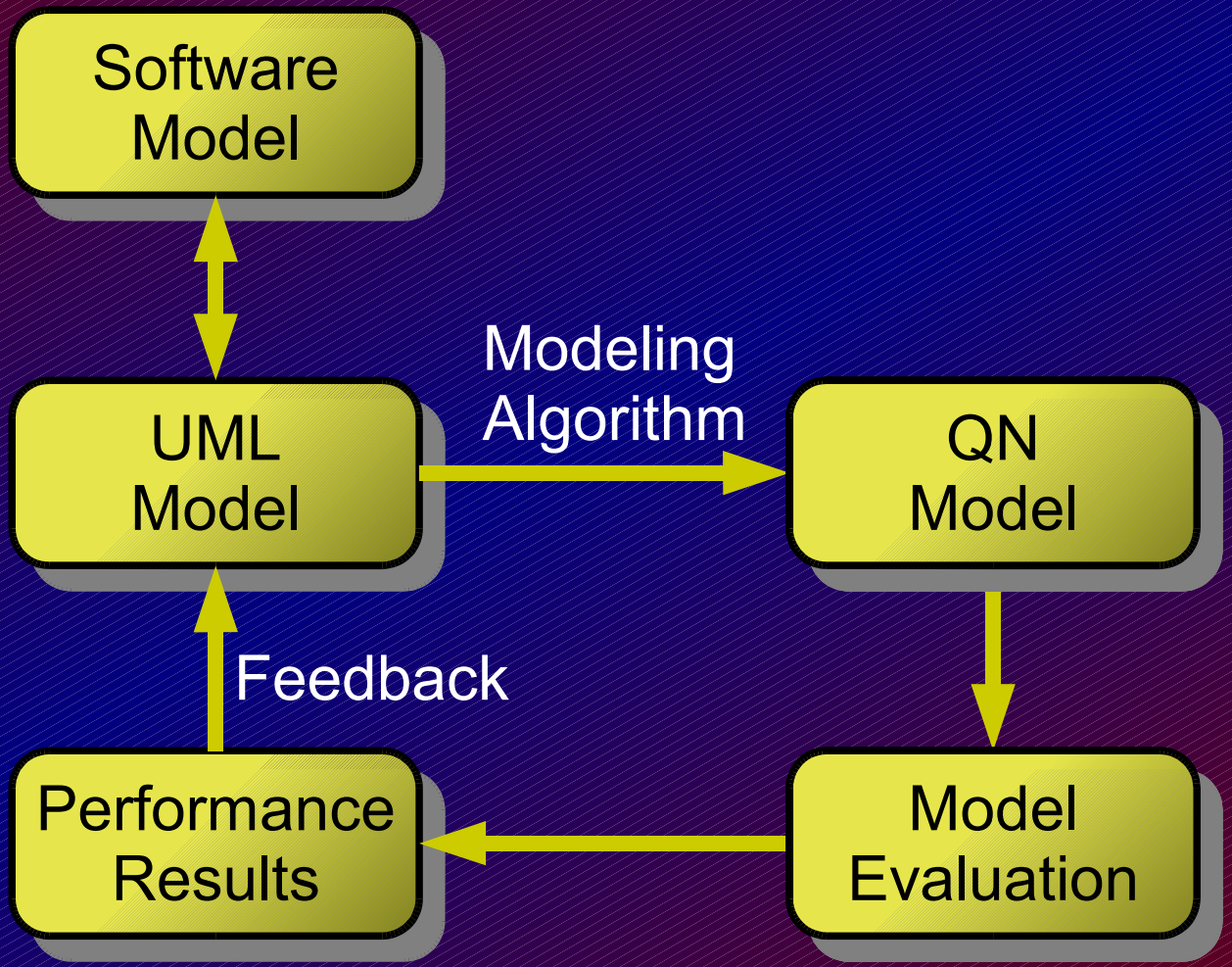
Valutazione delle prestazioni di Architetture Software con specifica UML mediante QN multiclasse

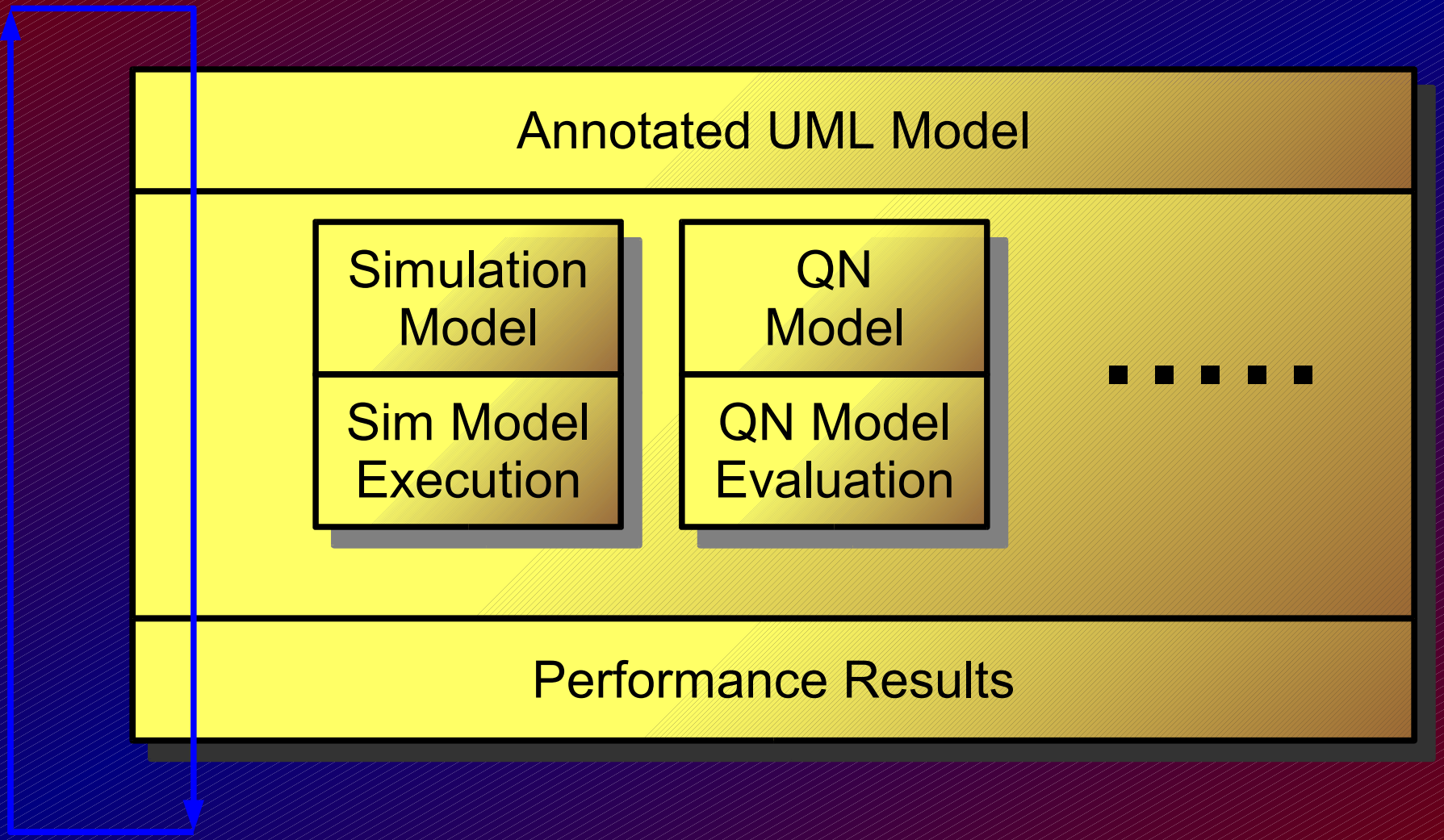
Moreno Marzolla
Dipartimento di Informatica
Università Ca' Foscari di Venezia

Riunione FIRB/PERF—1/2 dicembre 2004

Software Performance Modeling



Performance Modeling Framework



Performance Modeling with multiclass QN

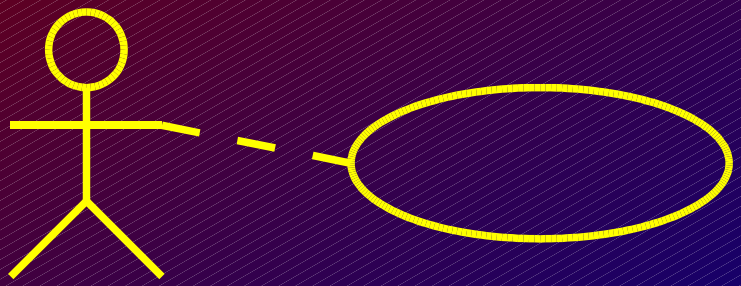
- Starting point
  - Use Case diagrams (workloads)
  - Deployment diagrams (hardware model)
  - Activity diagrams (system execution model)
- Target notation
  - Mixed Multiclass QN model
  - Use Case diagrams ⇨ Workloads
  - Deployment diagrams ⇨ Service centers
  - Activity diagrams ⇨ Topology

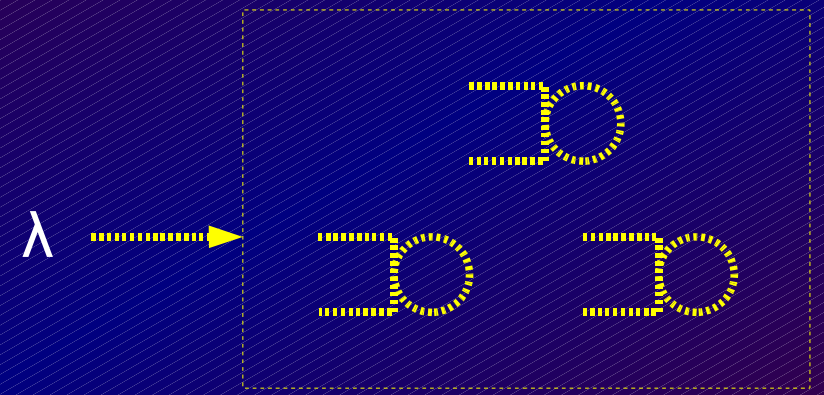
Why?

- Performance model generation can be done efficiently
  - O(*#transitions* + *#action states*)
- Performance model can be solved efficiently
  - If some constraints are satisfied
- The approach uses standard UML SPT profile annotations
  - Can be integrated with existing software performance modeling frameworks based on the profile

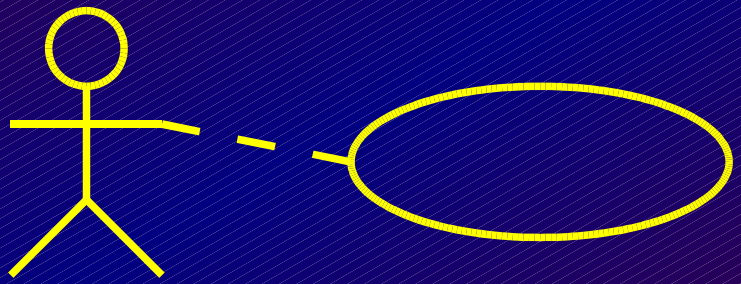
Translating UC diagrams

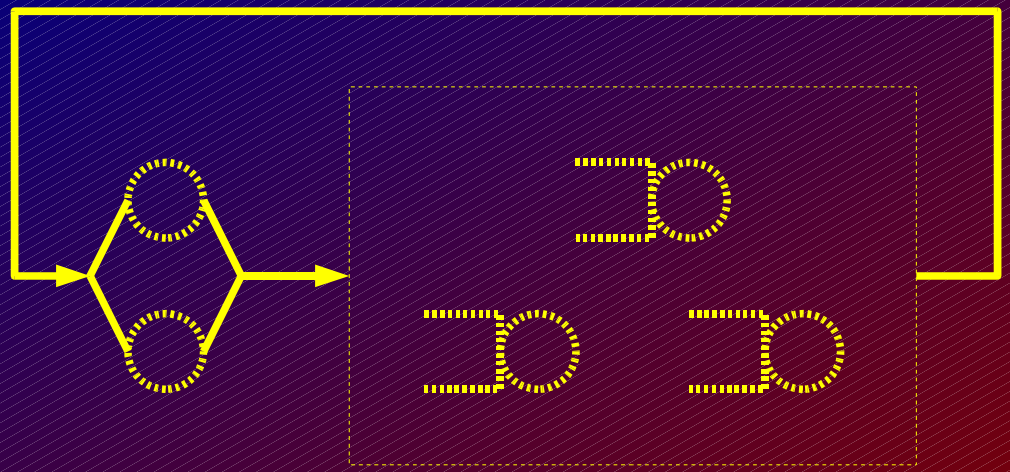
<<PAopenLoad>>



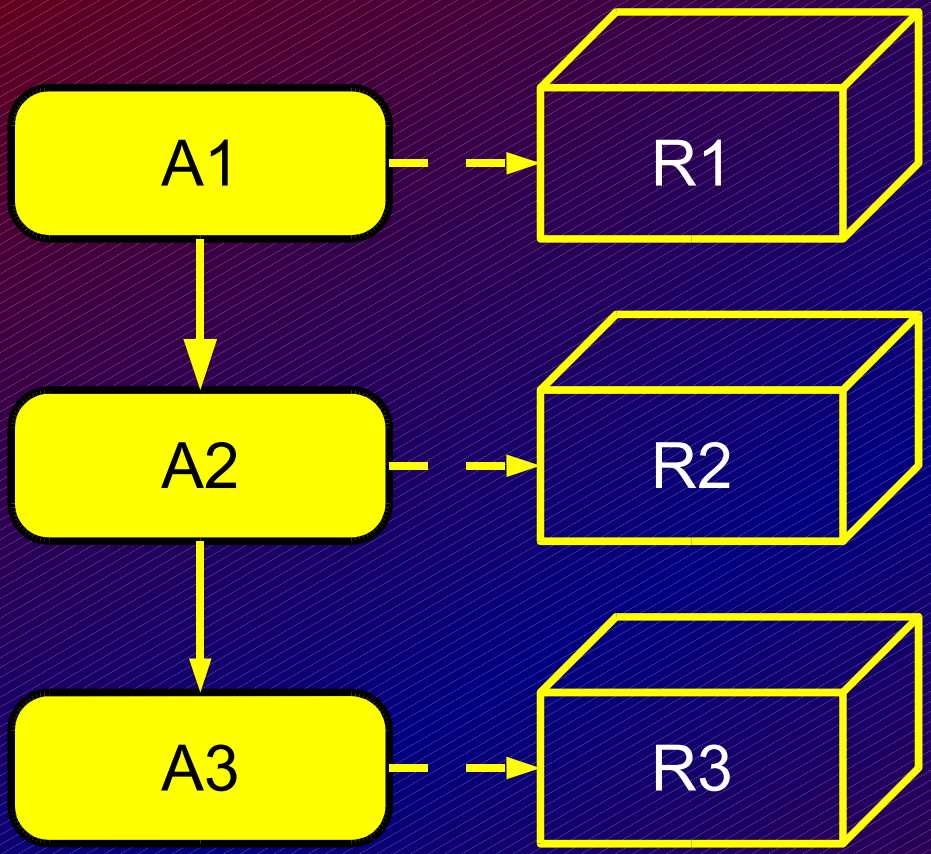


<<PAclosedLoad>>

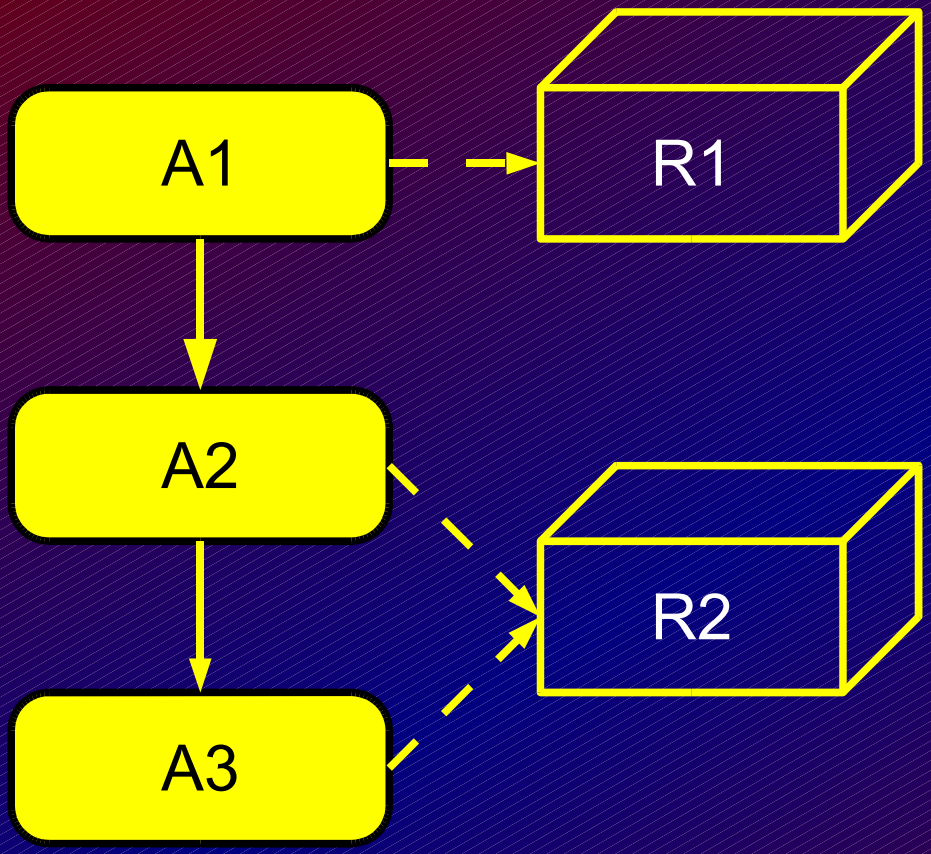


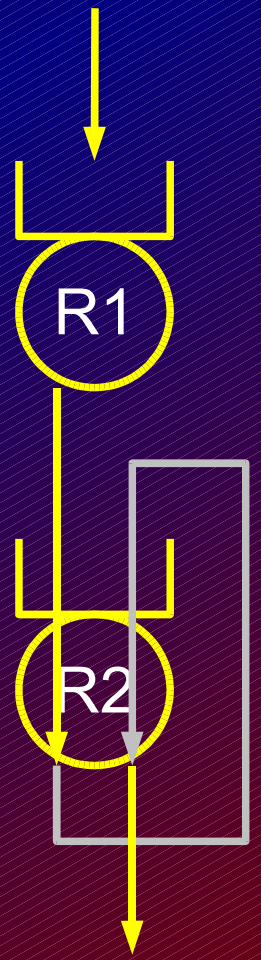


Translating Activity diagrams the easy case



Translating Activity diagrams the difficult case





Outline of the transformation algorithm

- Translate one Activity diagram at a time
  - Each Activity diagram corresponds to a single chain
- Resources correspond to service centers
- Translate an Activity diagram as follows
  - All actions requesting service from the same resource receive a unique label in the range [1..k]
  - If there is a transition with probability *p* from an action with label *r* requesting service from resource *i* to an action with label *s* requesting service from resource *j*
    - Set *P[i,r,j,s] = p*

Example