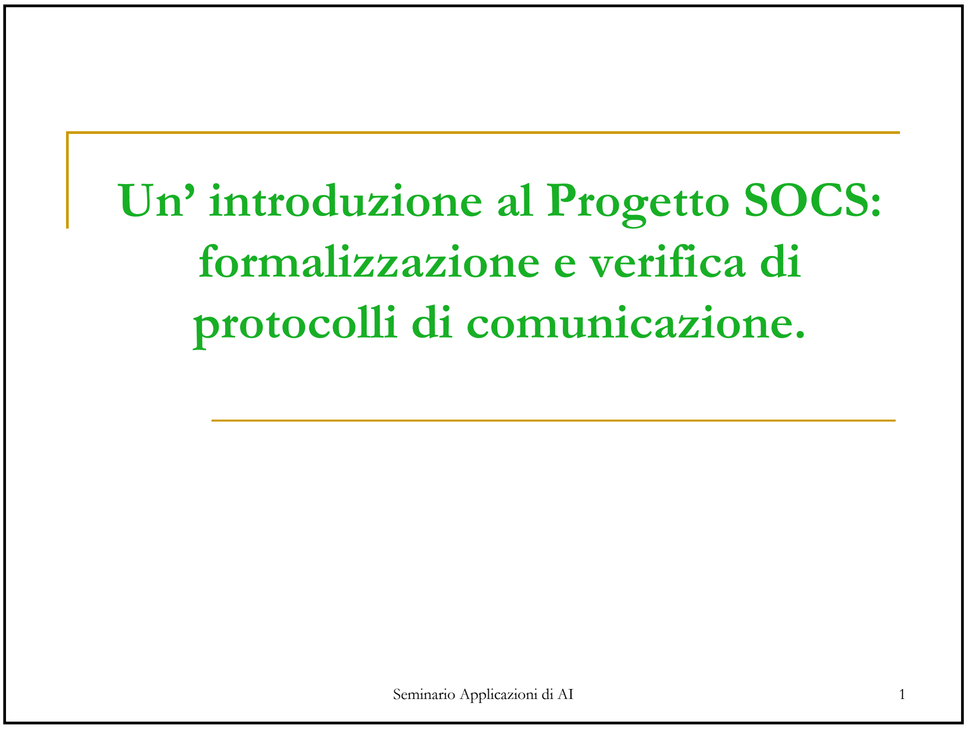
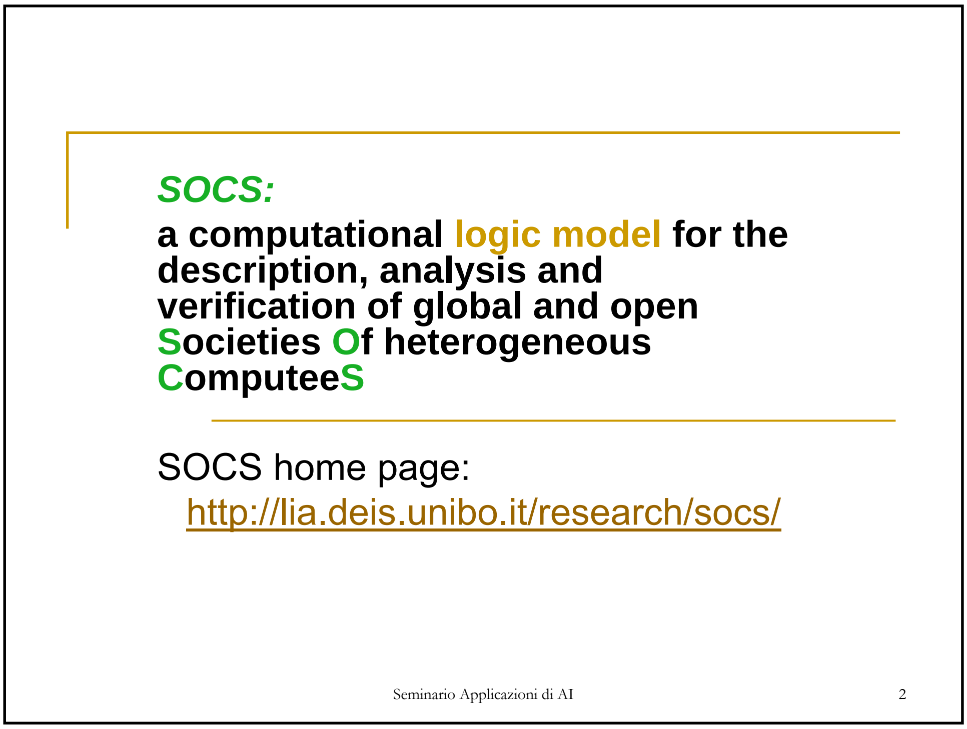


Un’ introduzione al Progetto SOCS: formalizzazione e verifica di protocolli di comunicazione.



SOCS:

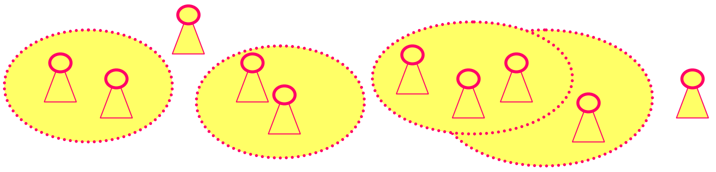
a computational logic model for the description, analysis and verification of global and open Societies Of heterogeneous ComputeeS

SOCS home page:
http://lia.deis.unibo.it/research/socs/

Societies Of ComputeeS (SOCS)

- 3 years project (end June 2005
- Funded by EU
- Partners:
  - University of Bologna
  - University of Ferrara
  - University of Pisa
  - University of Cyprus
  - Imperial College London
  - City University London

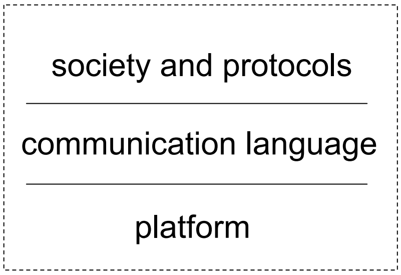
SOCS: Agents in a society



- Societies are groups of interacting agents (possibly with a common goal)
  - interactions are unconstrained (openness)
  - the semantics of interaction is defined in terms of protocols and expectations
  - the 'social' behaviour of computees can be observed
    - to give it an institutional meaning
    - to verify compliance to the protocols
    - to raise expectations, violations, sanctions
- We will focus now on social aspects

Basic architecture

- Data structures:
  - *SOKB* (Social Organization Knowledge Base)
  - *SEKB* (Social Environment Knowledge Base)
  - Social Integrity Constraints
  - Goals
- Roles (duties and capabilities)
- Entry / exit rules
- Semantics of interaction
  - at the protocol level
  - at the communication level
- Verification of interaction



society and protocols

communication language

platform

layered architecture

Social events

Happened events come from the outer world.

Happened events that are "socially relevant" are recorded by the society infrastructure

H*(Event[,Time])*

Once they are recorded, they become part of the SEKB

The history of happened events is the set

HAP = { H*(Event,Time)* }

Expectations

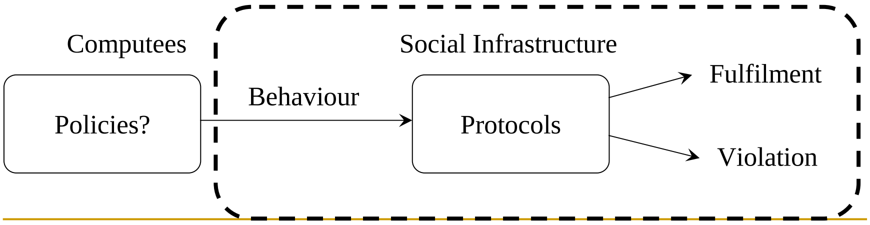
- Events can give raise to expectations in the society:

  E*(Event[,Time])*

  EN*(Event[,Time])*
- *Positive* expectations: events that are expected to occur
- *Negative* expectations: events that are *expected not to occur*
- The state of expectations is a conjunction EXP containing
  - literals of the form (¬)E*(Event,Time)*
  - literals of the form (¬)EN*(Event,Time)*
  - *Constraint Formulas* on the variables occurring in the other literals present in EXP

Protocols

- Computees behave according to their own policies
- Social expectations can be used:
  - to check the correct functioning of the society
  - to suggest to the computees a course of actions
- Protocols are defined through Social Integrity Constraints:
- The society generates expectations out of protocols & events



Computees

Social Infrastructure

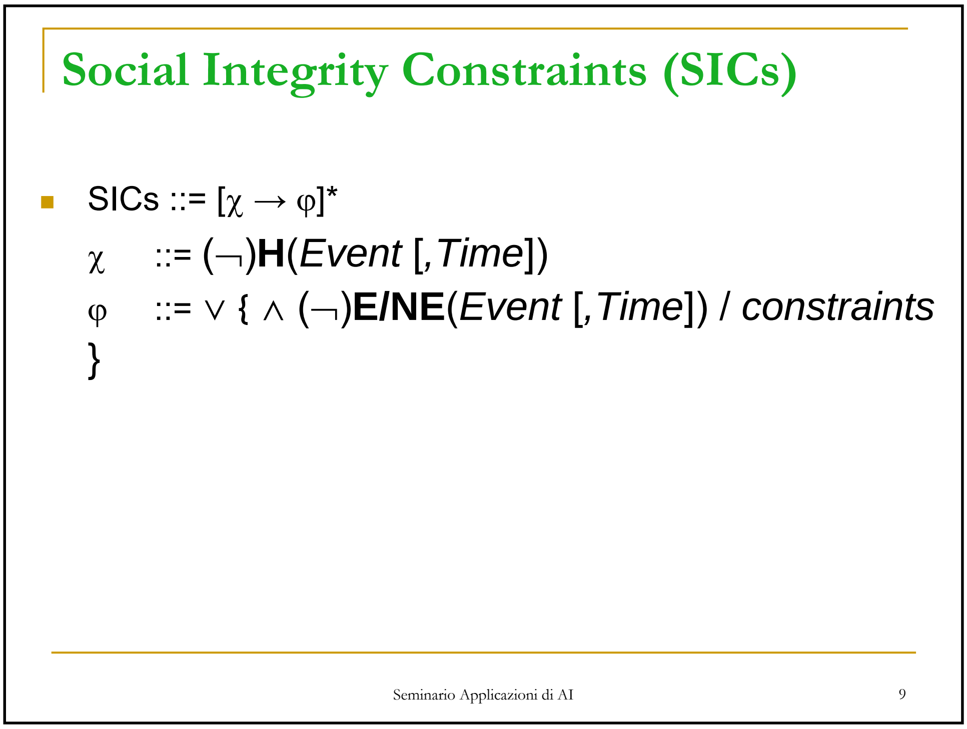
Policies?

Behaviour

Protocols

Fulfilment

Violation

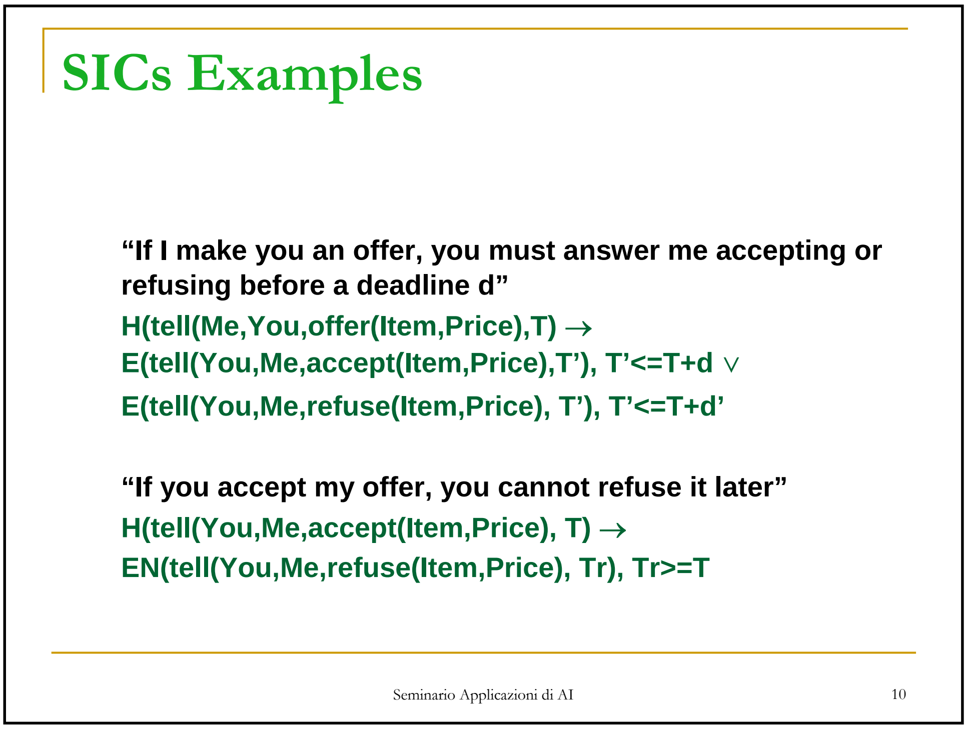


Social Integrity Constraints (SICs)

- SICs ::= $[\chi \rightarrow \varphi]^*$

χ ::= $(\neg)$**H**(*Event* [,*Time*])

φ ::= $\vee$ { $\wedge$ $(\neg)$**E/NE**(*Event* [,*Time*]) / *constraints* }



SICs Examples

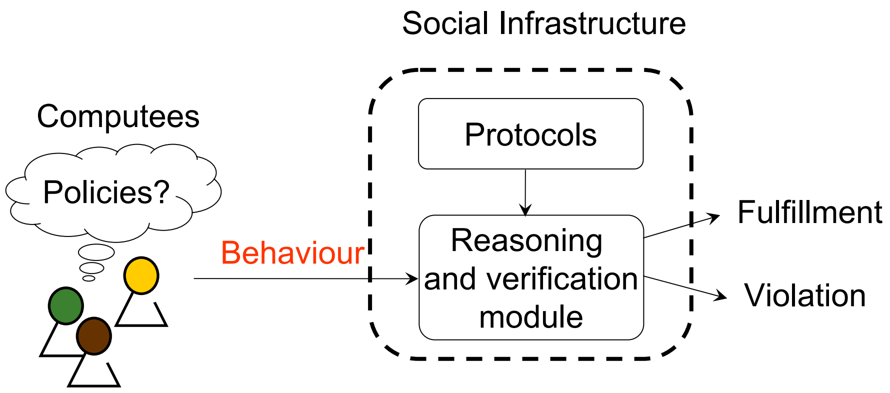
"If I make you an offer, you must answer me accepting or refusing before a deadline d"

H(tell(Me,You,offer(Item,Price),T) →
E(tell(You,Me,accept(Item,Price),T'), T'<=T+d ∨
E(tell(You,Me,refuse(Item,Price), T'), T'<=T+d'

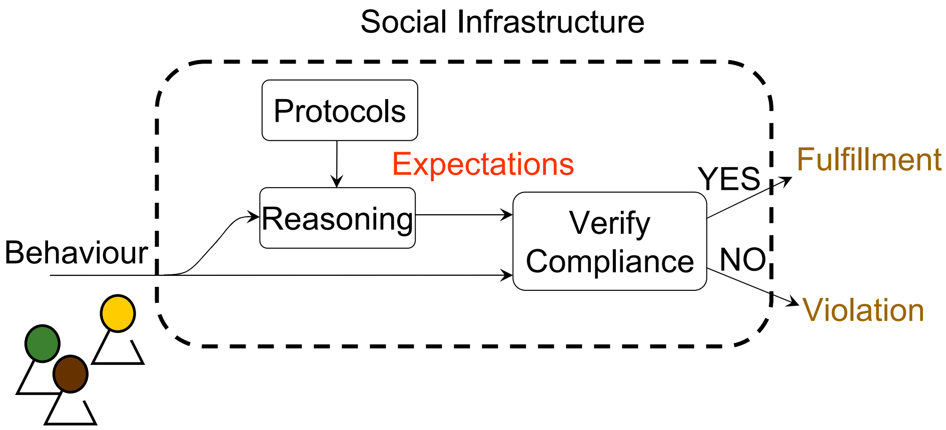
"If you accept my offer, you cannot refuse it later"

H(tell(You,Me,accept(Item,Price), T) →
EN(tell(You,Me,refuse(Item,Price), Tr), Tr>=T

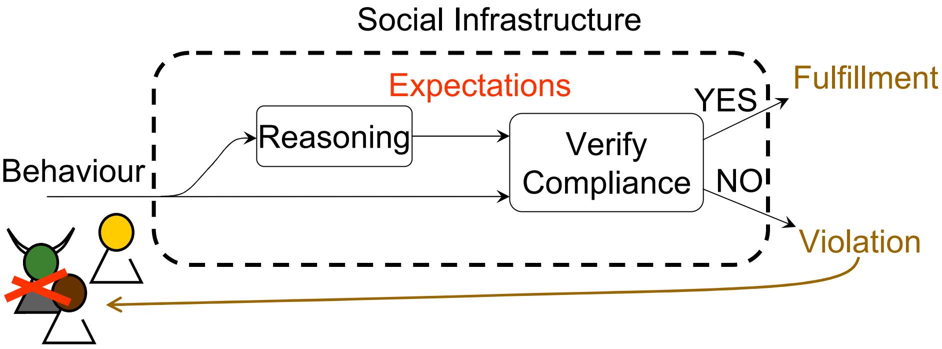
Compliance Verification



Social infrastructure



Social infrastructure



Social Infrastructure

Expectations

Behaviour

Reasoning

Verify Compliance

YES

NO

Fulfillment

Violation

(1)on-the fly verification of compliance to protocols

Social Integrity Constraints (ICs)

- Example of Social Integrity Constraint

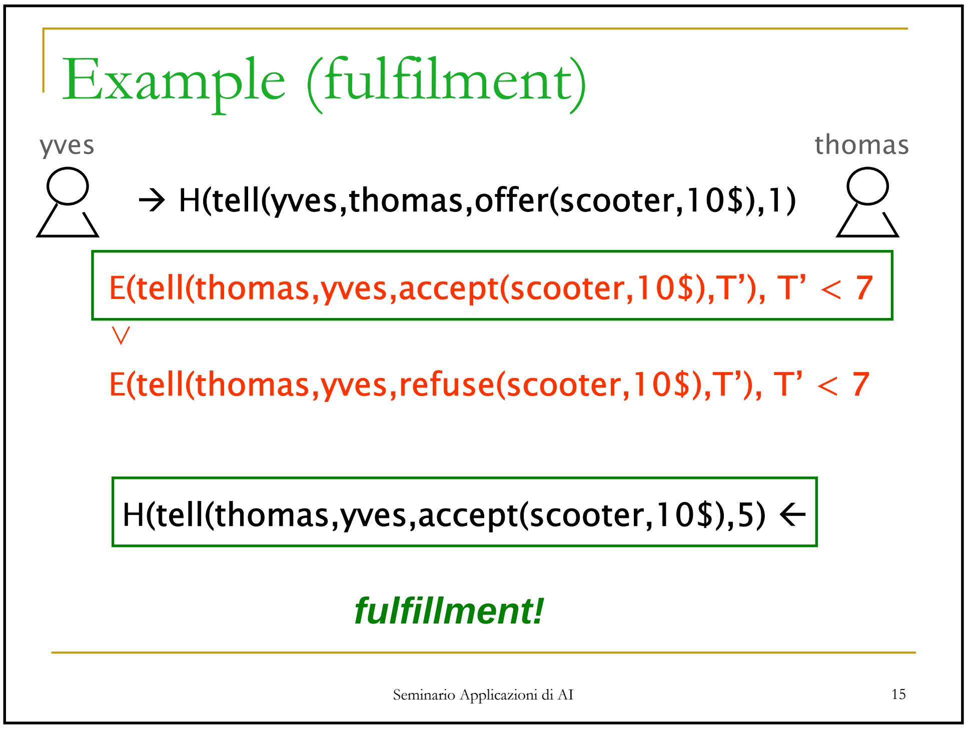
  Society where agents can exchange resources:

  If I make you an offer, you are expected to answer to me by either accepting or refusing before a deadline d

  H(tell(Me,You,offer(Item,Price),T) →
  E(tell(You,Me,accept(Item,Price),T'), T'<=T+d ∨
  E(tell(You,Me,refuse(Item,Price), T'), T'<=T+d'

  If you accept my offer, you are expected to not refuse it later

  H(tell(You,Me,accept(Item,Price), T) →
  EN(tell(You,Me,refuse(Item,Price), Tr), Tr>=T



Example (fulfilment)

yves

thomas

→ H(tell(yves,thomas,offer(scooter,10$),1)

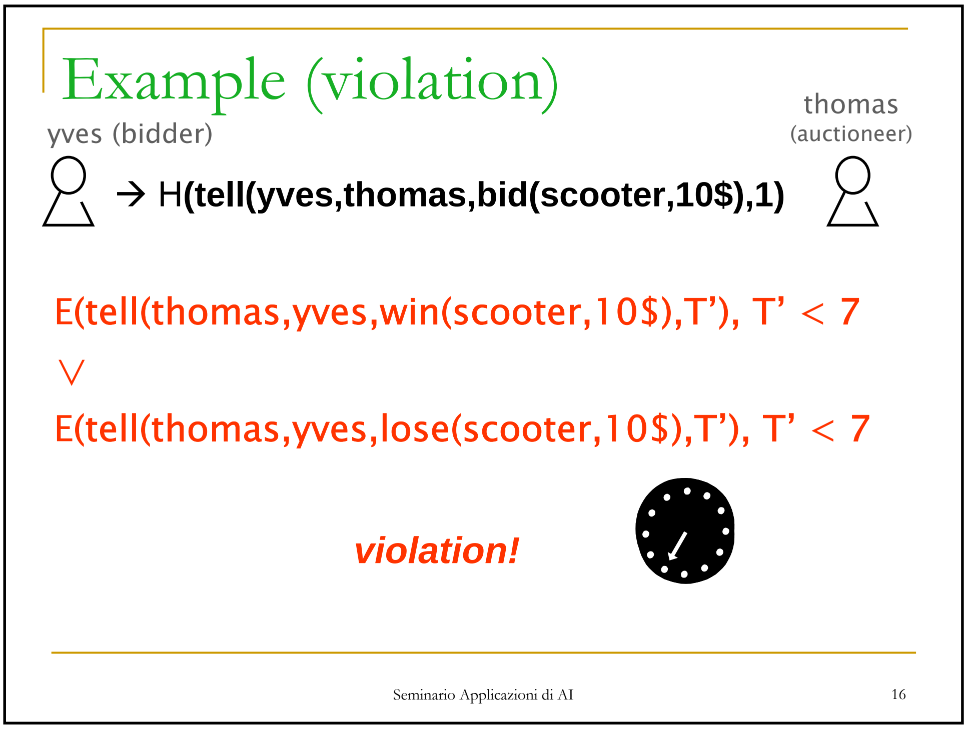
E(tell(thomas,yves,accept(scooter,10$),T'), T' < 7

∨

E(tell(thomas,yves,refuse(scooter,10$),T'), T' < 7

H(tell(thomas,yves,accept(scooter,10$),5) ←

fulfillment!



Example (violation)

yves (bidder)

thomas
(auctioneer)

→ H**(tell(yves,thomas,bid(scooter,10$),1)**

E(tell(thomas,yves,win(scooter,10$),T'), T' < 7

∨

E(tell(thomas,yves,lose(scooter,10$),T'), T' < 7

violation!

Example (violation)

yves

thomas

→ H(tell(yves,thomas,offer(scooter,10$),1)

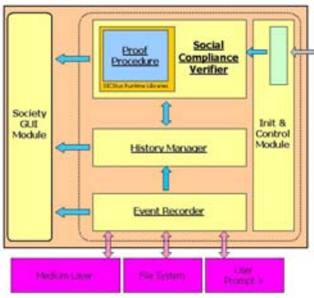
H(tell(thomas,yves,accept(scooter,10$),5) ←

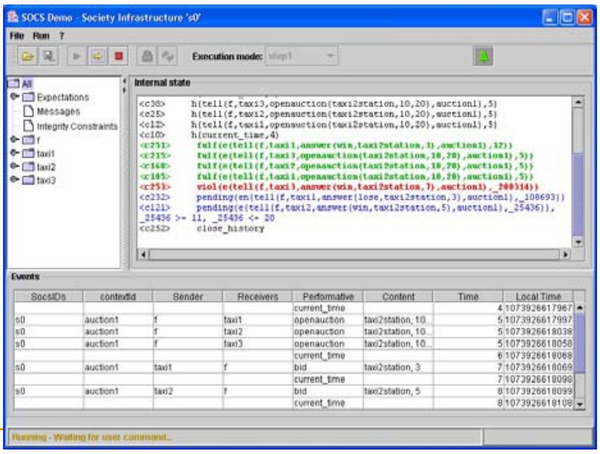
H(tell(thomas,yves,accept(Item,Price), T) ∨
EN(tell(thomas,yves,refuse(Item,Price), Tr), Tr>=T

H(tell(thomas,yves,refuse(scooter,10$),8) ←

violation!

First Prototype

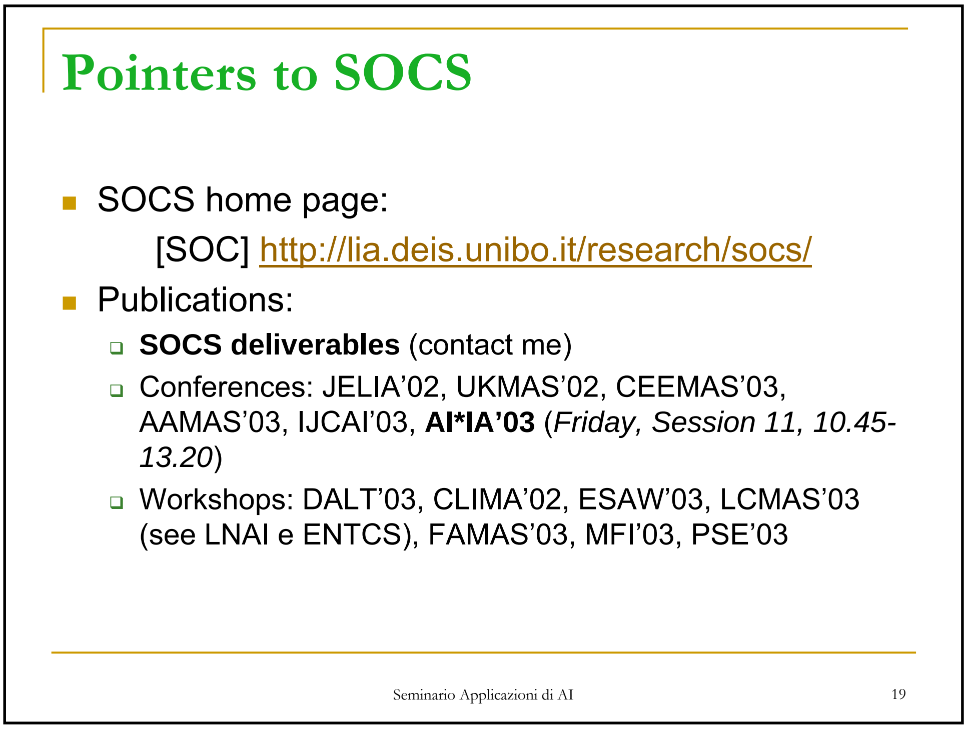




```
<c30>     h(tell(f,taxi3,openauction(taxi2station,10,20),auction1),5)
<c25>     h(tell(f,taxi2,openauction(taxi2station,10,20),auction1),5)
<c12>     h(tell(f,taxi1,openauction(taxi2station,10,20),auction1),5)
<c10>     h(current_time,4)
<c251>     fulf(e(tell(f,taxi1,answer(win,taxi2station,3),auction1),12))
<c215>     fulf(e(tell(f,taxi3,openauction(taxi2station,10,20),auction1),5))
<c160>     fulf(e(tell(f,taxi2,openauction(taxi2station,10,20),auction1),5))
<c105>     fulf(e(tell(f,taxi1,openauction(taxi2station,10,20),auction1),5))
<c253>     viol(e(tell(f,taxi3,answer(win,taxi2station,7),auction1),_200314))
<c232>     pending(en(tell(f,taxi1,answer(lose,taxi2station,3),auction1),_108693))
<c121>     pending(e(tell(f,taxi2,answer(win,taxi2station,5),auction1),_25436)),
_25436 >= 11, _25436 <= 20
<c252>     close_history
```

SocsIDs	contextId	Sender	Receivers	Performative	Content	Time	Local Time
				current_time		4	1073926617967
s0	auction1	f	taxi1	openauction	taxi2station, 10...	5	1073926617997
s0	auction1	f	taxi2	openauction	taxi2station, 10...	5	1073926618038
s0	auction1	f	taxi3	openauction	taxi2station, 10...	5	1073926618058
				current_time		6	1073926618068
s0	auction1	taxi1	f	bid	taxi2station, 3	7	1073926618069
				current_time		7	1073926618090
s0	auction1	taxi2	f	bid	taxi2station, 5	8	1073926618099
				current_time		8	1073926618108

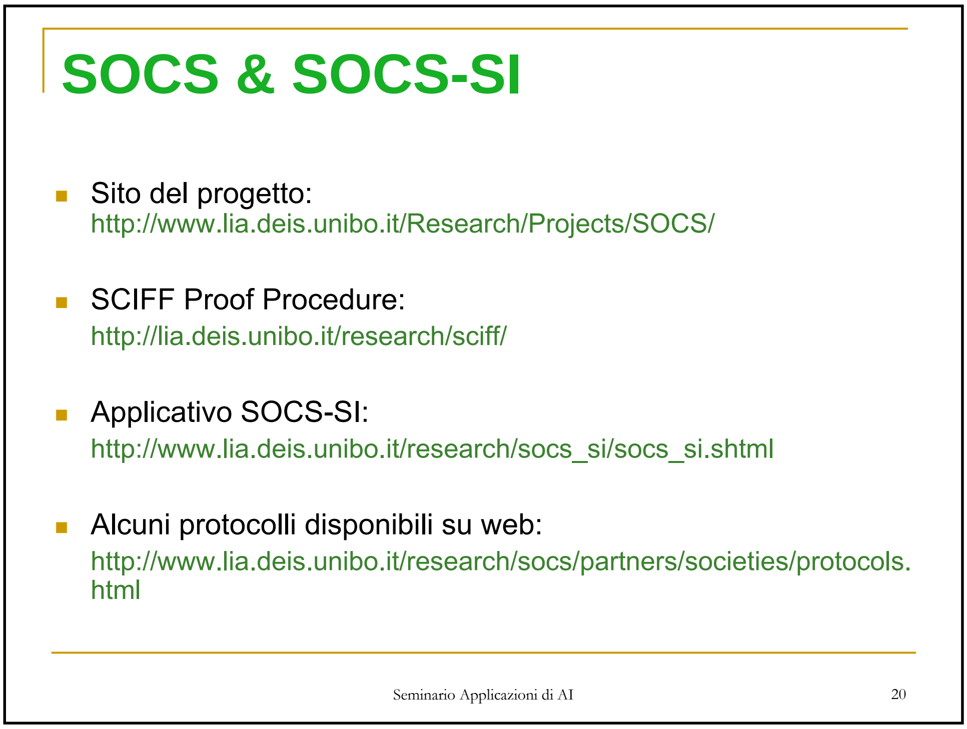
Running - Waiting for user command...



Pointers to SOCS

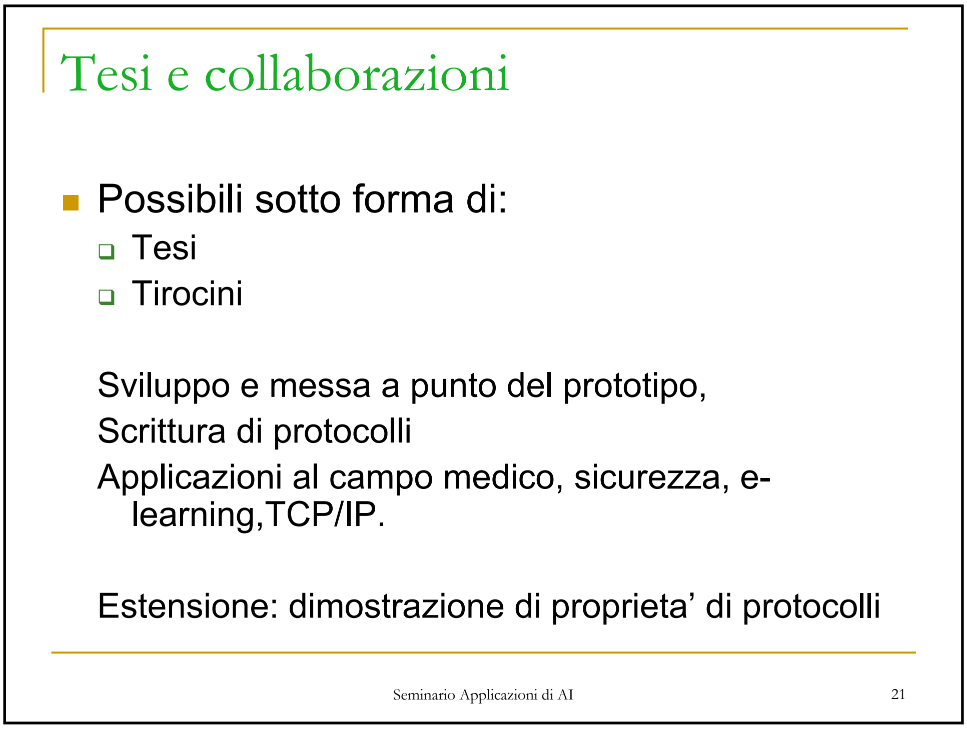
- SOCS home page:

  [SOC] http://lia.deis.unibo.it/research/socs/
- Publications:
  - **SOCS deliverables** (contact me)
  - Conferences: JELIA'02, UKMAS'02, CEEMAS'03, AAMAS'03, IJCAI'03, **AI*IA'03** (*Friday, Session 11, 10.45-13.20*)
  - Workshops: DALT'03, CLIMA'02, ESAW'03, LCMAS'03 (see LNAI e ENTCS), FAMAS'03, MFI'03, PSE'03



SOCS & SOCS-SI

- Sito del progetto:
  http://www.lia.deis.unibo.it/Research/Projects/SOCS/
- SCIFF Proof Procedure:
  http://lia.deis.unibo.it/research/sciff/
- Applicativo SOCS-SI:
  http://www.lia.deis.unibo.it/research/socs_si/socs_si.shtml
- Alcuni protocolli disponibili su web:
  http://www.lia.deis.unibo.it/research/socs/partners/societies/protocols.html



Tesi e collaborazioni

- Possibili sotto forma di:
  - Tesi
  - Tirocini

Sviluppo e messa a punto del prototipo,

Scrittura di protocolli

Applicazioni al campo medico, sicurezza, e-learning,TCP/IP.

Estensione: dimostrazione di proprieta' di protocolli