

METODI FORMALI DI MODELLAZIONE

Alessandro Bianchi, Sebastiano Pizzutilo, Gennaro Vessio

Laboratorio Menabrea

Dipartimento Informatica

Università di Bari

{alessandro.bianchi, sebastiano.pizzutilo, gennaro.vessio}@uniba.it

I Paradigmi dell'Informatica (1/2)

- L'attività di ricerca nell'ambito dell'informatica si svolge secondo tre paradigmi principali
 - Deduttivo / Logico-Matematico
 - Induttivo / Sperimentale
 - Realizzativo / Progettuale

I Paradigmi dell'Informatica (2/2)

- Il nostro approccio
 - parte dal paradigma deduttivo
 - le soluzioni proposte sono validate sperimentalmente
 - Talvolta realizziamo soluzioni applicative

Sistemi Critici e Complessi

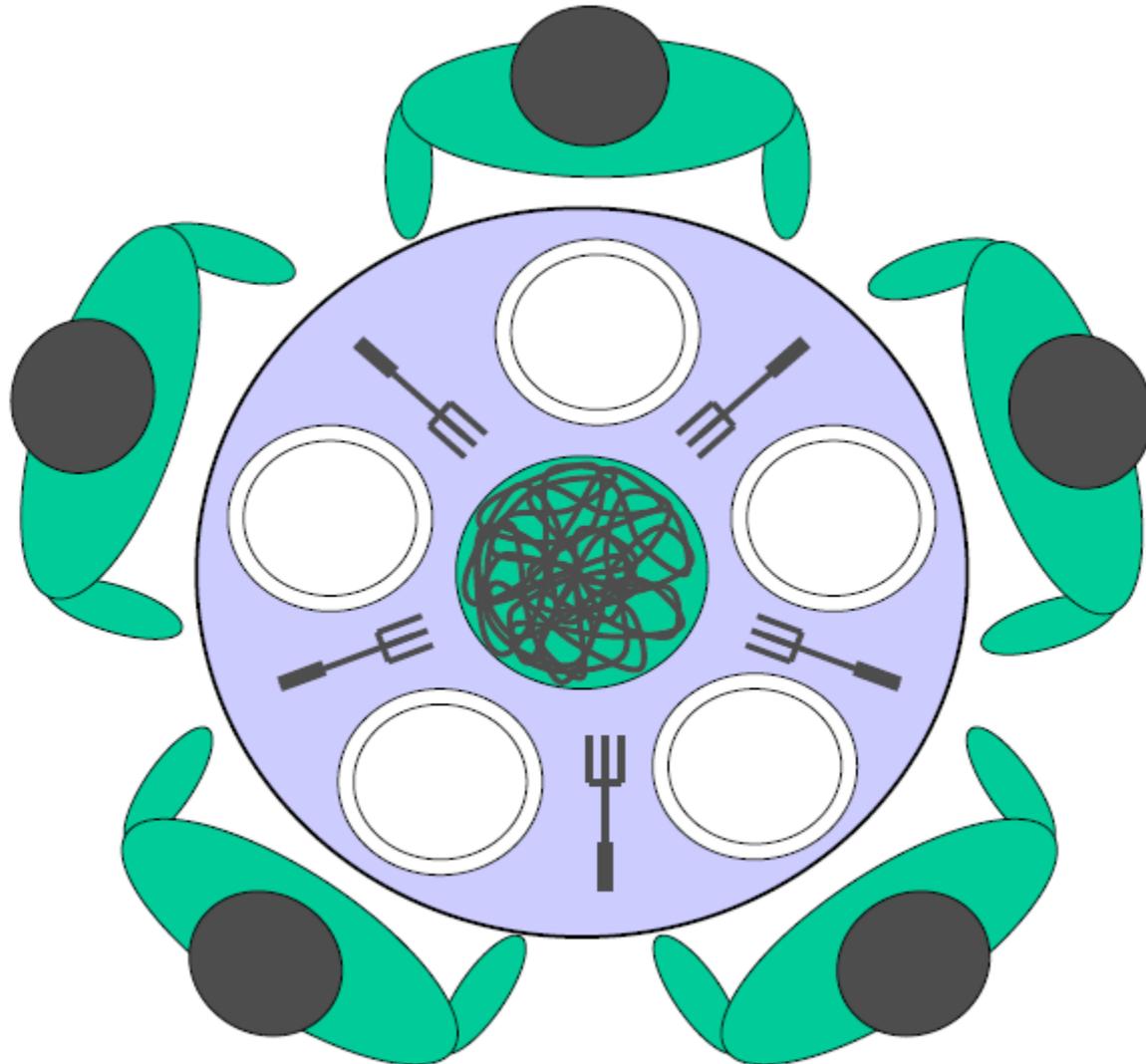
- L'informatica ha molti modi per definire / misurare la **complessità** di un sistema
- Non ha modo di stabilire la sua **criticità**
 - È di pertinenza del dominio applicativo
 - Informalmente: un sistema i cui malfunzionamenti possono provocare gravi danni

Problema

- L'analisi di sistemi complessi
 - è lunga, difficile, costosa
 - non sempre efficace
- Non ha senso svolgerla per sistemi di bassa criticità

Esempio di Proprietà

5 Filosofi a Cena

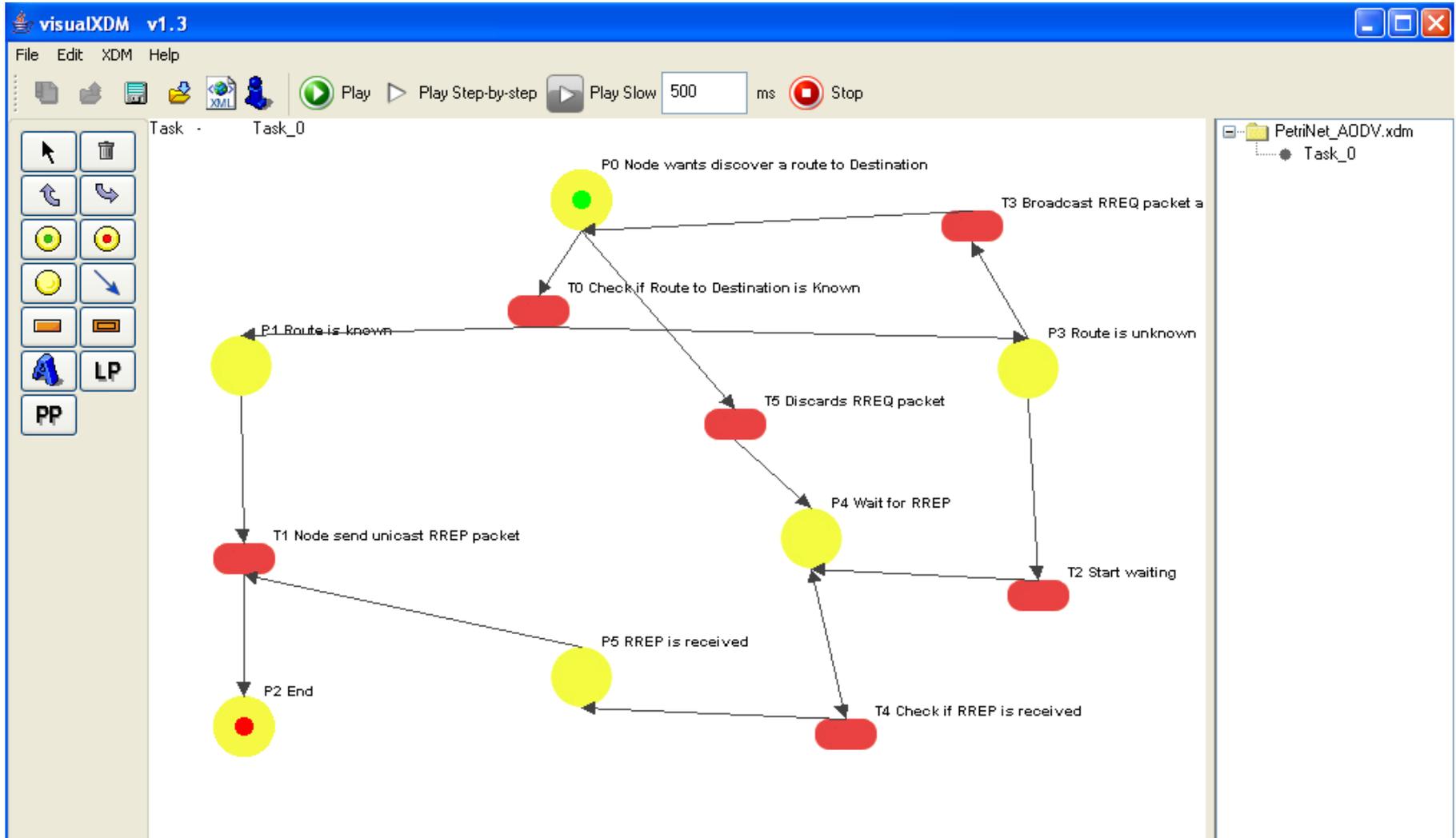


Il Nostro Approccio

- Specificare **formalmente** il sistema attraverso un opportuno **modello**
 - Prima di realizzarlo
- Analizzare le proprietà di interesse sul modello
- Se soddisfatte, procedere alla realizzazione, applicando fedelmente quanto descritto nel modello
- Se non soddisfatte?

ESEMPIO 1: Petri net

- $PN = \langle N, P, T, A, SP, EP, M \rangle$



ESEMPIO 2: Abstract State Machine

- ASM=insieme di regole if cond then update

AgentProgram(a) =

```
    if  $\neg$ (isEmpty(requests(self))) then {
        RREQ = top(requests(self))
        nextHop = sender of top(requests(self))
        update routingTable(self)
        receivedRREQ(self, dest) := true
        Router(RREQ, nextHop)
    }
    if wishToInitiate(self, dest) = true then
        Initiator(dest)
    if  $\neg$ (isEmpty(replies(self))) {
        RREP = top(replies(self))
        nextHop = select c.nextHop □
        hostInRT(routingTable(self)) with
        RREP.init = c.dest
        update routingTable(self)
        UnicastRREP(RREP, nextHop)
        dequeue RREQ from replies(self)
    }
```

UnicastRREP(RREP, a) =

enqueue RREP into replies(a)

ESEMPIO 3: ASM per i 5 Filosofi

```
PhilosopherProgram(pi) = {  
    if owner(rightFork(self)) = undef  $\wedge$   
    owner(leftFork(self)) = undef then {  
        owner(rightFork(self)) := self  
        owner(leftFork(self)) := self  
    }  
    if owner(rightFork(self)) = self  $\wedge$   
    owner(leftFork(self)) = self then {  
        owner(rightFork(self)) := undef  
        owner(leftFork(self)) := undef  
    }  
}
```