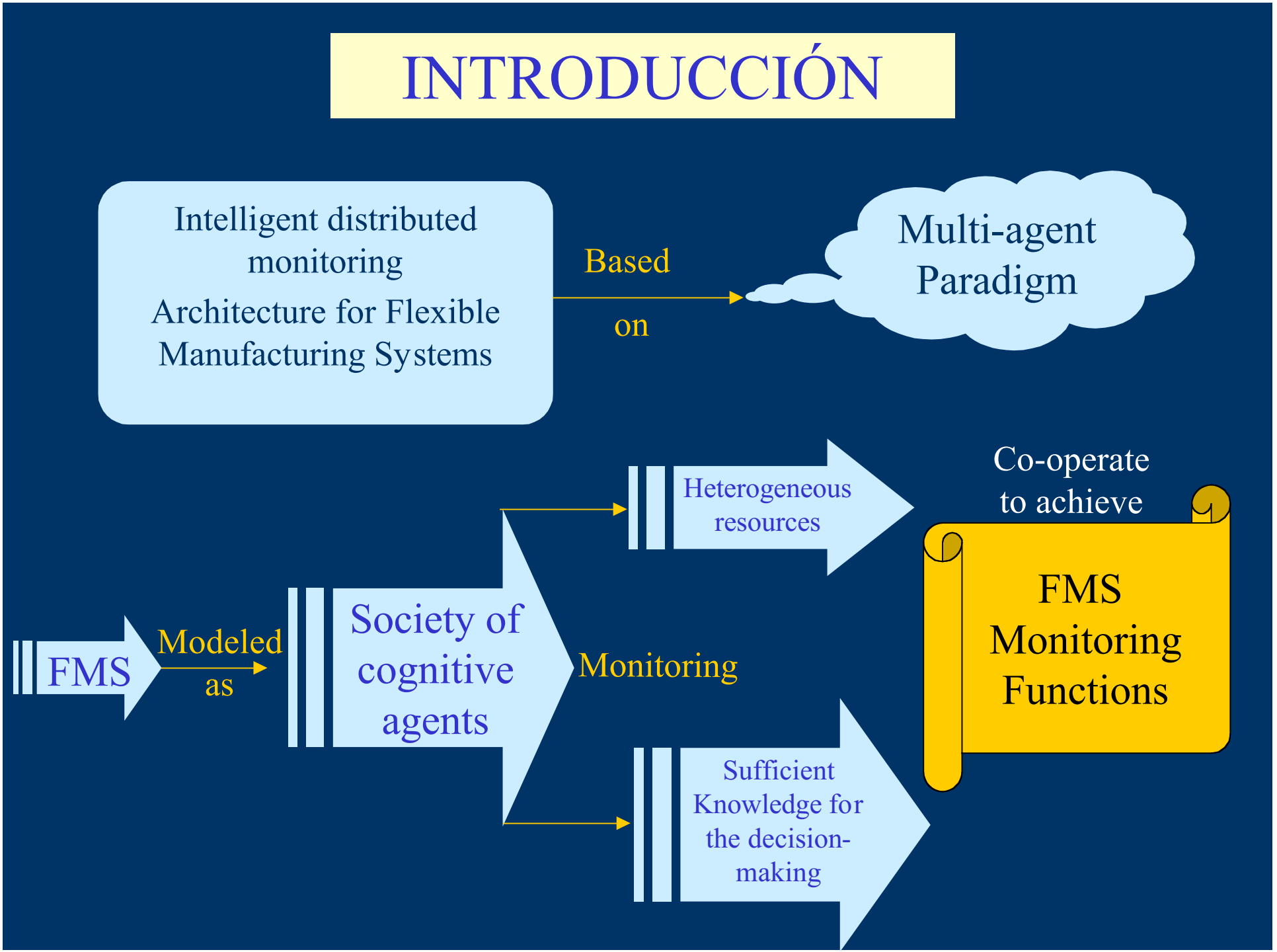
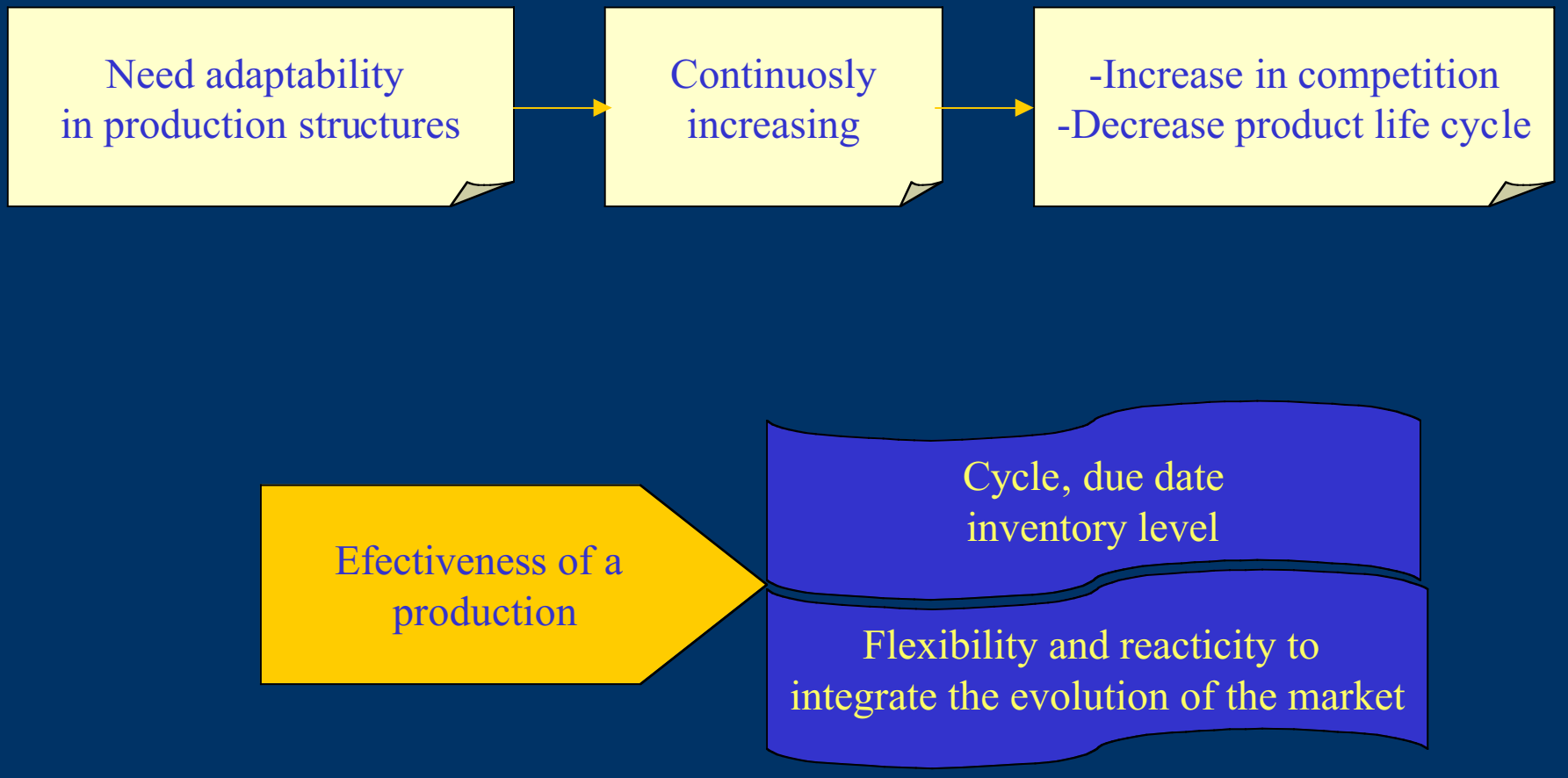
INTRODUCCIÓN



INTRODUCCIÓN



INTRODUCCIÓN

Multi Agents Systems (MAS)

- More Suitable for dynamic monitoring
- Build Complex and flexible advanced intelligent systems

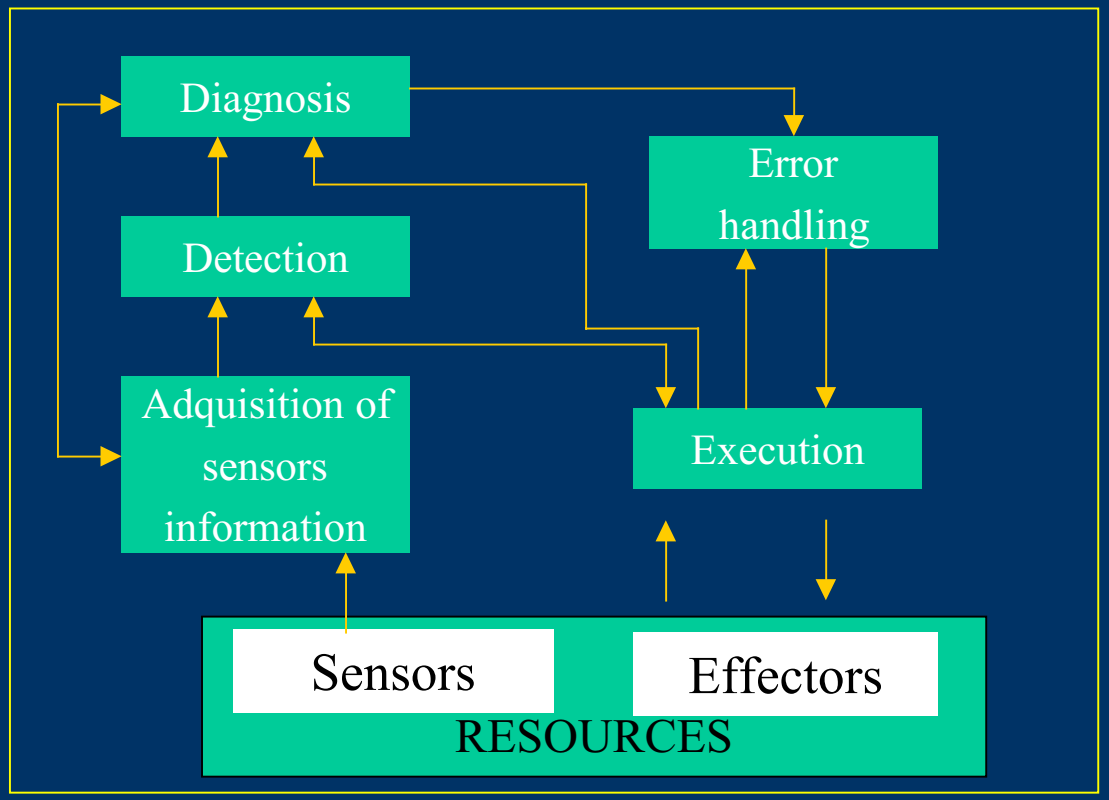
Monitoring Systems

More	Than
- decentralized	- centralized
- emergent	- planned
- concurrent	- sequential

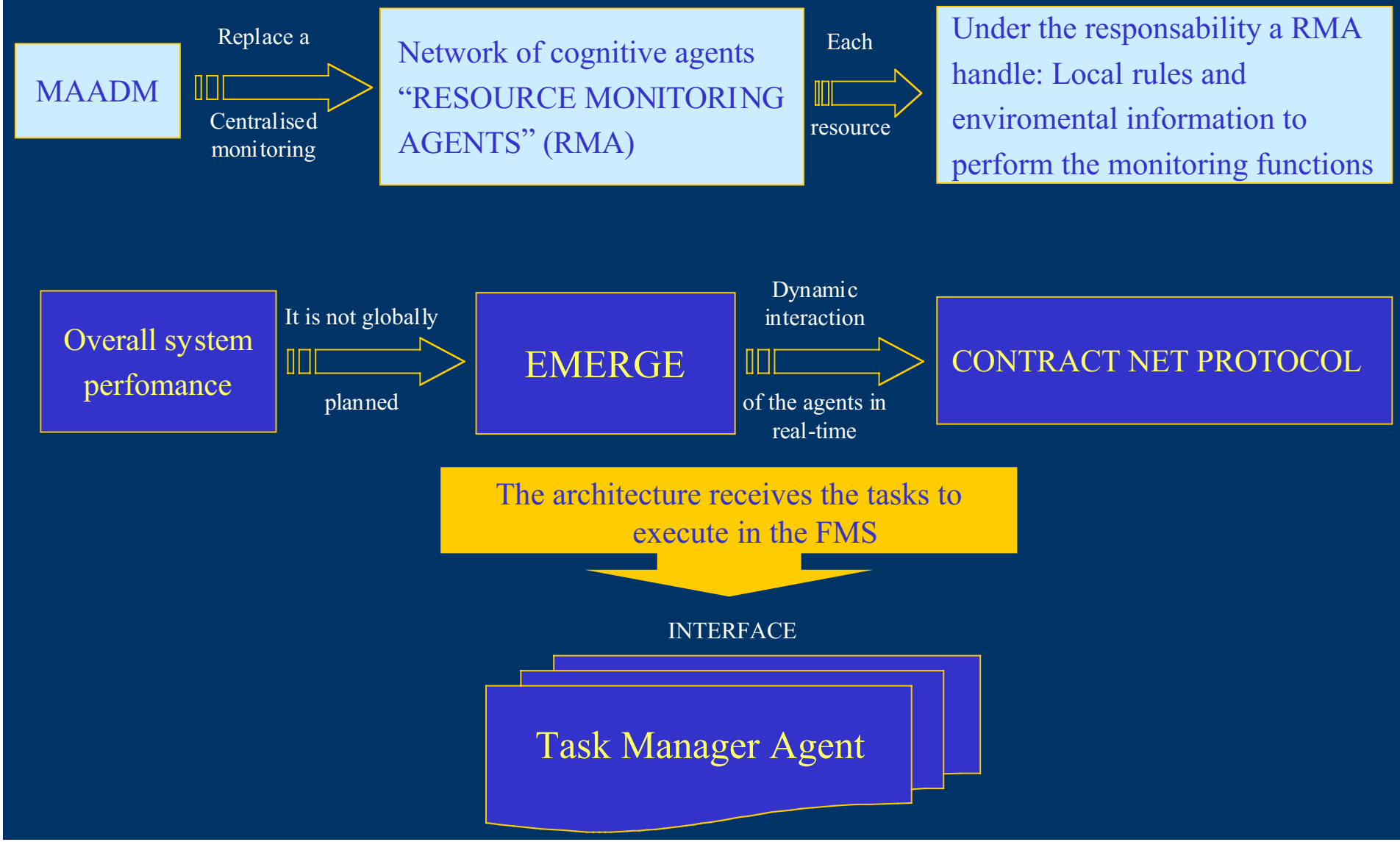
Multi-agent Architecture for Distributed and Real Time Monitoring

Monitoring systems

Functions
- Error detection
- Error diagnosis
- Error handling



Multi-agent Architecture for Distributed and Real Time Monitoring



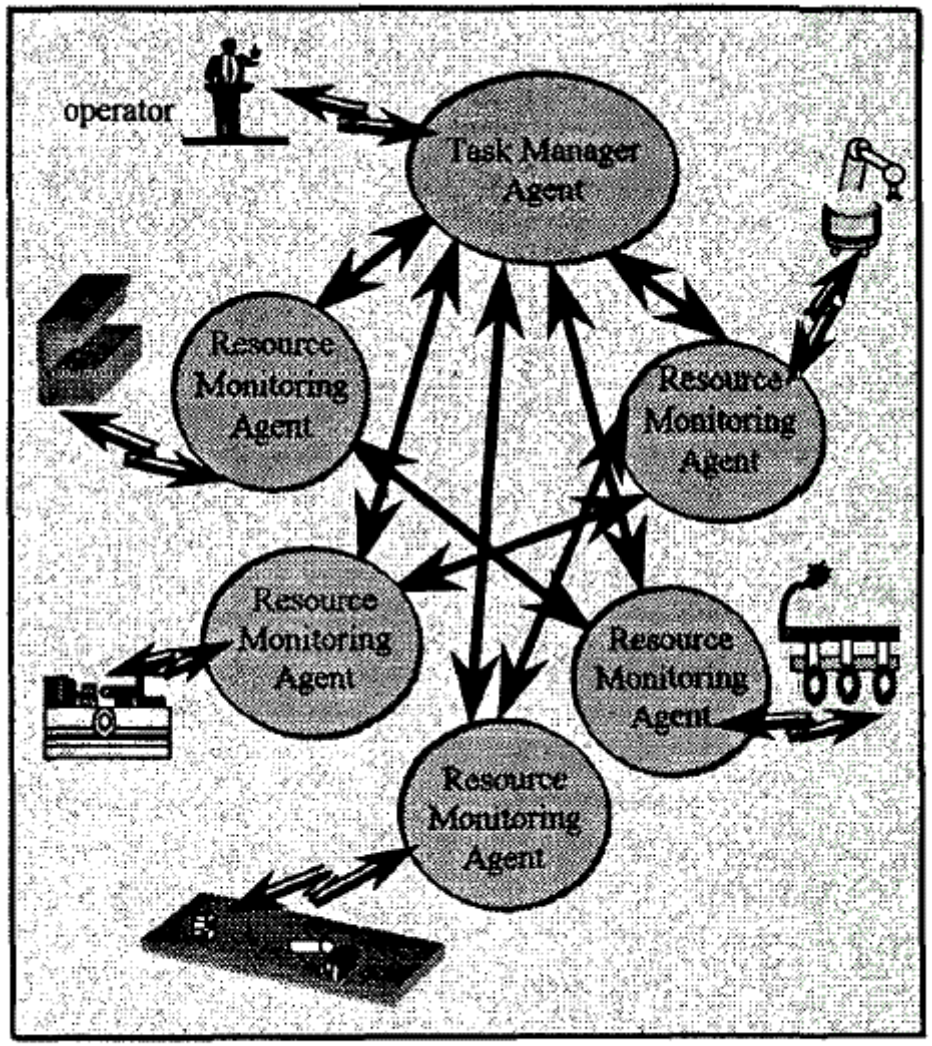


Fig. 2 Multi-agent architecture for distributed and real -time monitoring

MAA

- The monitoring of the FMS is distribuited over the RMAs
- Global Monitoring is achieved by passing appropiate messages between the RMAs
- The Contract Net Protocol is used to co-ordinate and co-operate the RMAs, and ro reshedule the operations in failure

Resource Monitoring Agent Model

Cognitive Agent Model

Distinguishes

- **Knowledge**
- **Control (Reasoning)**
- **Co-operation**
- **Communication**

Knowledge (internal and external activities)

TYPES

- **Self-Knowledge: own information for monitoring its resource (resource state database, diagnosis production rules and error hendling production rules)**

- **Other Knowledge: information about other RMAs, wich can operate whit him**

Resource Monitoring Agent Model

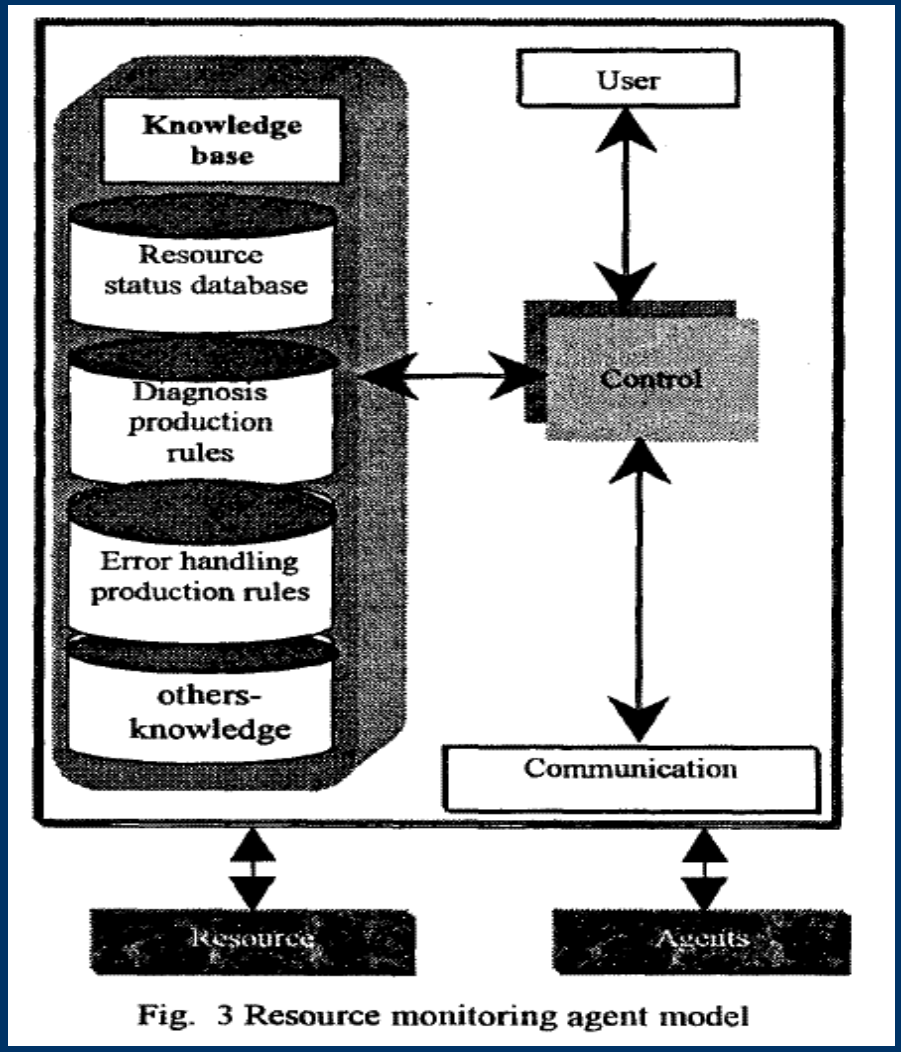


Fig. 3 Resource monitoring agent model

Resource Monitoring Agent Model

Communication

(synchronous and is done by sending messages)

TYPES OF MESSAGES

- **Information request messages: part dimensions, part weight,presence or no presence of the part, resource status, tools status, etc**

- **Failure Alert Messages: resource in failure, tool in failure, etc**

- **Cooperation and co-ordination messages: contracts, bids, acknowledgements, etc**

Resource Monitoring Agent Model

Control

(RMAs decision-making capability to acomplish the monitoring functions and to co-operate with others agents)

SET OF METHODS

- **Detection:** **It interacts with the resources status database to detect the abnormal situations ocurring on each resource.**
- **Diagnosis:** **It identifies and recognises faults, using a rule-based expert system. The knowledge bse is a set of production rules.**

  If $event_1$ and $event_n$ then $fault_1$,.....,$fault_m$
- **Error Handling:** **It determinate the corrective actions that handle the faults. They are:resheduling, urgent halt, transit to degradaded mode, operator call and actionleap**
- **Contract Net Protocol**

Resource Monitoring Agent Model

RESCHEDULING

It reschedules locally operations that couldn´t be executed by the resource in failure using the Contract Net Protocol, developed to resolve the problem of dynamic scheduling in FMS.

- Detection: It interacts with the resources status database to detect the abnormal situations ocurring on each resource.

- Diagnosis: It identifies and recognises faults, using a rule-based expert system. The knowledge bse is a set of production rules.

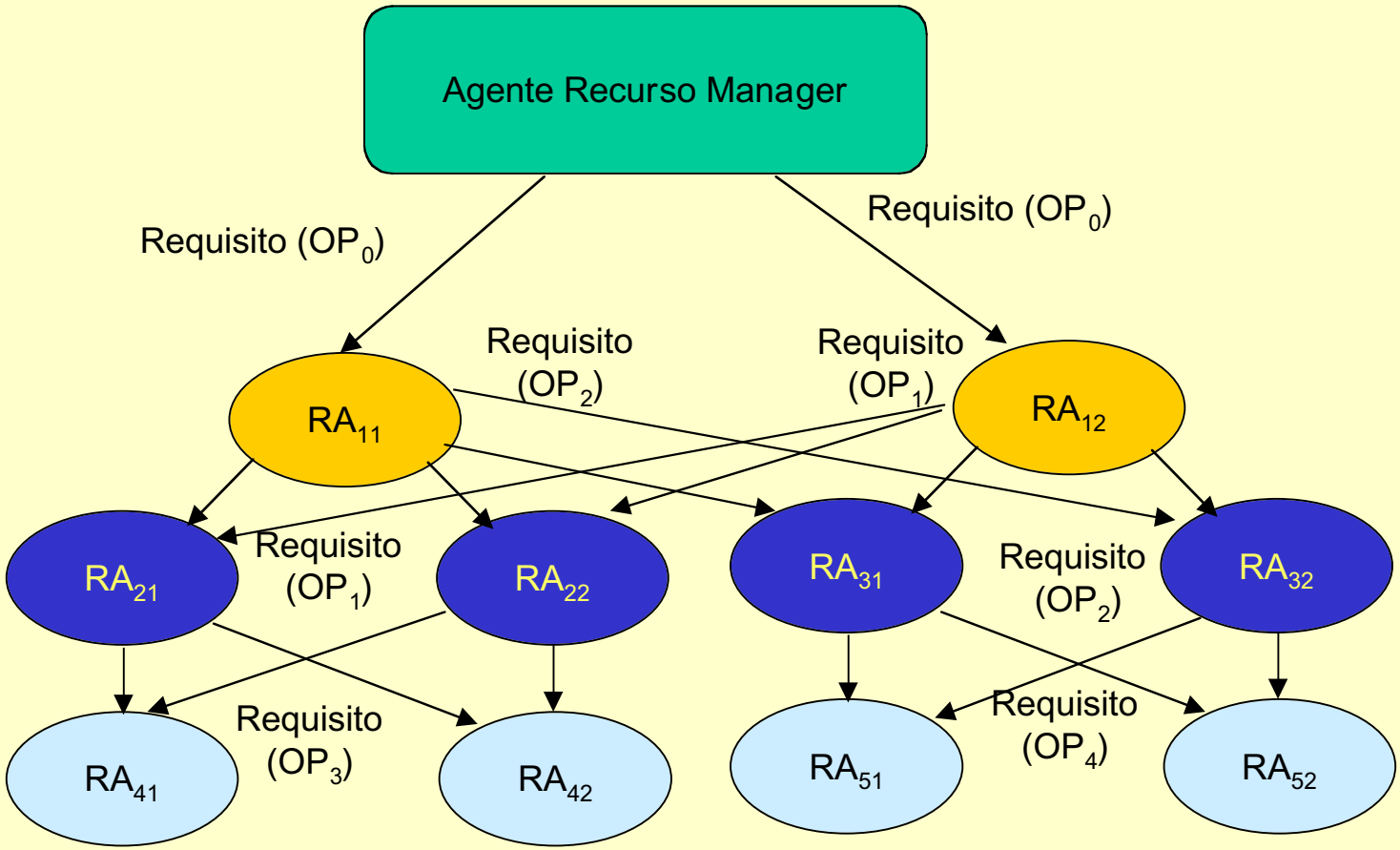
If $event_1$ and $event_n$ then $fault_1$,.....,$fault_m$

- Error Handling: It determinate the corrective actions that handle the faults. They are:resheduling, urgent halt, transit to degradaded mode, operator call and actionleap

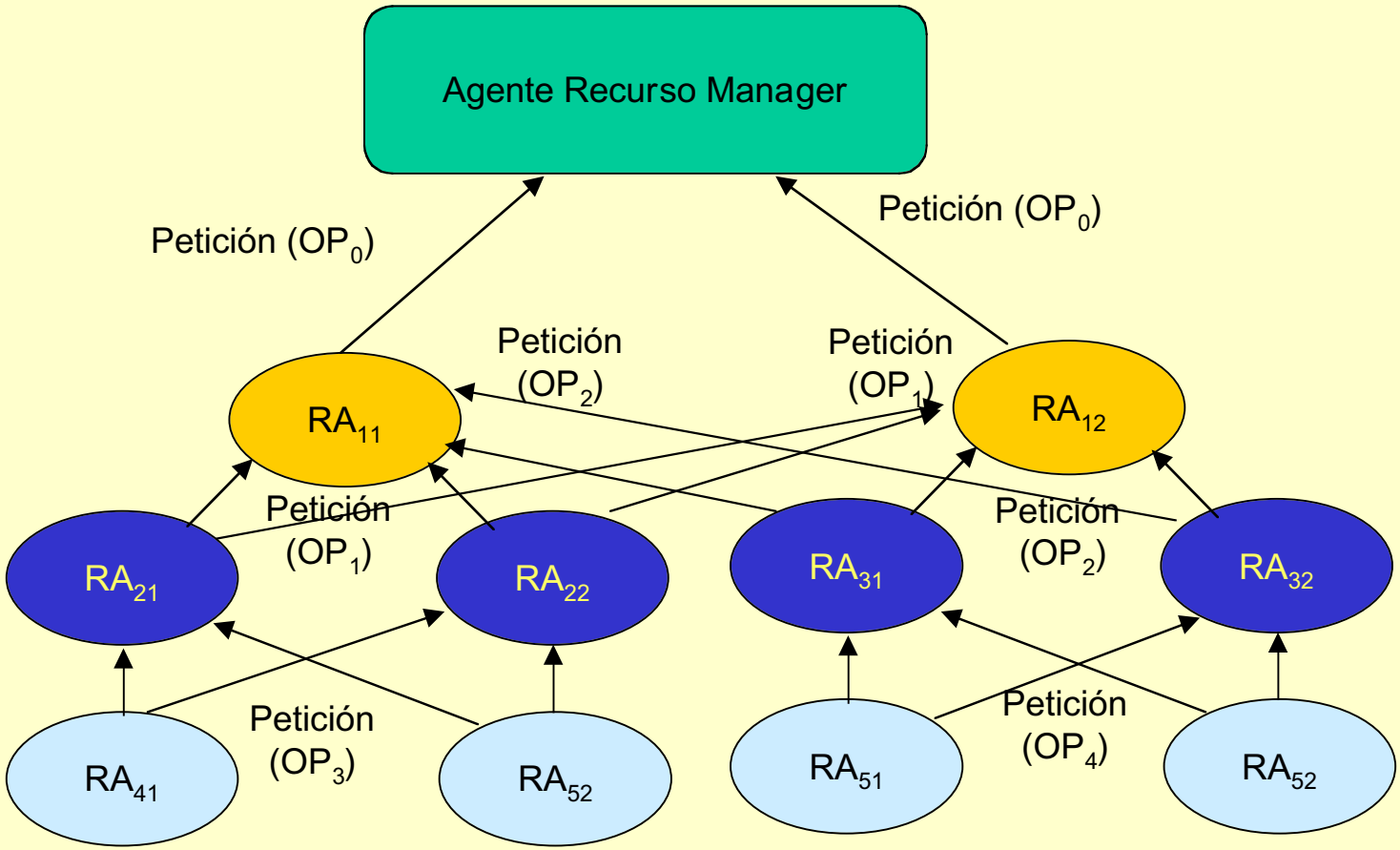
- Contract Net Protocol

Contract Net Protocol

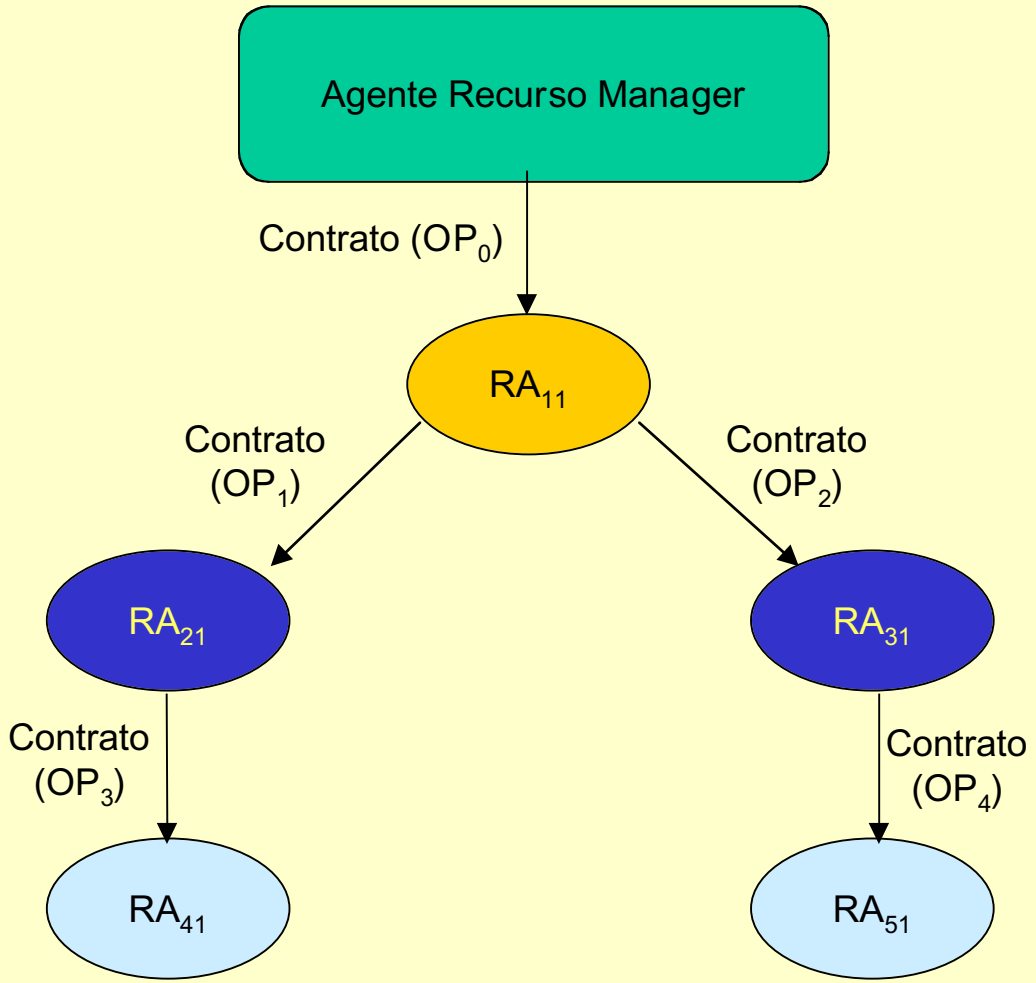
Propagación de Requisito



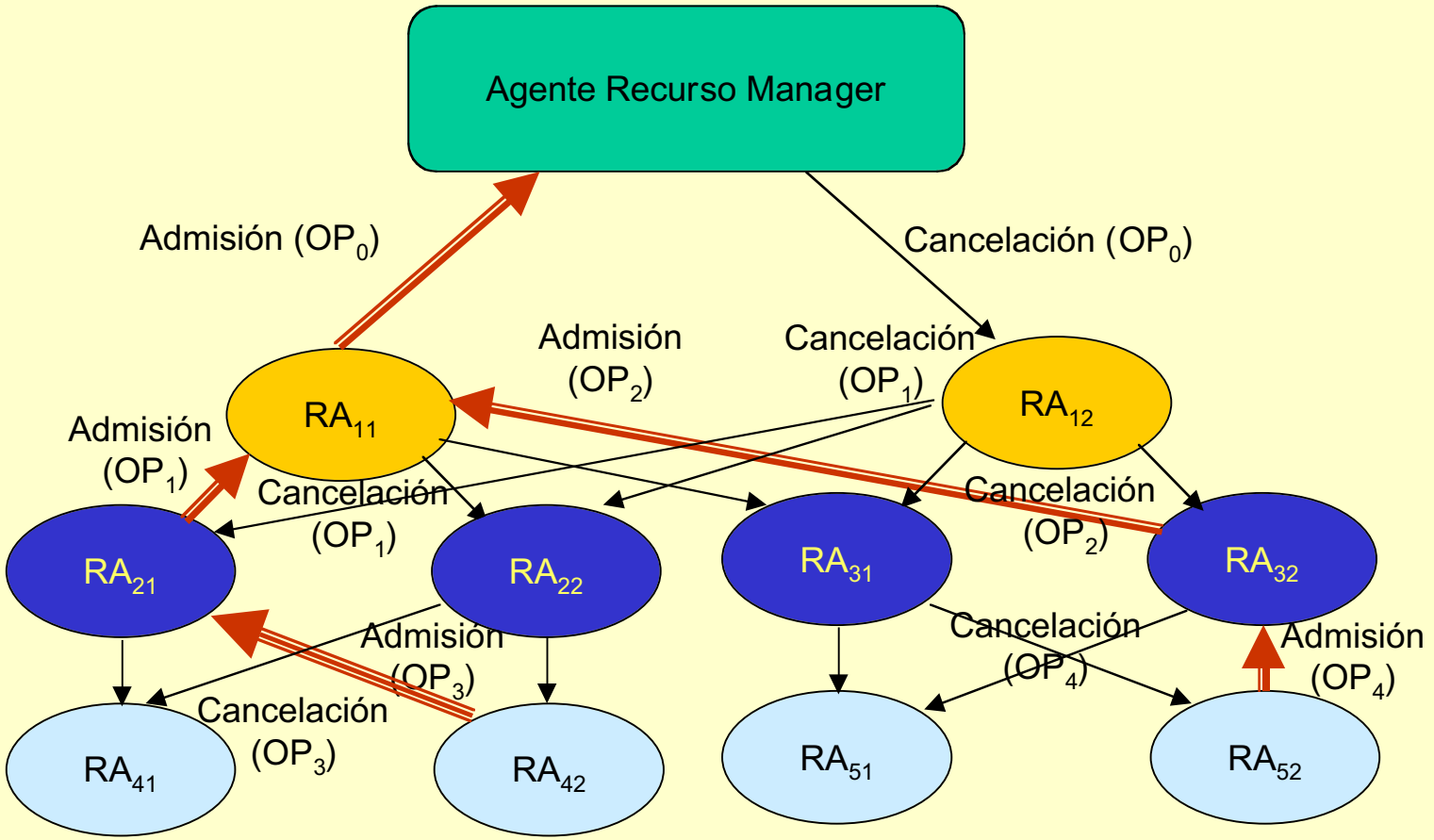
Realimentación de Petición



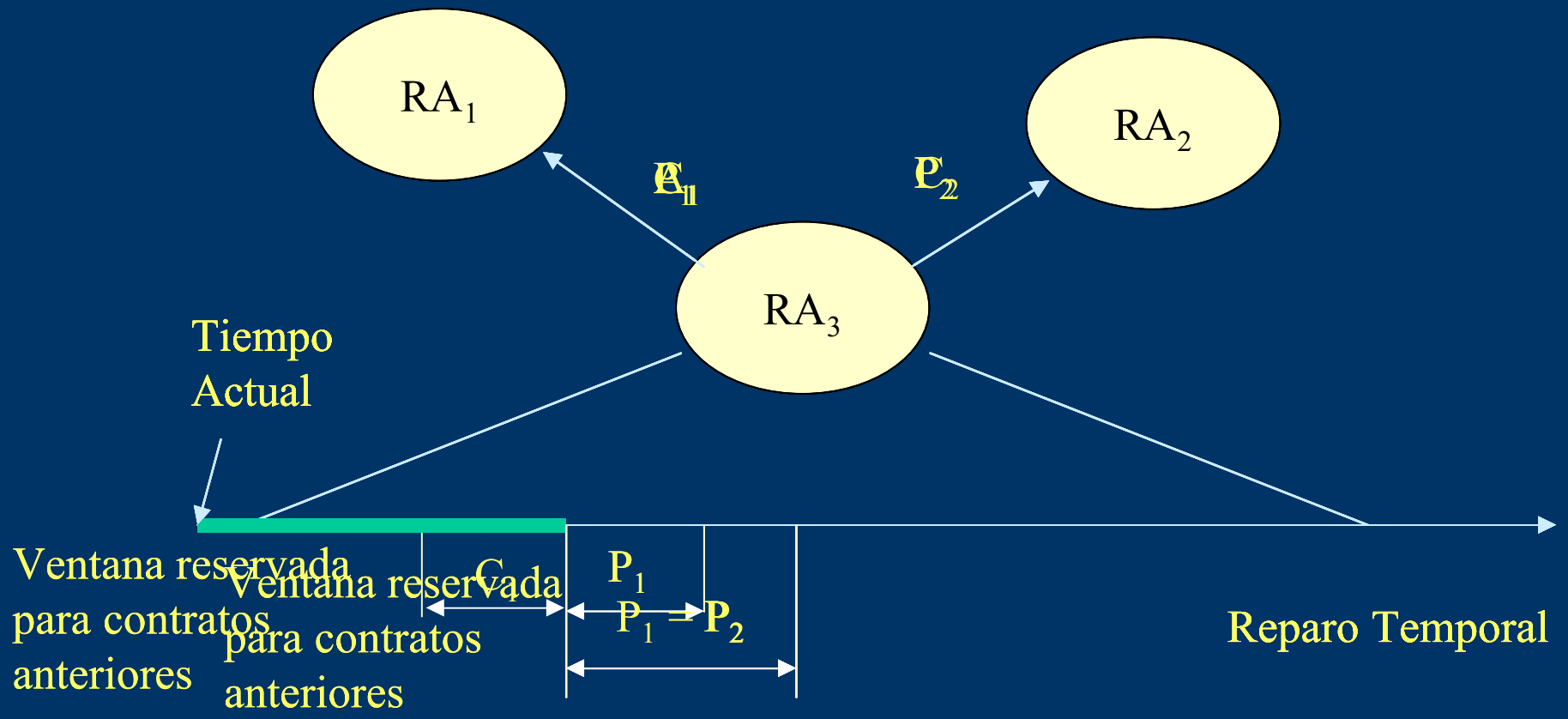
Propagación de Contrato



Realimentación de Aceptación y Cancelación de petición



Resolución de conflictos e incertidumbres



Conclusiones

Las principales ventajas de la arquitectura propuesta son flexibilidad en un ambiente variable, reconfiguración y adaptibilidad inherente, tolerancia a fallos rápida y eficiente, Desarrollo modular y autonomía operacional.

Los resultados experimentales obtenidos en la simulación, demostraron que la arquitectura multiagente is una plataforma de trabajo bastante apropiada para monitorización distribuida y en tiempo real de FMS.

La implementación de la arquitectura multiagente fue llevada a cabo en Java por ser orientada a objetos, distribuida, segura, dinámica y programación de alto desempeño.
El *Java Expert System Shell* fue integrado en el software, para el desarrollo del sistema experto basado en reglas, usado en el diagnóstico y manejo de error.