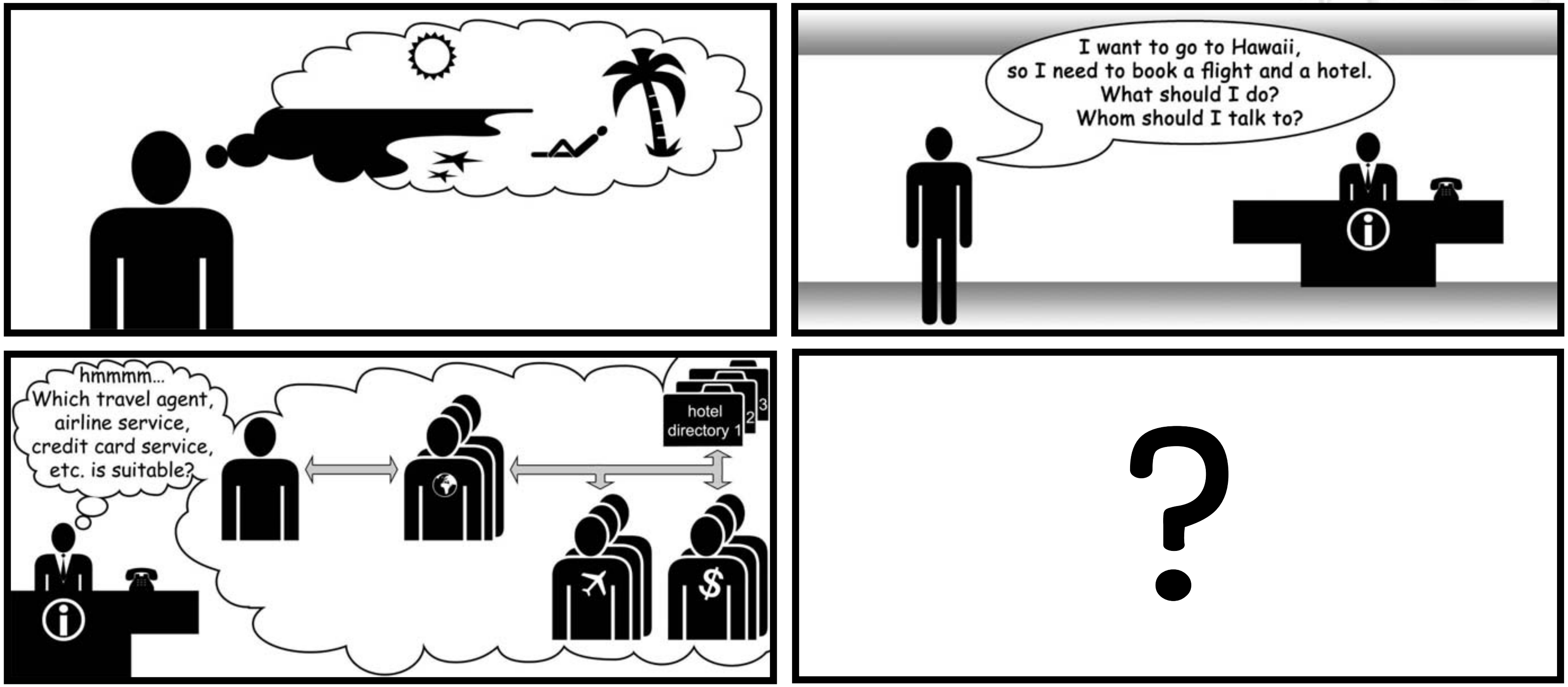
Run–Time Model Checking of Interaction and Deontic Models for Multi–Agent Systems

Nardine Osman

May 2006

Motivating Example



Overview

Goal: achieving predictable, reliable interactions.

From the available literature, two approaches have emerged:

- The use of explicit models of interaction (in a generic process or state-machine language) to describe forms of interaction.
- The specification of deontic constraints: constraints imposed by individual agents on the actions they will or will not allow.

Problem: Given an interaction model and an agent with given deontic constraints wishing to participate in that model, could that combination work?

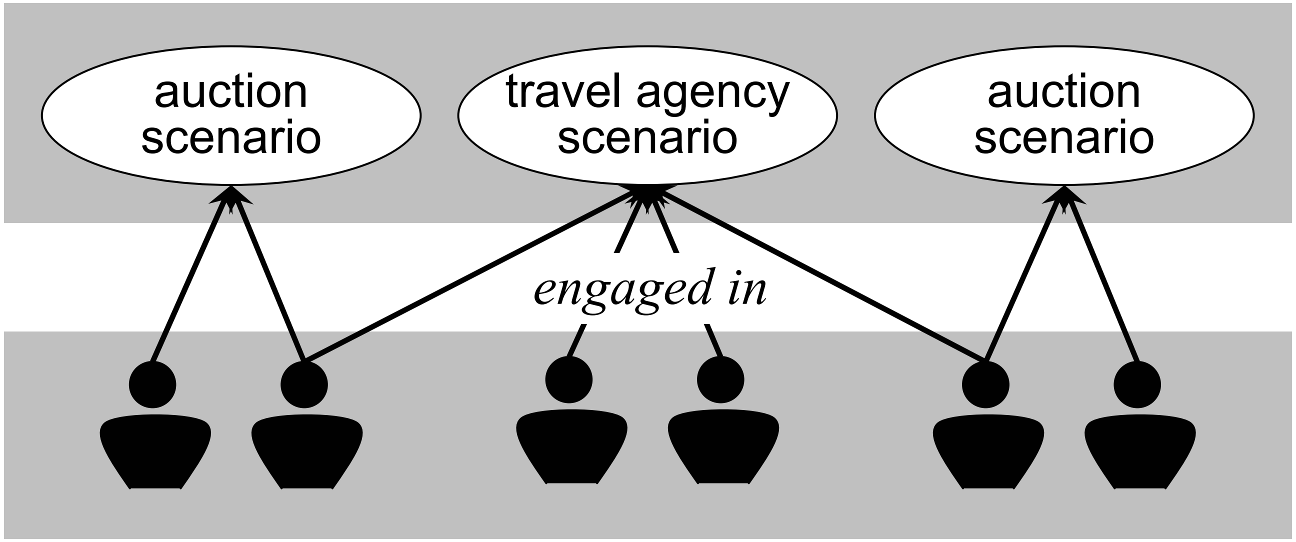
Solution: We use a dynamic model checker to verify that a given combination of interaction and deontic models works.

In this talk…

- Modeling Multi-Agent Systems
- Verifying Multi-Agent Systems
- The verification Process

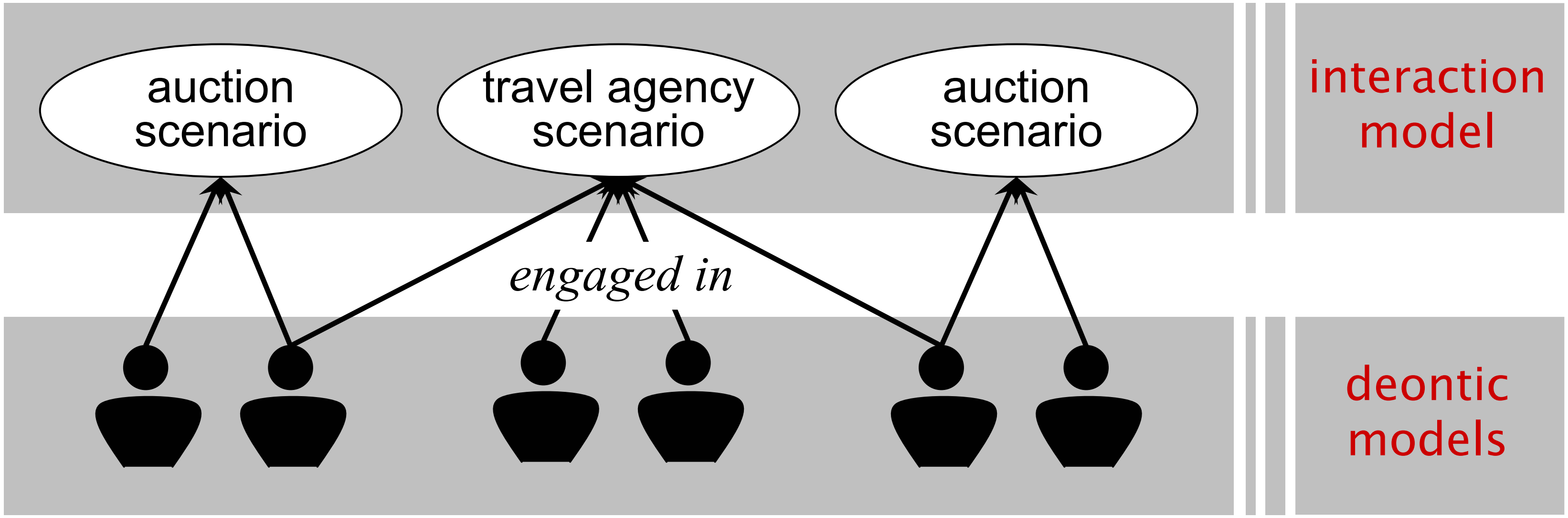
Modeling Multi-Agent Systems

- MAS are composed of interacting agents.
- Different agents are grouped into different (and possibly multiple) interactions.
- Interaction groups are created dynamically and automatically by the agents.



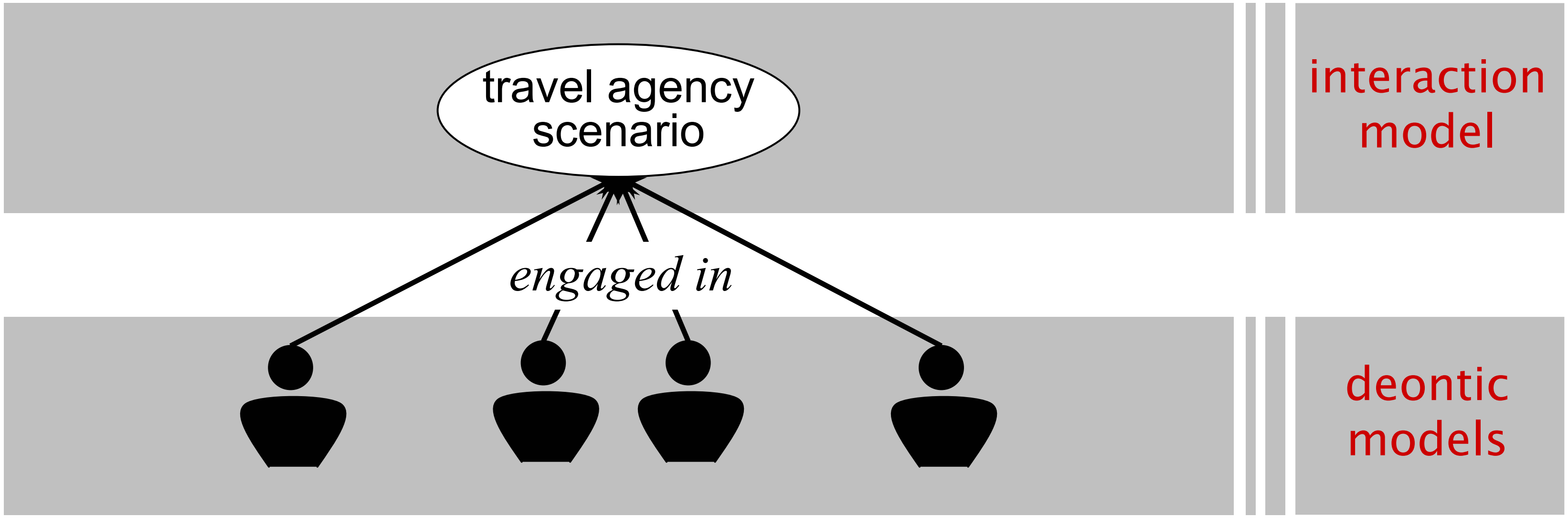
Modeling Multi-Agent Systems

- MAS are modeled on two layer: interaction layer and agents layer
  - Interaction model: specifies rules on the interaction
  - Agent model: specifies rules on the agents



Modeling Multi-Agent Systems

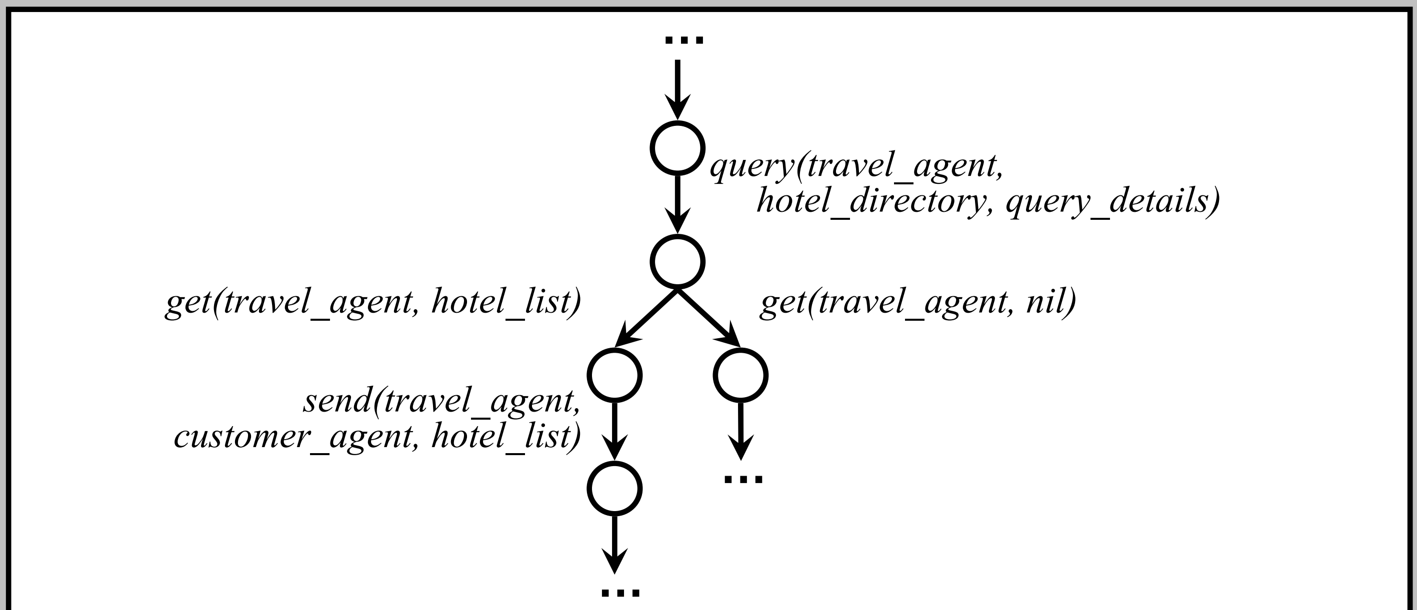
- For one scenario, there is one interaction model, and several deontic models (one for each agent).



Modeling Multi-Agent Systems: Example

SYSTEM MODEL

interaction model



deontic model

can(travel_agent, query(hotel_directory)) $\leftarrow$

can(travel_agent, access(hotel_directory)).

can(customer_agent, $\neg$ *pay_with(master_card)).*

...

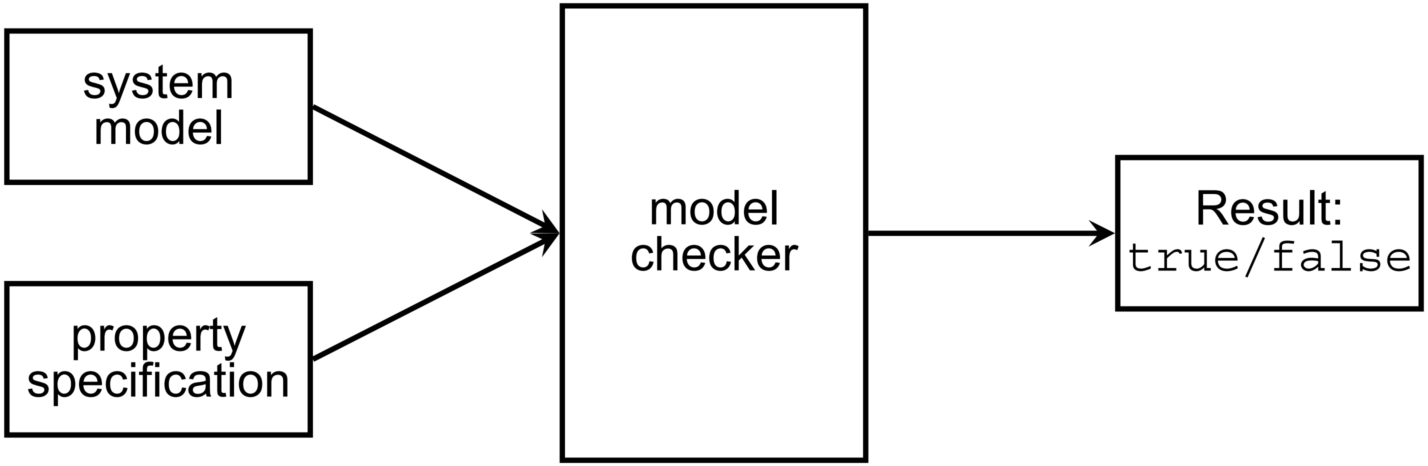
Verifying Multi-Agent Systems

- In open systems consisting of autonomous agents, it is necessary for agents to be capable of automatically verifying, at run-time, dynamic protocols affected by dynamic deontic rules.

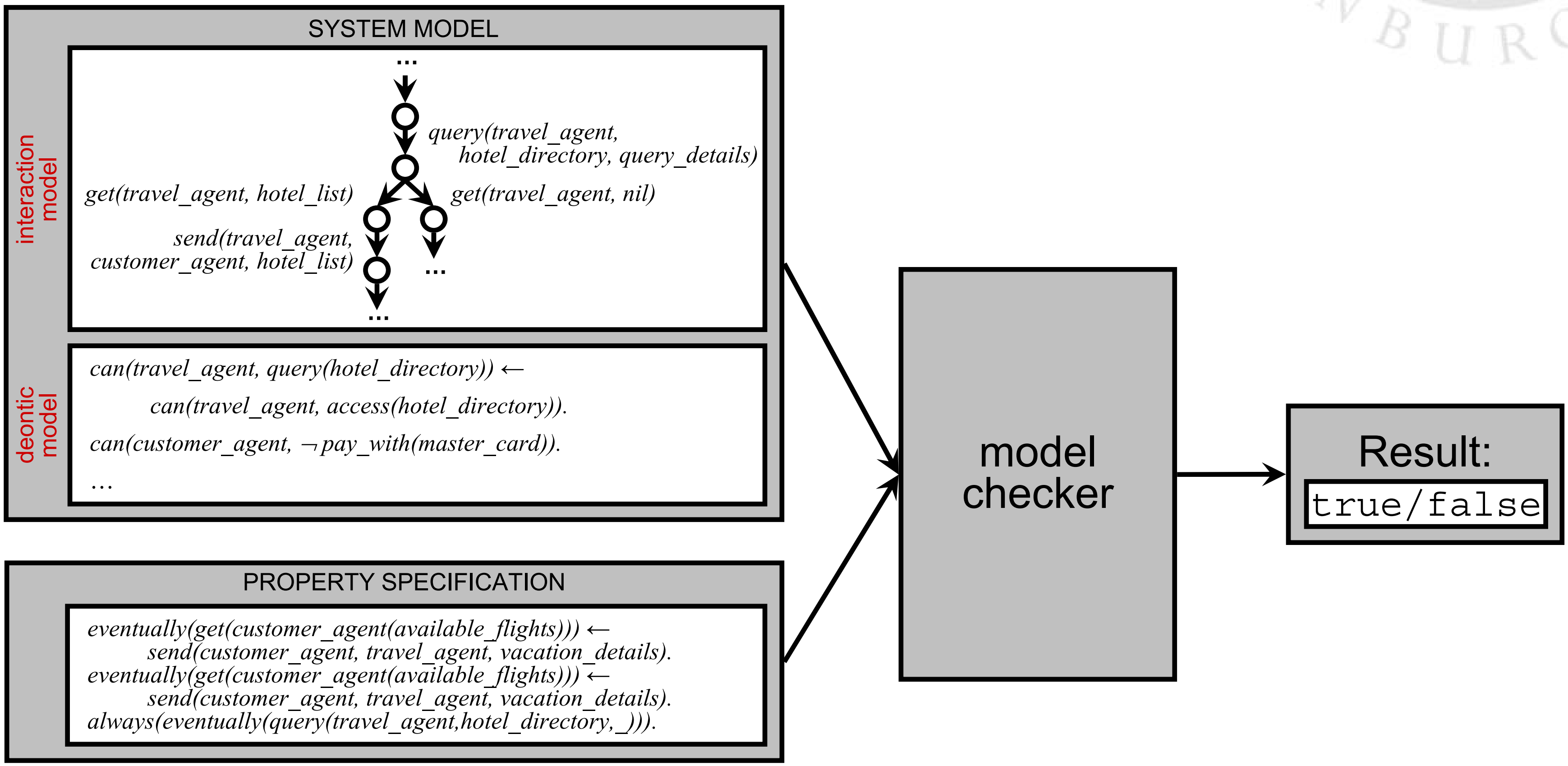
  $\Rightarrow$ use model checking since it provides a fully automatic verification process.

- Model Checking Problem:

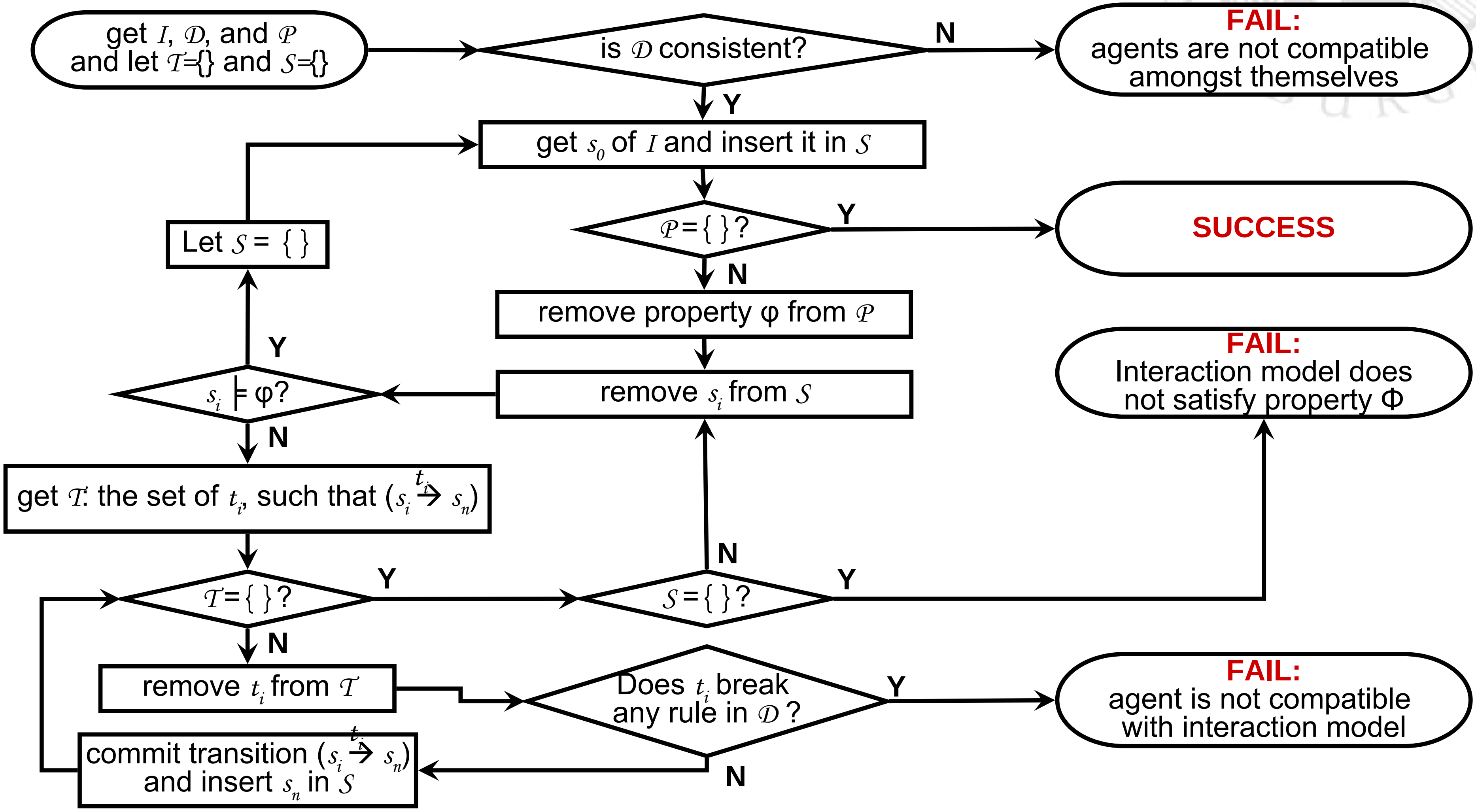
  Given a finite transition system S and a temporal formula ϕ, does S satisfy ϕ?



Verifying Multi-Agent Systems: Example



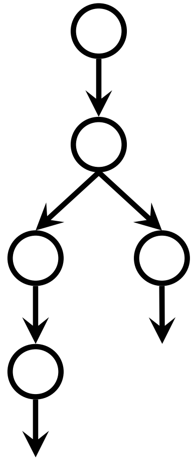
Verification Process



Verification Process

Model checking is applied to the state-space of system model.

- Local model checking:
  - State-space:
    - not generated before verification
    - generated and traversed at run-time (one step at a time)
  - Satisfaction of property : verified at initial node
    - if result is achieved, then terminate
    - else, make a transition(s) to next state(s)
      - if the transition violates a deontic (or trust) rule, then terminate with failure
      - else, repeat with respect to the new state(s)



Verification Process

Satisfaction is verified according to the mu-calculus proof rules:

$satisfies(E, \mathrm{tt})$	$\leftarrow$	$\mathtt{true}$
$satisfies(E, \phi_1 \vee \phi_2)$	$\leftarrow$	$satisfies(E, \phi_1) \vee satisfies(E, \phi_2)$
$satisfies(E, \phi_1 \wedge \phi_2)$	$\leftarrow$	$satisfies(E, \phi_1) \wedge satisfies(E, \phi_2)$
$satisfies(E, \langle A \rangle \phi)$	$\leftarrow$	$\exists F. \; ((E \xrightarrow{A} F) \; \wedge \; satisfies(F, \phi))$
$satisfies(E, [A]\phi)$	$\leftarrow$	$\forall F. \; ((E \xrightarrow{A} F) \; \rightarrow \; satisfies(F, \phi))$
$satisfies(E, \mu Z.\phi)$	$\leftarrow$	$satisfies(E, \phi)$
$satisfies(E, \nu Z.\phi)$	$\leftarrow$	$dual(\phi, \phi') \wedge \neg satisfies(E, \phi')$

Verification Process

Transitions follow the LCC transition rules:

$$\frac{}{M \Leftarrow A \xrightarrow{in(M)} nil} \qquad \frac{}{M \Rightarrow A \xrightarrow{\overline{out}(M)} nil} \qquad \frac{}{null \xrightarrow{\#} nil}$$

$$\frac{}{(A \leftarrow C) \xrightarrow{\#(X)} A}\ sat(C) \wedge X\ in\ C \qquad \frac{A \xrightarrow{a} E}{(A \leftarrow C) \xrightarrow{a} E}\ sat(C) \wedge (a \neq \#/\_)$$

$$\frac{B \xrightarrow{a} E}{A \xrightarrow{a} E}\ A ::= B \qquad \frac{A \xrightarrow{a} E}{A\ or\ B \xrightarrow{a} E} \qquad \frac{B \xrightarrow{a} E}{A\ or\ B \xrightarrow{a} E}$$

$$\frac{A \xrightarrow{a} E}{A\ par\ B \xrightarrow{a} E\ par\ B} \qquad \frac{B \xrightarrow{a} E}{A\ par\ B \xrightarrow{a} A\ par\ E} \qquad \frac{A \xrightarrow{a} E \quad B \xrightarrow{\overline{a}} F}{A\ par\ B \xrightarrow{\tau} E\ par\ F}$$

$$\frac{A \xrightarrow{a} nil}{A\ then\ B \xrightarrow{a} B} \qquad \frac{A \xrightarrow{a} E}{A\ then\ B \xrightarrow{a} E\ then\ B}\ E \neq nil$$

Conclusion

- Verifying MAS:
  - through the verification of interaction and deontic models
  - using a dynamic model checker
- This permits the verification of:
  - errors within the interaction protocol
  - conflicts between the interaction protocol and agents' constraints
  - conflicts between the agents

References

- N. Osman, D. Robertson, C. Walton. *Run-Time Model Checking of Interaction and Deontic Models for Multi-Agent Systems*, AAMAS'06.
- D. Robertson. *A lightweight coordination calculus for agent social norms*, DALT'04.
- C. R. Ramakrishnan, I. V. Ramakrishnan, Scott A. Smolka, Yifei Dong, Xiaoqun Du, Abhik Roychoudhury, V. N. Venkatakrishnan. *XMC: a logic-programming-based verification toolset*, CAV 2000.