

# Asynchronous ASMs

## Master/Slave Agreement

Egon Börger

Dipartimento di Informatica, Università di Pisa

<http://www.di.unipi.it/~boerger>

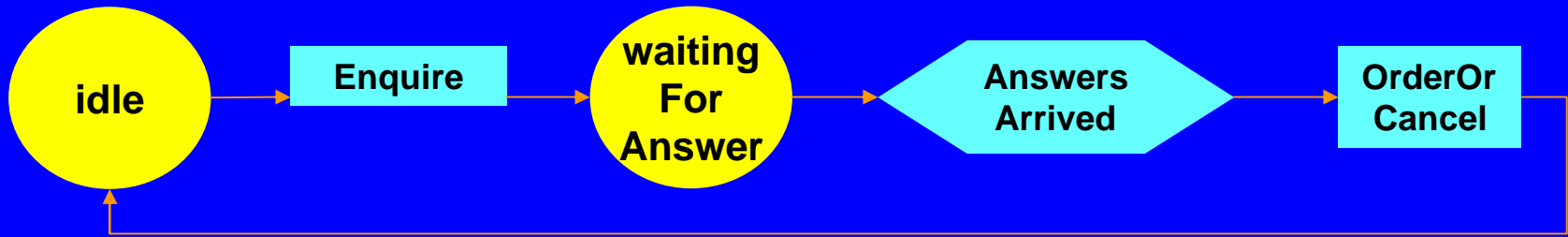
# Master/Slave Agreement : problem statement

- Goal: Design a distributed algorithm for a master launching orders to slave agents, to be confirmed by them and to be executed iff all of them confirm to the master the acceptance of the order.
- Algorithmic Idea:
  - the master enquires about a job to be launched and then waits for answers from the slaves
  - the slaves answer the enquiry and then wait for the order or the cancellation of the launched job from the master
  - the master orders or cancels the job depending on whether all slaves answered to accept it or not
- Eventually the master becomes idle, with all slaves either idle too or executing the accepted job.

# Master/Slave Agreement : Agent Signature

- **master**: a distinguished agent
  - **order**: Order external function yielding jobs to be sent to the slaves
  - **ctl\_state**: {idle, waitingForAnswer}
- **Slaves**: a set of agents equipped with
  - **asked**: {true,false} recording whether a job request has been launched by the master to the slave
  - **answer** : {accept,refuse,undef} recording whether a job request has been accepted by the slave
  - **ctl\_state**:{idle, waitingForOrder, busy}
- **Initially** ctl\_state = idle, order = answer =undef, asked = false
- NB. Abstraction from message passing: functions asked, answer, order shared in writing resp. reading among slaves and master

# Master/Slave Agreement ASMs



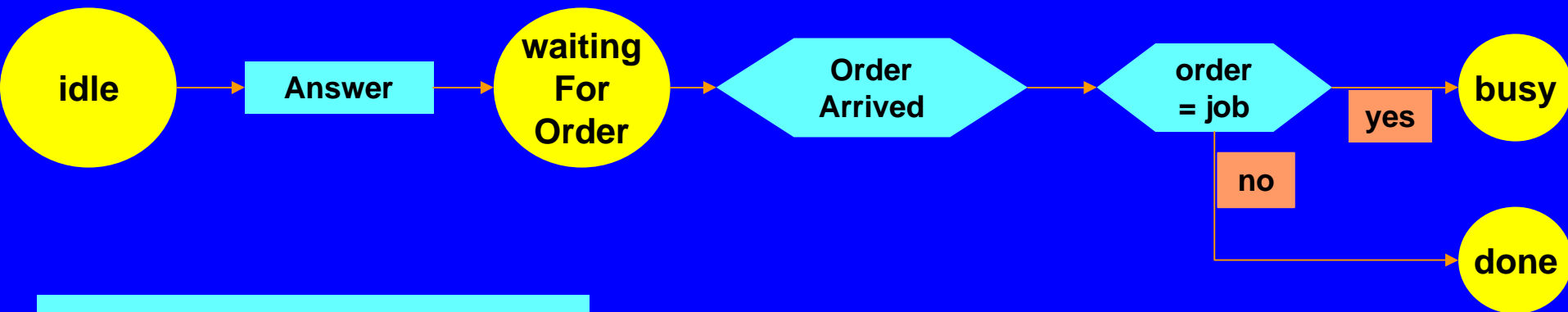
**Enquire** ° **forall**  $s \in \text{Slave}$   $s.\text{asked} := \text{true}$

**Answers Arrived** ° **forall**  $s \in \text{Slave}$   
 $s.\text{answer} \in \{\text{accept}, \text{refuse}\}$

**OrderOrCancel** °

clear answer

if for some  $s \in \text{Slave}$   $s.\text{answer} = \text{refuse}$   
 then  $\text{order} := \text{cancel}$   
 else  $\text{order} := \text{job}$



**Answer** ° if asked then  
 choose  $r \in \{\text{accept}, \text{refuse}\}$   
 $\text{answer} := r$   
 $\text{asked} := \text{false}$

**OrderArrived** °  $\text{order} \in \{\text{job}, \text{cancel}\}$

# Master/Slave Agreement Correctness

- Proposition: In every run of a set of master and slaves, all equipped with the corresponding master/slave ASM, after the master has started an Enquiry, eventually the master becomes idle and
  - either all slaves become done or
  - all slaves become busy executing the job ordered by the master
- Proof. Follows by run induction.

# Reference

- W.Reisig: Elements of Distributed Algorithms  
Springer-Verlag 1998
  - See the definition of a Petri net in Section 30 (Fig.30.1)  
and a detailed correctness proof in Section 75
- E. Börger, R. Stärk: Abstract State Machines. A  
Method for High-Level System Design and Analysis  
Springer-Verlag 2003, see  
<http://www.di.unipi.it/AsmBook>
  - See Chapter 6.1