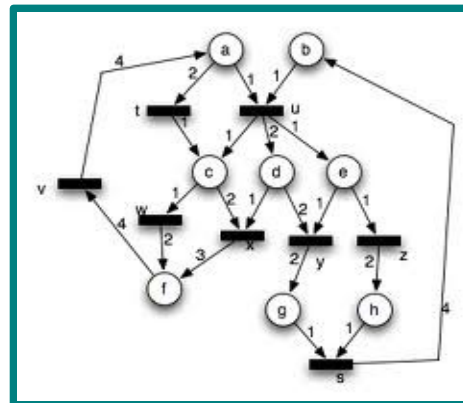
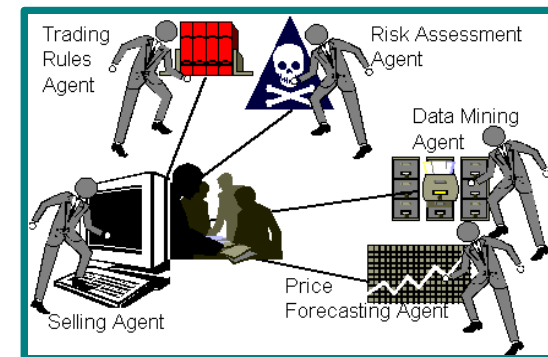
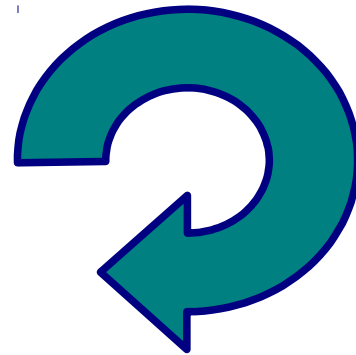
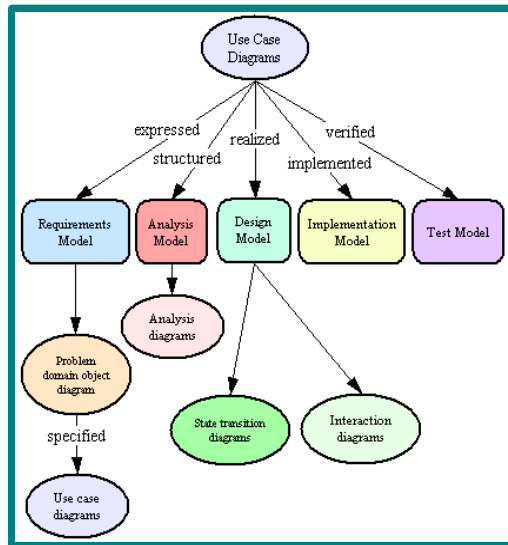
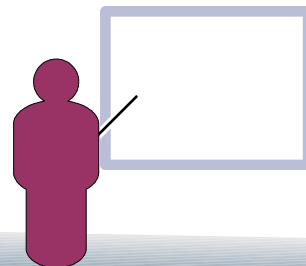


Modeling Agent Interactions: Agent UML and Multiagent Interaction Protocol Nets

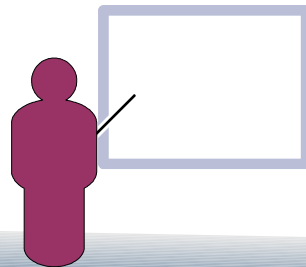


Marco Alberti
Corso di Sistemi Multi-Agente L-S del prof. Andrea Omicini
Laurea Specialistica in Ingegneria Informatica
II Facoltà di Ingegneria, Cesena



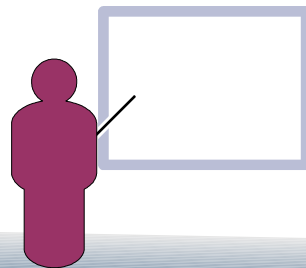
Overview

- *Oggetti ed agenti*
 - Classi di *oggetti*
 - Classi di *agenti*
 - Reattività e (pro)attività
 - La comunicazione
- L'interazione in AUMML
 - Diagrammi di sequenza
 - Parallelismo
 - Le *decision box*
 - Notazione semplificata
 - Esempio: asta all'inglese
 - Carenze del modello
- MIP-Nets
 - Reti di Petri
 - A-Nets
 - IP-Nets
 - Da AUMML alle IP-Nets
 - Creare una MIP-Net
 - Il *dispiegamento*
 - A-Nets *minimali*
- Bibliografia



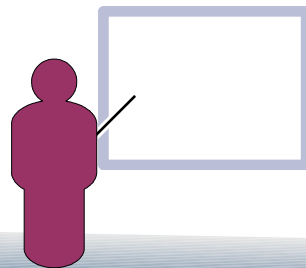
Classi di oggetti

- Com'è fatto un *oggetto*?
 - Ha uno stato interno (variabili o altri *oggetti*)
 - Ha dei metodi (visibili o meno dall'esterno)
 - Comunica invocando dei metodi (chiamate a procedura)



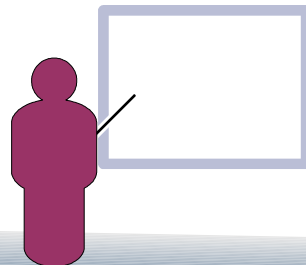
Classi di *agenti*

- Com'è fatto un *agente*?
 - Ha uno stato interno (variabili o *oggetti* ma anche convinzioni, obiettivi, piani...)
 - Può avere dei metodi (non visibili dall'esterno)
 - Esegue delle azioni (queste note all'esterno)
 - Comunica tramite *atti comunicativi* (pattern di tipo *message passing*)



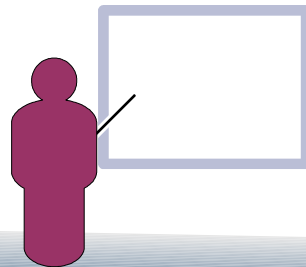
Reattività e (pro)attività

- Sia gli *oggetti* che gli *agenti* sono reattivi
 - realizzano degli output in risposta a determinati input secondo il proprio stato interno
- Gli agenti sono anche pro-attivi
 - non è necessario che ricevano degli input per decidere di realizzare un output
 - le decisioni sono determinate dallo stato interno (*agent head automaton*)



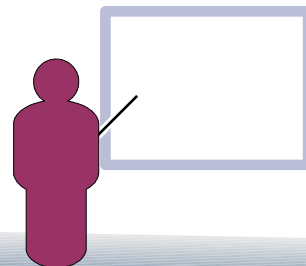
La comunicazione

- Gli *oggetti* comunicano tramite invocazione di metodi
 - il controllo viene passato ad un altro *oggetto* che lo restituisce una volta terminata la chiamata
- Gli *agenti* comunicano tramite scambio di messaggi
 - i messaggi vengono ricevuti e inoltrati tramite un interfaccia di comunicazione supportato dall'infrastruttura
 - non c'è passaggio di controllo

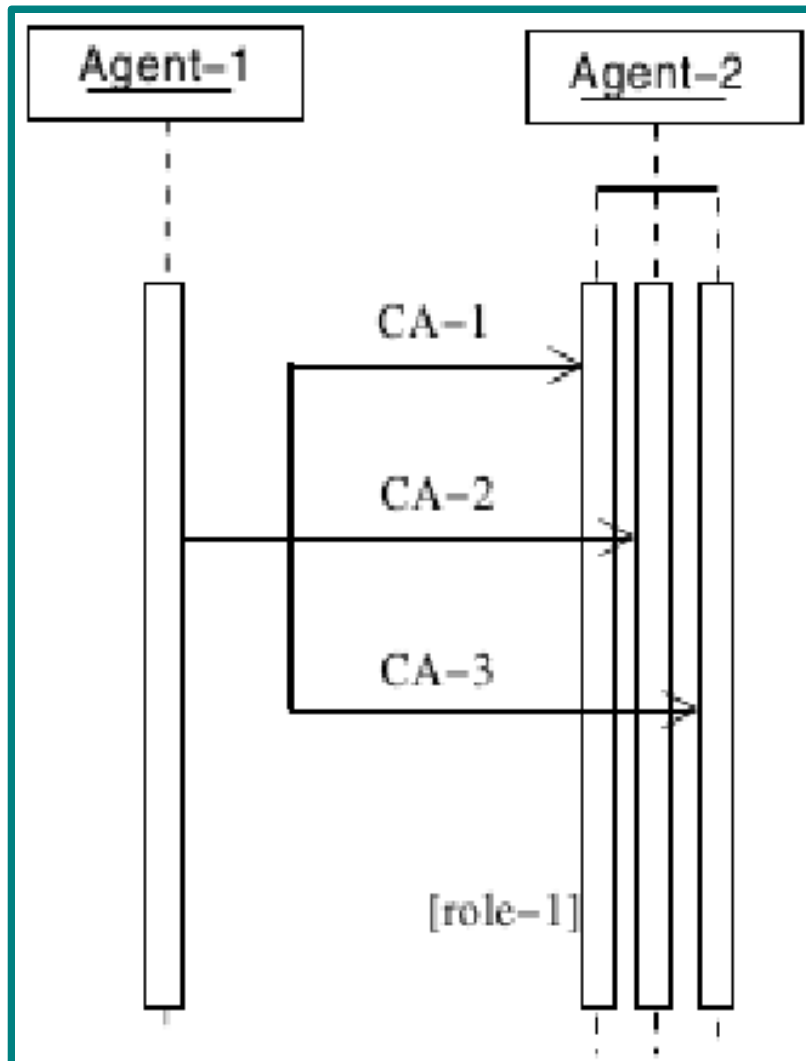


Diagrammi di sequenza

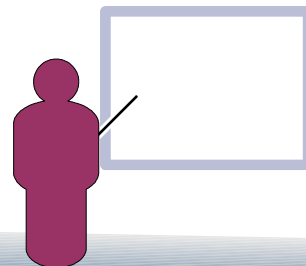
- I diagrammi di sequenza esprimono le interazioni tra due o più *agenti*
- I thread dell'*agente* vengono identificati dalle linee verticali dette *lifelines*
- Le barre verticali indicano processi esecutivi eseguiti in parallelo
- Le comunicazioni (asincrone) sono indicate dalle frecce
- Il *ruolo* dell'*agente* è indicato tra le parentesi quadre accanto ad ogni *lifeline*
- Le scelte (esclusive e non) compiute dall'*agente* sono indicate dai *rombi* (barrati con una X o meno)



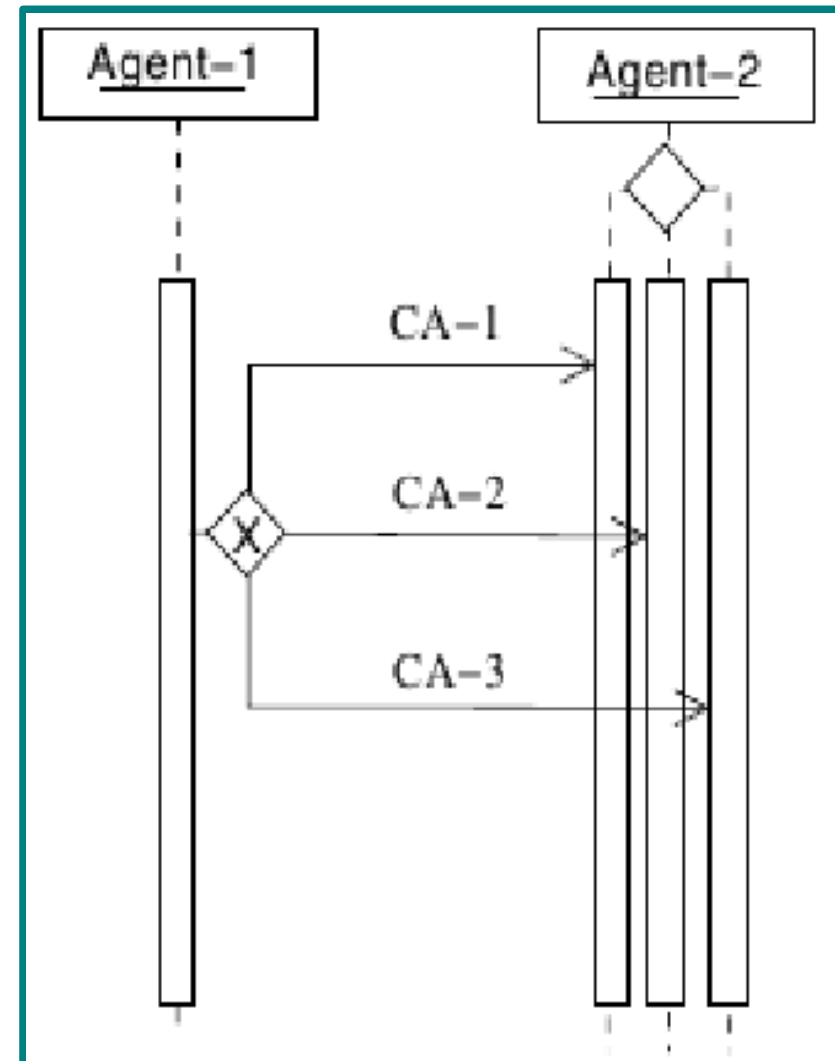
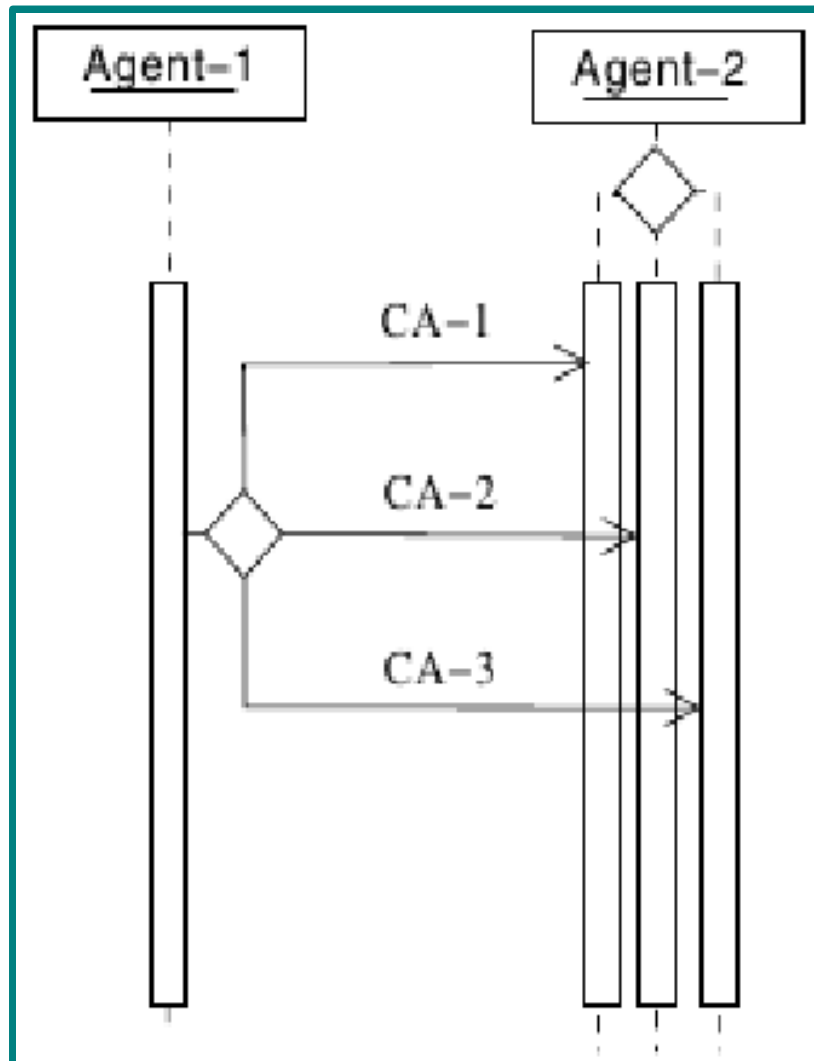
Parallelismo



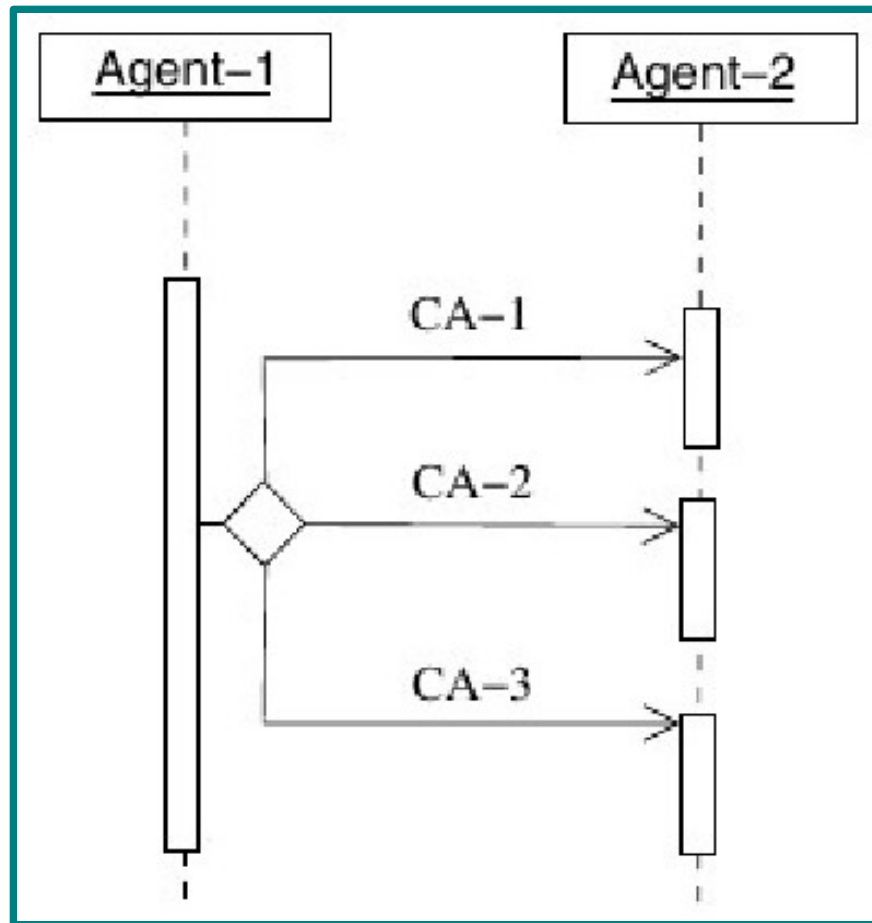
- Più *lifelines* riconducibili ad un medesimo *agente* significano che tale *agente* è in grado di ricoprire altrettanti *ruoli* contemporaneamente.
- I processi di esecuzione, indicati dalle barre verticali, avvengono indipendentemente e in parallelo.



Le *decision box*

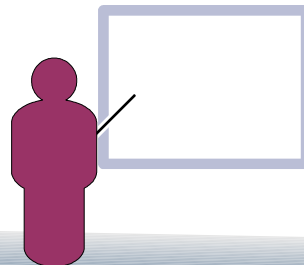


Notazione semplificata

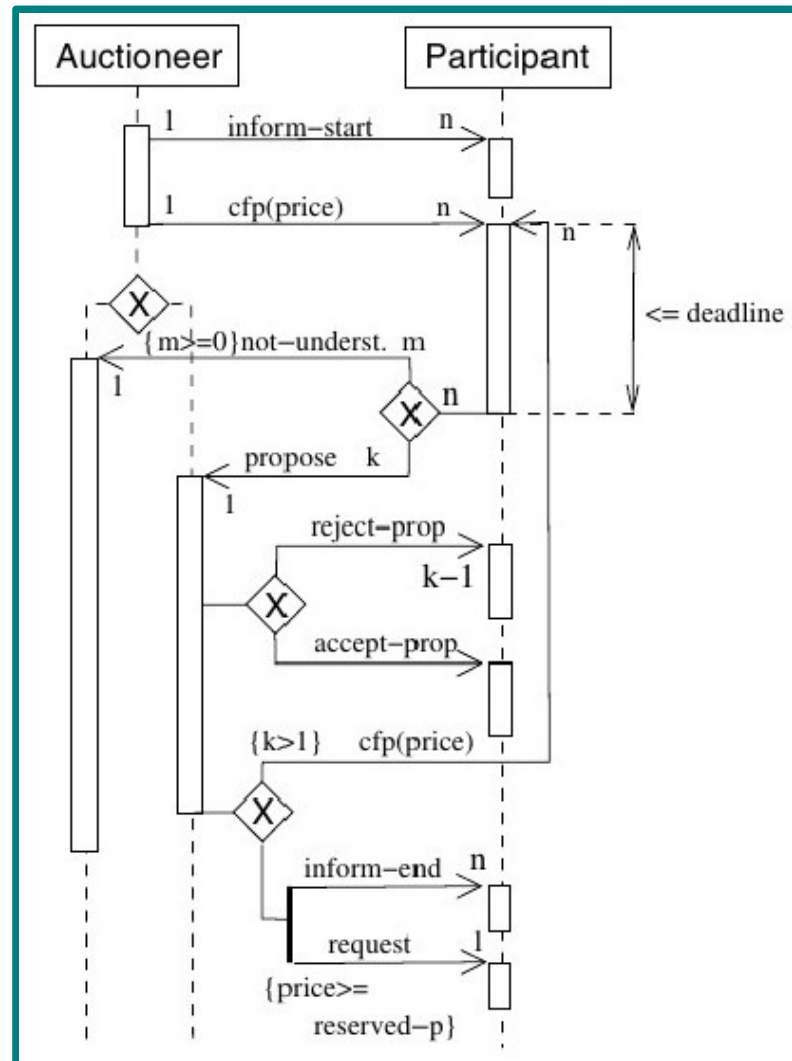


Attenzione, il modello è fuorviante!

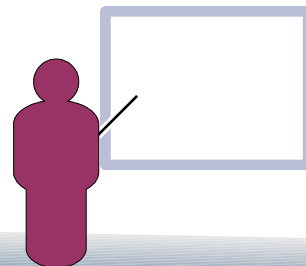
- Le *lifelines* vengono collassate ma il processamento dei messaggi rimane parallelo.
- Più *ruoli* vengono associati alla *lifeline*.



Esempio: asta all'inglese

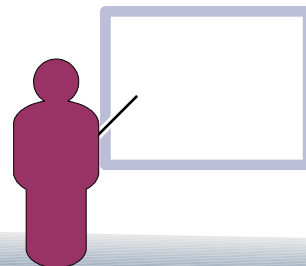


- **n** – numero di partecipanti
- **k** – partecipanti che accettano l'offerta
- **m** – partecipanti che non rispondono
- **m + k = n**

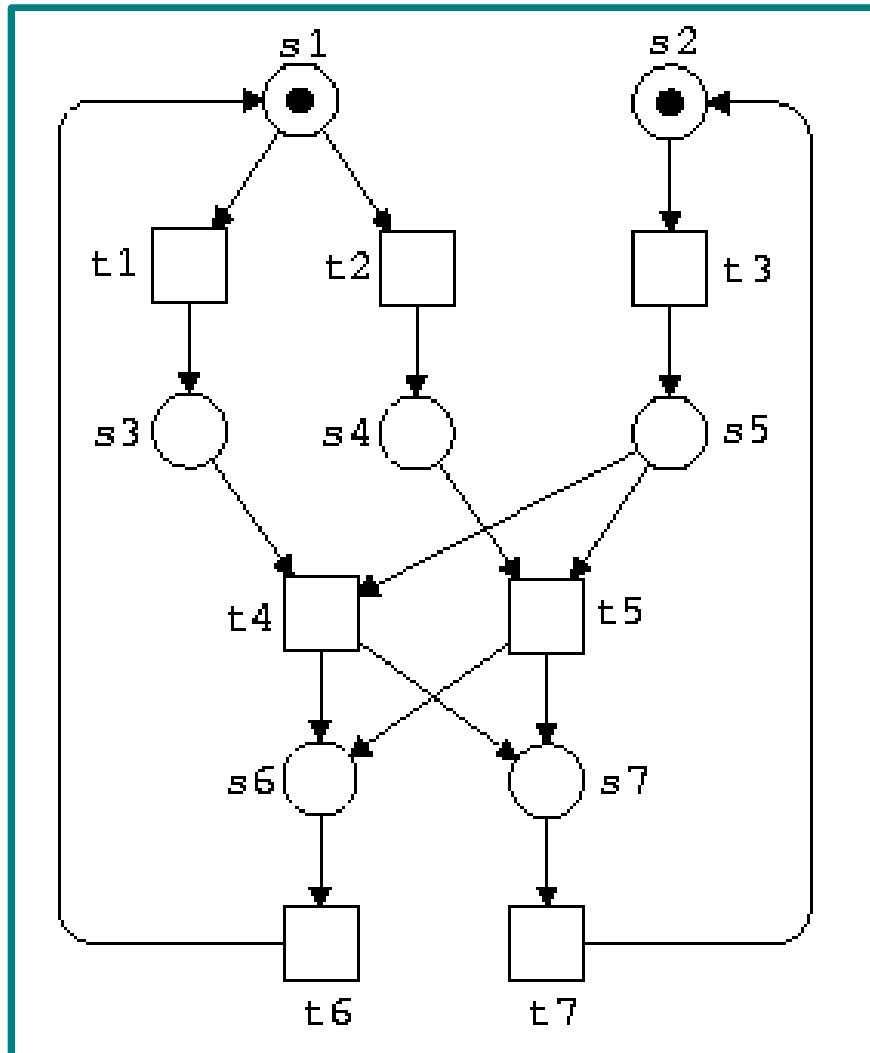


Carenze del modello

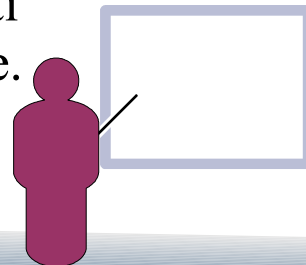
- Il diagramma rappresenta un possibile scenario o tutti quelli possibili?
- Nella prima decision box c'è una molteplicità **n**. Significa che **n** agenti devono, o possono, rispondere in una delle due maniere?
- I messaggi possono andare perduti?
- Come inviare un messaggio ad un preciso agente nel set?
- Come garantire che alla fine il banditore deciderà effettivamente di chiudere l'asta?
- Come esprimere un comportamento vietato?



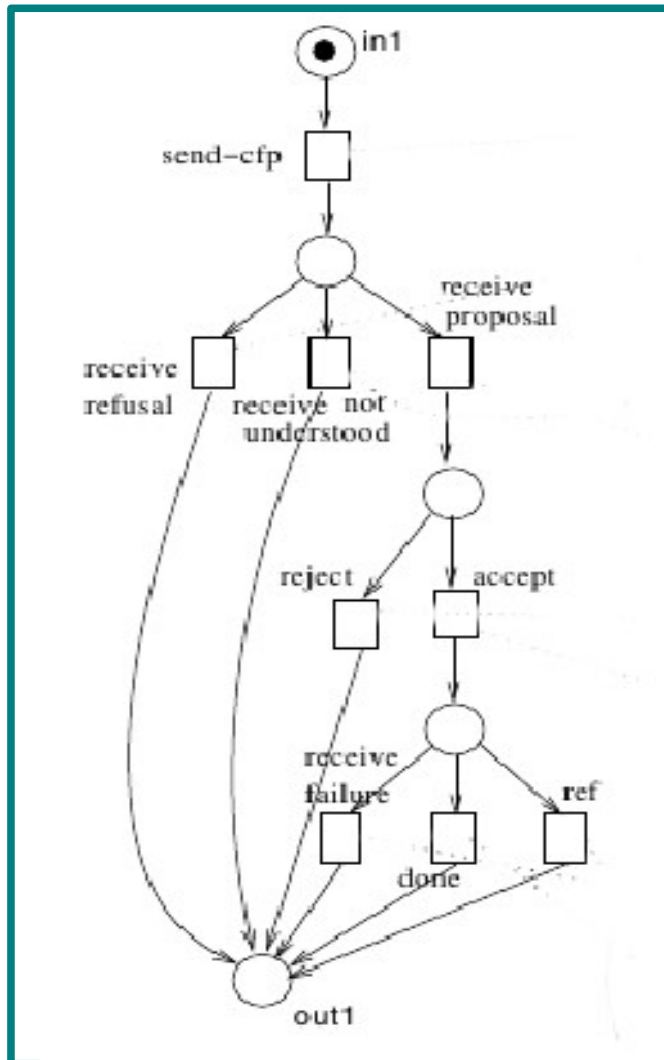
Reti di Petri



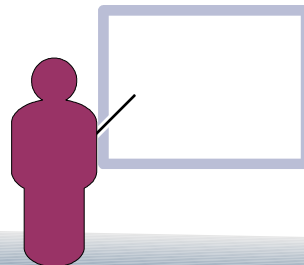
- Le reti di Petri descrivono la dinamica dei sistemi:
CERCHI – stati del processo
RETTANGOLI – transizioni
FRECCE – archi di collegamento
PALLINI – flussi di controllo
- Una transizione è abilitata se e solo se è presente un pallino in ognuno degli stati che hanno archi in entrata alla transizione.
- Scattando, la transizione rimuove un pallino da ciascuno di questi stati e ne genera un altro in ognuno degli stati che hanno archi entranti provenienti dalla transizione.



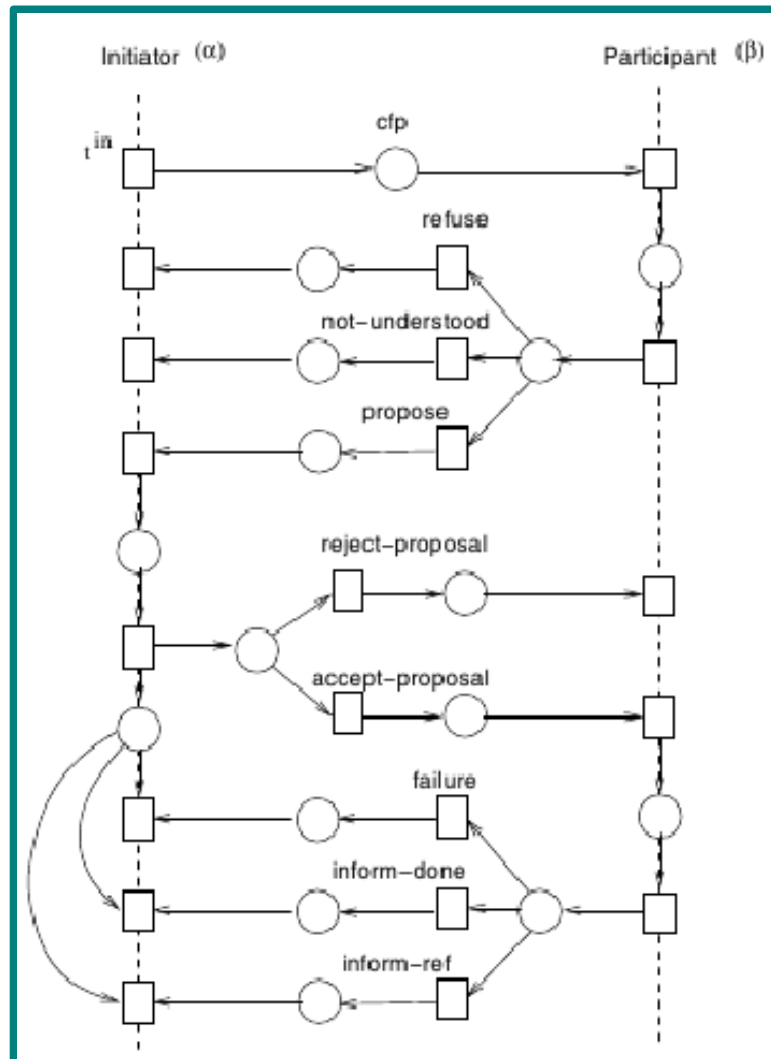
A-Nets



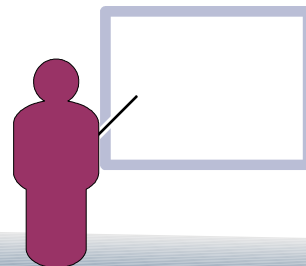
- Una rete di Petri che descrive il comportamento di un *agente* è detta A-Net.
- Deve avere uno stato iniziale e uno finale, rispettivamente non preceduto e non seguito, da alcun altro componente.
- Ogni componente deve contribuire alla realizzazione del comportamento dell'*agente*; non vi devono cioè essere rami pendenti.



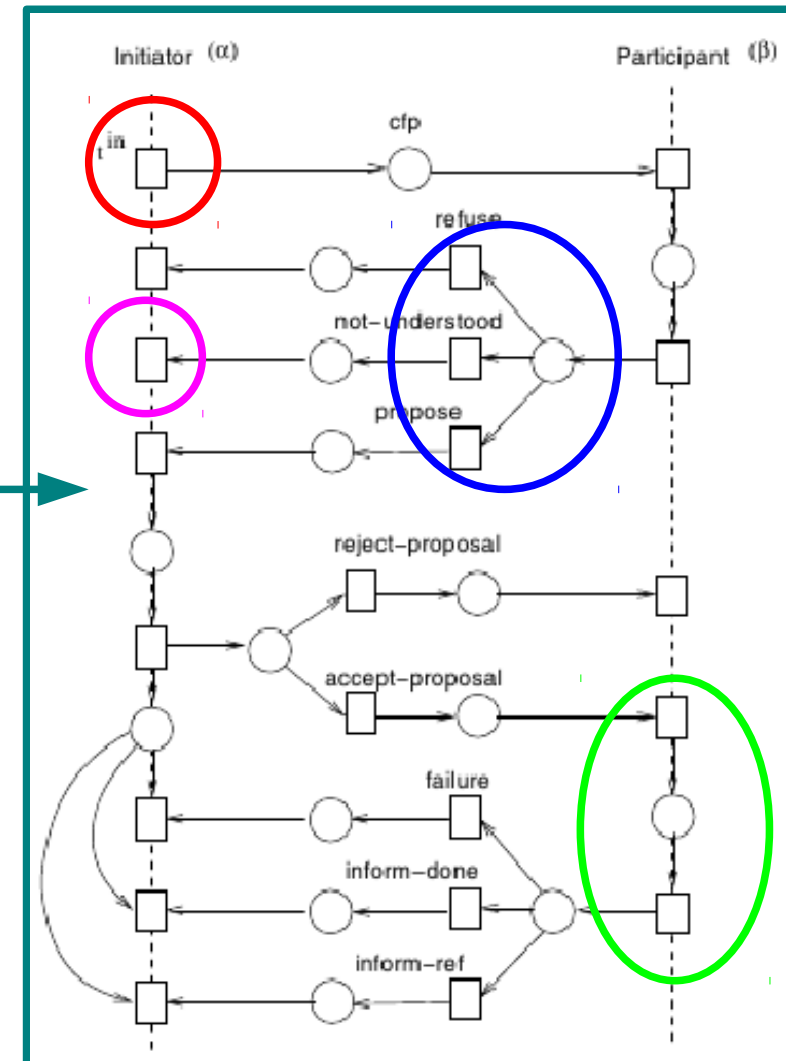
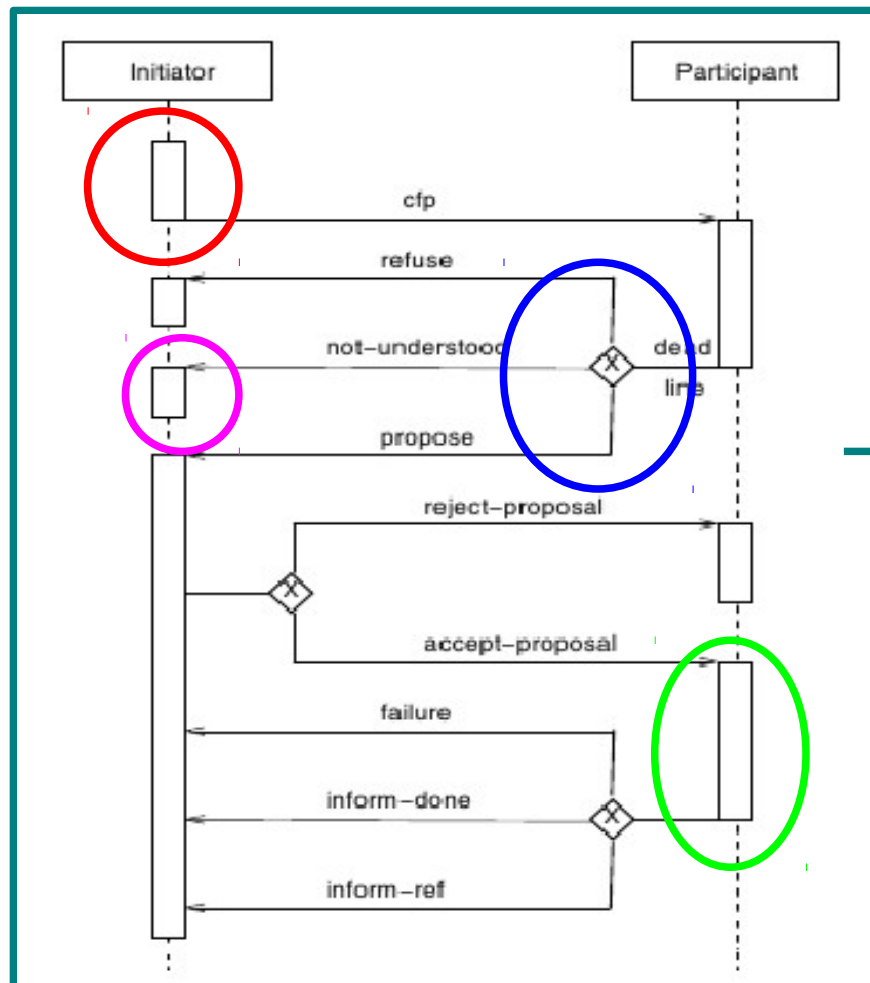
IP-Nets



- Una rete di Petri che descrive un protocollo di interazione è detta IP-Net.
- Deve avere una transizione iniziale non preceduta da alcun altro componente e un set di transizioni finali non seguite da alcun altro componente.
- Deve essere possibile distinguere due set disgiunti di transizioni riconducibili agli agenti coinvolti.

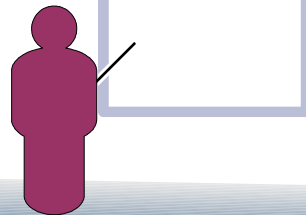
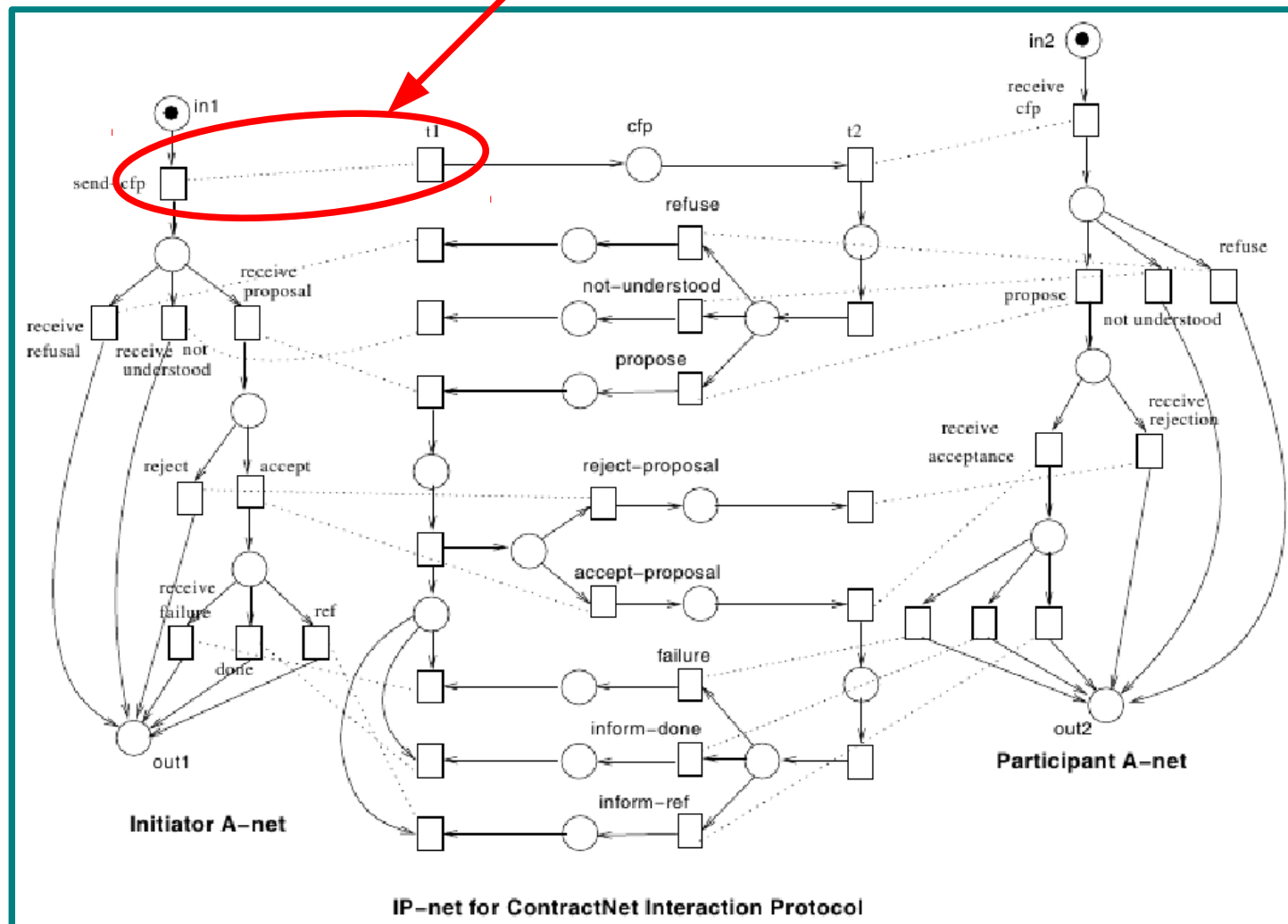


Da AUML alle IP-Nets



Creare una MIP-Net

sincronizzazione

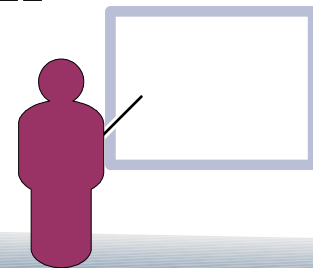


Il *dispiegamento*

Si definisce *dispiegamento* di una MIP-Net, la rete di Petri (A-Net) risultante dalle seguenti operazioni:

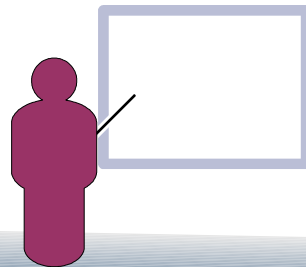
- le coppie di transizioni formate da quelle delle A-Nets e quelle dei rispettivi set delle IP-Nets vengono “sincronizzate”
- viene aggiunto uno stato, seguito da una transizione, che si comporterà da nuovo stato iniziale dell'intero sistema
- viene aggiunto uno stato, preceduta da una transizione, che si comporterà da nuovo stato finale dell'intero sistema

Se la rete risultante è *sound* allora lo è anche il sistema realizzato dalla MIP-Net.



A-Nets *minimali*

- Anche se le A-Nets e le IP-Nets che compongono una MIP-Net sono *sound*, non è detto che lo sia anche la MIP-Net. Occorre verificare che lo sia il suo *dispiegamento*.
- Osservando però una IP-Net (che sia *sound*) è possibile derivare delle A-Net (anch'esse *sound*) che garantiscono la *soundness* anche della MIP-Net.
- Tali A-Net si dicono *minimali*, perchè utilizzano il minimo numero di transizioni per realizzare il protocollo della IP-Net.



Bibliografia

- **GIVING LIFE TO AGENT INTERACTIONS**

Juliana Küster Filipe

Laboratory for Foundations of Computer Science, School of Informatics

The University of Edinburgh, King's Buildings

Edinburgh EH9 3JZ, United Kingdom

jkfilipe@inf.ed.ac.uk

- **MIP-NETS: A COMPOSITIONAL MODEL OF MULTIAGENT INTERACTION**

Sea Ling and Seng Wai Loke

School of Computer Science and Software Engineering, Monash University

P O Box 197, VIC 3145, Australia

sling@csse.monash.edu.au, swlok@csse.monash.edu.au

- **UML CLASS DIAGRAMS REVISITED IN THE CONTEXT OF AGENT-BASED SYSTEMS**

Bernhard Bauer

Siemens AG, Corporate Technology, Information and Communications

Otto-Hahn-Ring 6

81739 Munich, Germany

bernhard.bauer@mchp.siemens.de

