UNIVERSITY OF TRENTO - Italy

Computer-Aided Generation of Enforcement Mechanisms for Error-Tolerant Policies

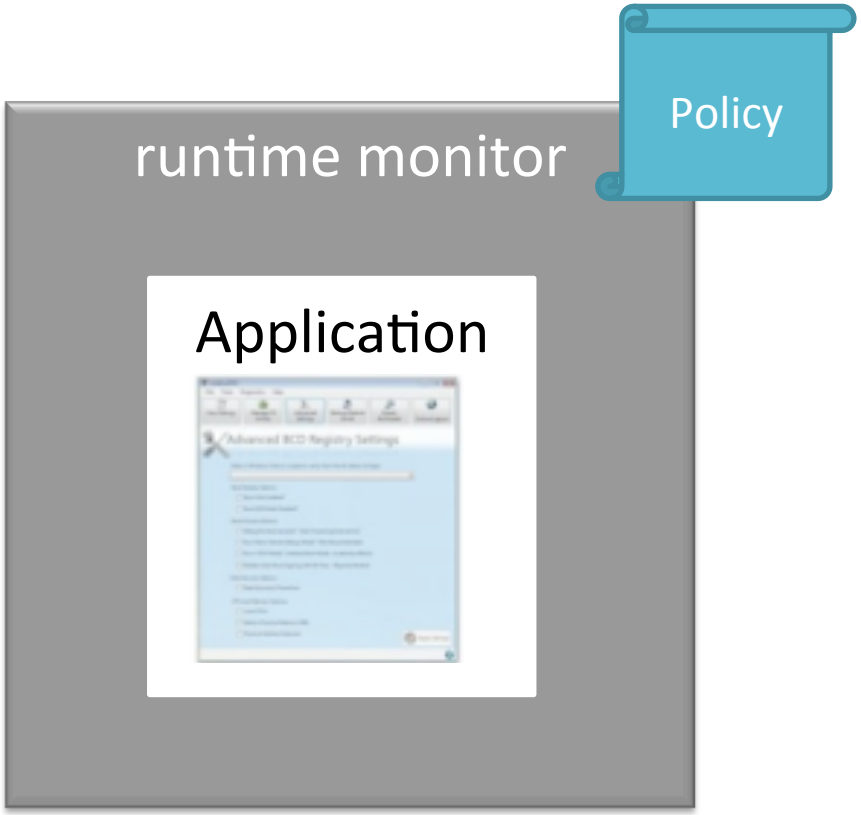
Nataliia Bielova, Fabio Massacci

University of Trento, Italy

POLICY'11

7 June 2011

This talk is about…



In e-health scenario…

- Users:
  - doctors, nurses, pharmacists

- Application:
  - implementing a business process

- Policy:
  - default protocol
    - simple
    - well understood by the users
  - health and privacy regulations

- Runtime monitor enforcing policy

Once the runtime monitor is deployed…

- + it ensures compliance
- - when an insignificant deviation from the protocol happens
  - runtime exception etc. disturbs the users
  - they start being convinced that "The system doesn't work"
  - if such disruptions occur too often, then…
  - the user **will try to bypass** the runtime monitor
- => some of the policy violations might be amended

How to specify such deviations?

- With all possible deviations in the policy?
- No. It will make policy too complicated for the end users!
- The policy should remain simple and well-understood
- Alternative: let runtime monitor
  - enforce simple policies
  - tolerate insignificant deviations

Case study: drug dispensation process from Hospital San Raffaele (Milan)

- high-level business process
- must comply with specific rules (File F), e.g.
  - if the patient is involved in the clinical trial for testing the drug,
    - the doctor should insert a special research protocol number
    - in order to reimburse this drug later by the clinical trial funds.
  - if the drug is highly sensitive,
    - the doctor should review the therapeutical notes

Main steps of the default protocol

- The doctor selects a drug for his patient.
- If the drug is highly sensitive, the doctor reviews therapeutical notes. Otherwise therapeutical notes can be emitted.
- In case the drug is registered for research program, the doctor should insert the research protocol number. Otherwise the doctor skips this step.
- The doctor performs other actions needed for the drug prescription.

Venial errors

- The doctor forgot to review the therapeutical notes
- (he closes the window instead of clicking “Done”)
- This error is called a **venial error**
- But the doctor should not be allowed to violate the policy systematically
- it should be allowed a limited number of times, for example *k* times per day

Amendable errors

- The doctor did not insert the research protocol number for the research drug
- Then the whole reimbursement process for this drug can go wrong!
- This should be "corrected" by inserting another special number that will be used during audit
- This error is called an **amendable error**
- This number of corrections should be limited as well

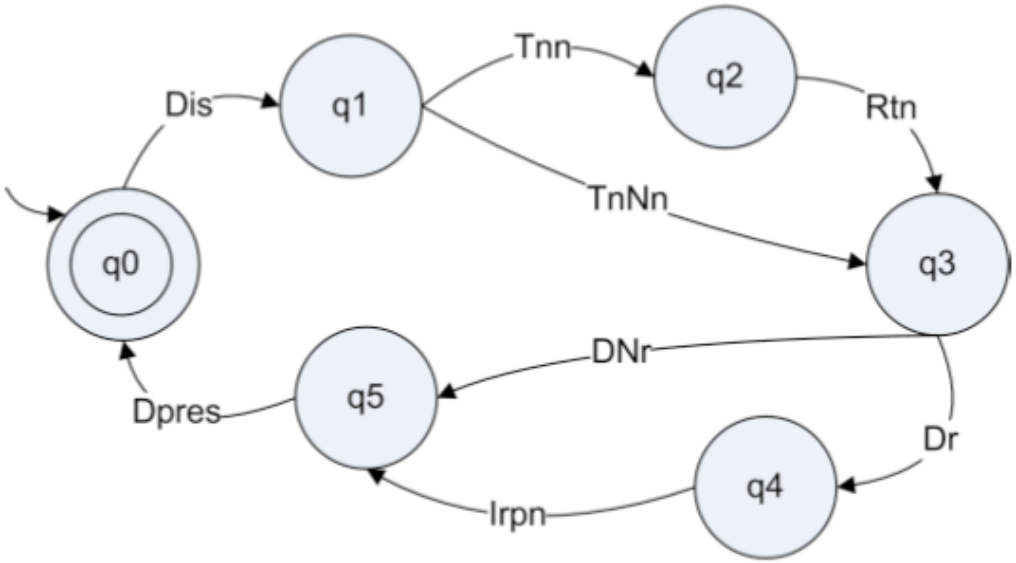
More formally…

- Deviations are given with expected *ex* and correcting *c* functions

Deviation *e*	Expected action *ex(e)*	Correction *c(e)*
The doctor did not review the therapeutical notes	The doctor reviews therapeutical notes	The doctor did not review the therapeutical notes
The doctor did not insert the research protocol number for the research drug	The doctor inserts the research protocol number	Insert a special number for auditing

More formally…

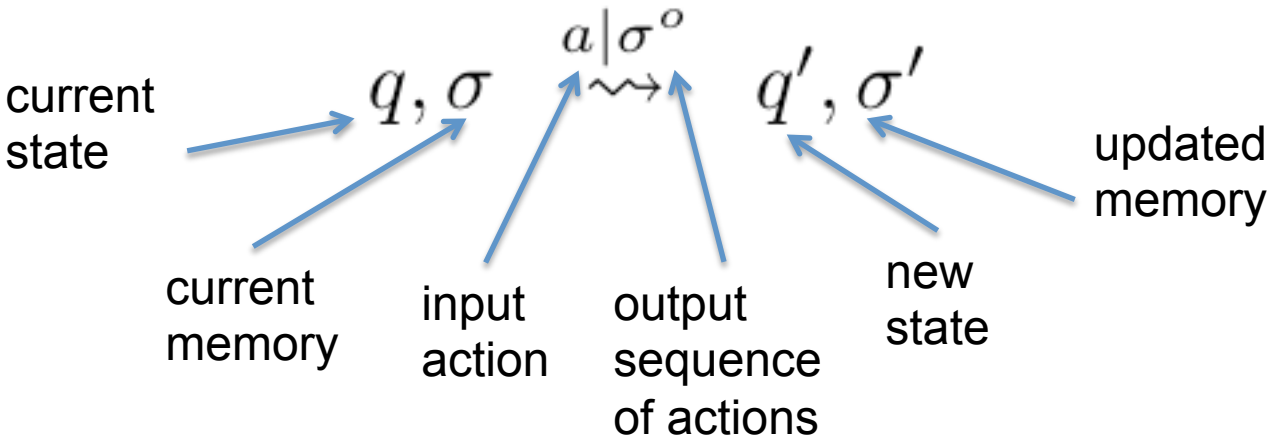
- Default protocol represented by a finite state automaton (policy automaton)



- Number of possible errors *k*

Runtime monitor model

- Execution transformer:
  - **edit automaton (EA) (+ memory)**
  - can insert and suppress actions



Automatic construction of edit automaton

- Automatic construction:
  - **Input**: default protocol, number of errors *k,* functions for deviations *ex* and *c*,
  - **Output**: edit automaton

- State is augmented with an error counter *s*
- if an error *e* is venial (*c(e) = e*), accept *e* and increase *s*
- if an error *e* is amendable (*c(e) ≠ e*), accept *c(e)* and increase *s*
- when an accepting state of default protocol is reached, reset the couner *s*

Automatic construction of edit automaton

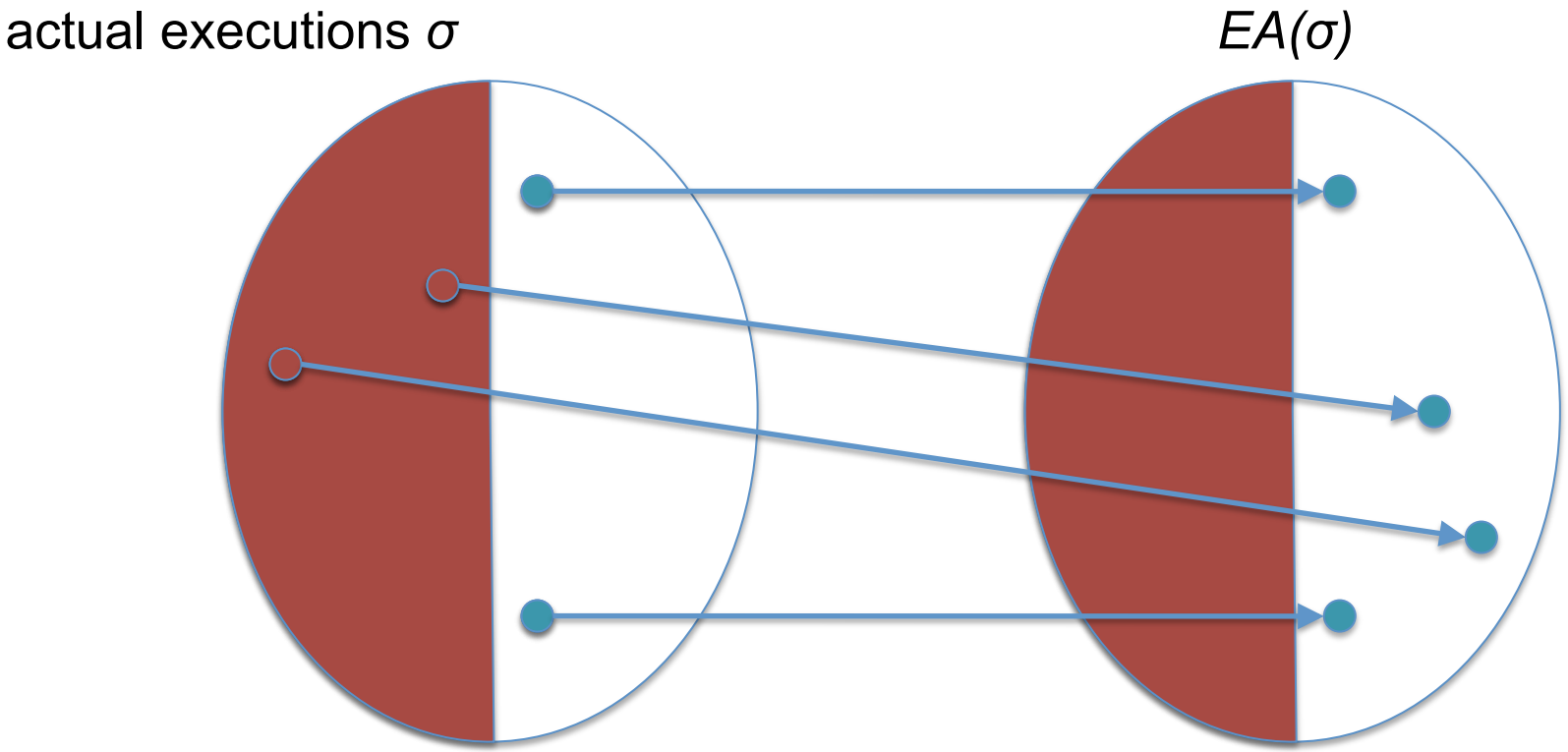
s	user action e	is allowed?	s<k	c(e) =e?	result	s’
0	The doctor selects the drug	yes			proceed to next state with e	0
0	The doctor did not review the therapeutical notes	no	yes	yes	proceed to next state with e	1
1	The doctor did not insert the research protocol number for the research drug	no	yes	no	proceed to next state with c(e)	2
2	*another error*	no	no	yes/no	move to error state and reset the counter	0

Formal properties: soundness & transparency

- Assume:
  - σ is an actual execution
  - $P(\sigma)$ is true if an actual execution complies with the default protocol
  - $EA(\sigma)=\sigma'$ edit automaton transforms an actual execution

Soundness & transparency

- An enforcement mechanism EA is **sound** with respect to *P* if for all *σ: P(EA(σ))*
- An enforcement mechanism EA is **transparent** with respect to *P* if for all *σ: P(σ) => EA(σ)=σ*



Transparency

- Theorem 1 [Transparency]

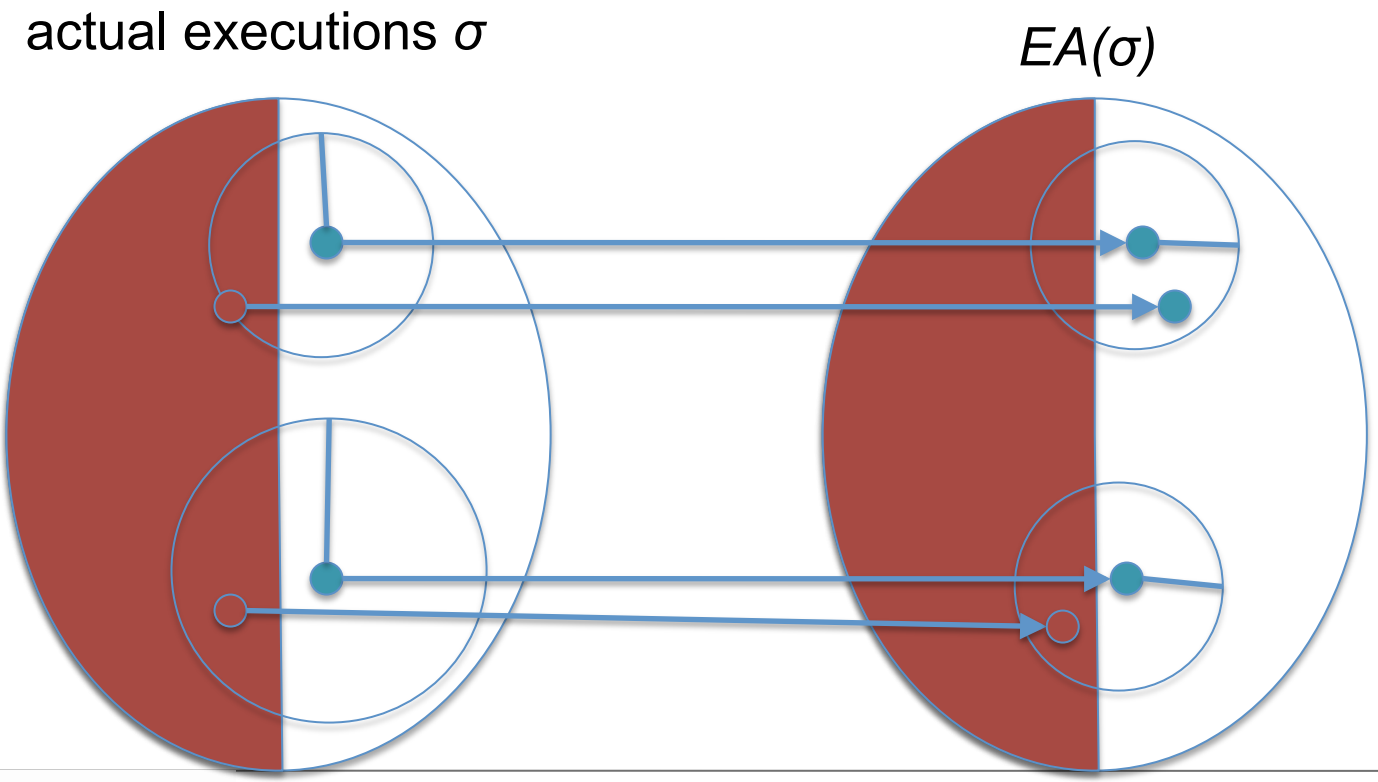
  Given <*P, k, c, ex*>, the constructed enforcement mechanism EA is **transparent** with respect to *P.*

Formal properties: predictability

- Predictable within ε
  - if the execution σ is close to a valid execution σ', then *EA* *(σ)* should be close to σ'

Formal properties: predictability

- Predictable within ε
  - if the execution σ is close to a valid execution σ', then *EA* *(σ)* should be close to σ'



Formal properties: predictability

- Predictable within ε
  - if the execution σ is close to a valid execution σ', then *EA* (σ) should be close to σ'
- Metrics to define "close" relation:
  - replacing distance *d* between executions
- Definition:
  - An enforcement mechanism EA is **predictable within ε** with respect to *P,* if for all σ' such that $P(\sigma')$:

    for all $v \geq \varepsilon$ *exists* $\delta > 0$ *such that for all* σ*:*

    $d(\sigma,\sigma') \leq \delta \Rightarrow d(EA(\sigma), EA(\sigma')) \leq v$

Predictability

- Theorem 2 [Predictability]

  Given <*P, k, c, ex*>, a constructed enforcement mechanism EA is **predictable within k.**

Conclusions

- Policies must stay simple
- Some deviations should be tolerated
- An enforcement mechanism modeled as edit automata automatically constructed for it
- Proven to be transparent and k-predictable (for tolerating k errors)