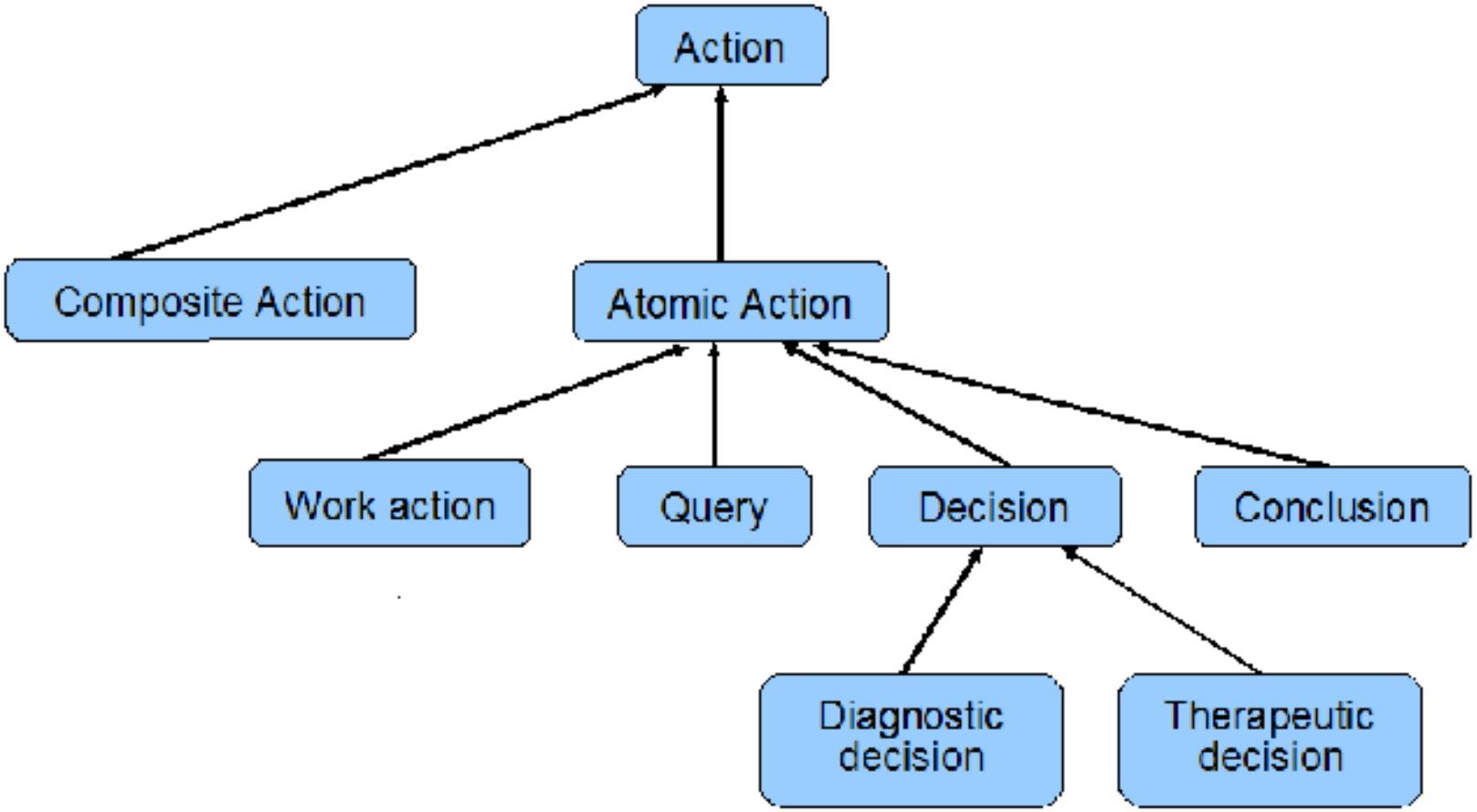
GLARE

- GLARE is a joint-project between the Dept. Comp. Sci. Univ. Alessandria(It), Dept. Comp. Sci. Univ. Torino (It) and Azienda Ospedaliera San Giovanni Battista in Turin (It)
- GLARE is a domain-independent prototypical system for acquiring, representing and executing clinical guidelines

Architecture of the system



Representation Formalism Hierarchy of Action Types



The model checking approach

- In the model checking approach, given
  - a *model* describing all the possible evolutions of the system and
  - a *specification* expressed in a temporal logic

  the model is checked to see whether it satisfies the specification

SPIN

- In the model checker SPIN

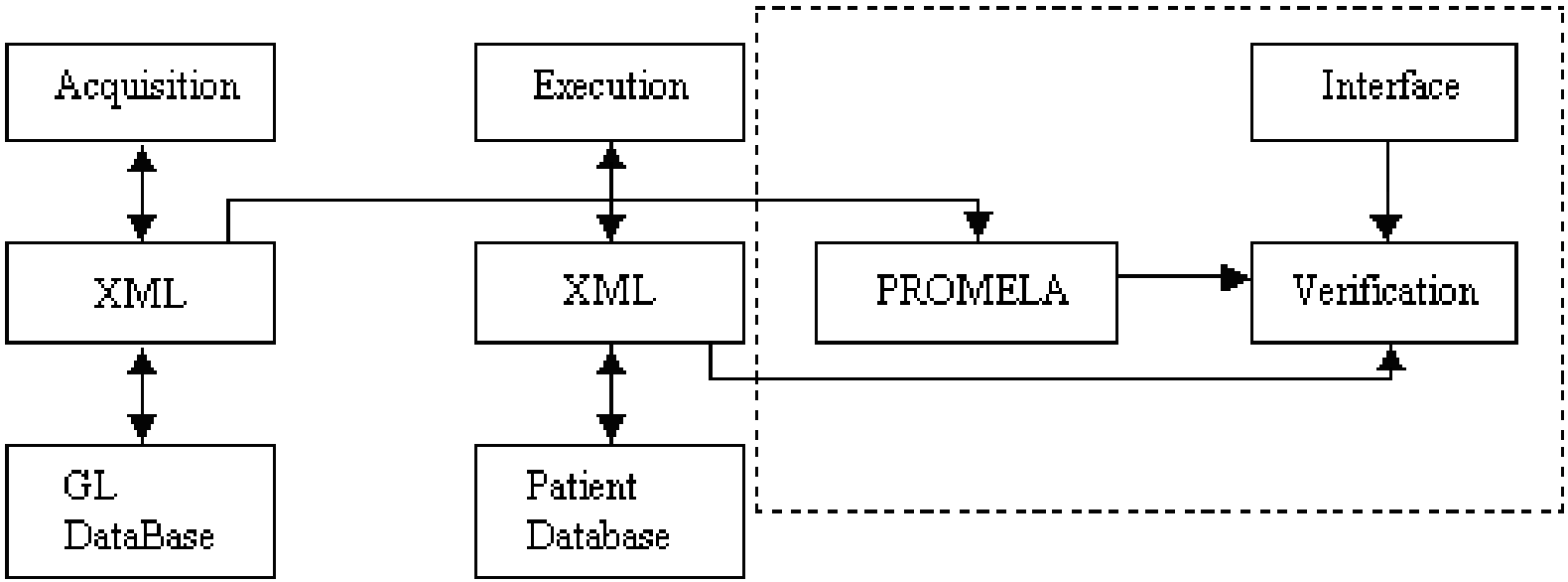
  - the model is given in the input language ***Promela*** and

  - the property to be checked is a formula of the linear time temporal logic (***LTL***).

PROMELA

- PROMELA (a PROcess MEta LAnguage) allows a high level model of a distributed system to be defined: each process is modelled in a pseudo C code, including synchronization primitives.
- The guideline and the agents which interact with it are modelled as Promela processes

New architecture of the system (1)



Representing GLARE clinical guidelines using PROMELA (1)

- The *Guideline* agent models the overall behaviour of the guideline.
- Each construct in the guideline is mapped to a Promela statement or (for complex statements) to a Promela piece of code.

Representing GLARE clinical guidelines using PROMELA (2)

- The *Physician* agent is modelled as a non-deterministic process which interacts with the guideline by evaluating the patient data, choosing among the different alternative feasible paths.

Representing GLARE clinical guidelines using PROMELA (3)

- The *Outside* agent, representing the outside world, provides up to date values for data (together with the time of their measurement) when they are not already available from the database. It also stores data in the database, executes work actions and reports about their success or failure.
- The *Database* agent models the behaviour of the patient database, allowing for data insertion and retrival.

Example of Enquiry in PROMELA (1)

The datum required by the query action is searched for in the database.

If the datum is found, the physician evaluates if it is still reliable.

In this case, the query action is completed

Otherwise, a second interaction between the guideline and the outside world is carried out

Example of Enquiry in PROMELA (2)

```
A:  LGtoDB!data[0].D,data[0].A;
LGfromDB?data[0].D,data[0].A,data[0].V,data[0].T;
   if ::(data[0].V[0] == MISSING)-> {
      LGtoOUTSIDE!data[0].D,data[0].A;
      LGfromOUTSIDE?data[0].D,data[0].A, data[0].V,data[0].T;
         }
   :: else -> {
      LGtoPH!data[0].D,data[0].A,data[0].T;
      LGfromPH?data[0].D,data[0].A, data[0].V,data[0].T;,valid;
      if :: !(valid)->{
         LGtoOUTSIDE!data[0].D,data[0].A;
         LGfromOUTSIDE?data[0].D, data[0].A, data[0].V,data[0].T;
         }
      fi;
   }
   fi;
```

Observations

- The stroke guideline only uses qualitative constraints on the temporal ordering of actions
- The evaluation of temporal constraints is not required during the execution of the guideline
- Timestamps are only used by the Physician who has to decide if they are still reliable
- We only need to represent whether the value of the timestamps is known or not

The Verification Task

- A property which has to be verified is mapped into an LTL formula, as required by SPIN.
- SPIN converts the negation of the temporal formula into a Büchi automaton and computes its synchronous product with the system global state space.
- If the language of the resulting Büchi automaton is empty then the property is true on all the possible executions; otherwise, a counterexample is provided.

The Verification Task: properties (1)

- **Properties concerning a guideline “per se”:** one can check if the guideline contains a path of actions satisfying a given set of properties
- **Properties of a guideline in a given context**: specific contexts of execution may impose several limitations on the executable actions of guidelines

The Verification Task: properties (2)

- **Properties of a guideline when applied to a specific patient:** provided that the model checker has in input all the data in the patient record, the feasibility of a given action, or path of actions on the specific patient can be proved
- **Integrated proofs:** any combination of the above types of proofs is feasible

The Verification Task: example - inconsistencies in the guideline (1)

During the verification of the stroke guideline we have been able to discover some inconsistencies in the original formulation of the guideline.

The Verification Task: example - inconsistencies in the guideline (2)

- If a recovery treatment has been excluded, later on the guideline cannot prescribe it

- Given the LTL formula:

  □ (conclusion ==recovery_treatment_excluded → ¬ ◊ proc_recovery_treatment == started)

SPIN produces a counterexample to this property.

The Verification Task: example - contextualization

- Let us suppose that the angiography is not available in the hospital.
- We want to check if the angiography is eventually required on every execution of the guideline

$$\Diamond(\text{required\_test} == \text{angiography})$$

A counterexample is returned by the model checker

Related Work

- [Marcos 2003; ten Teije 2006] propose a theorem proving approach is to deal with the problem of protocol verification.
- In [S.Bäumler 2006] CTL model checking techniques are used in the verification of the guidelines properties

Semantics

- There is a wide agreement about the importance of providing a clear semantic model for clinical guidelines
- In our approach the semantics of guidelines is provided through their mapping to Promela, by modelling them as automata.

Future Work

- The experimentation of the approach is still ongoing
- As a future work, we are interested in:

  - experimenting the approach on different guidelines

  - developing a more declarative and logical semantics for guidelines