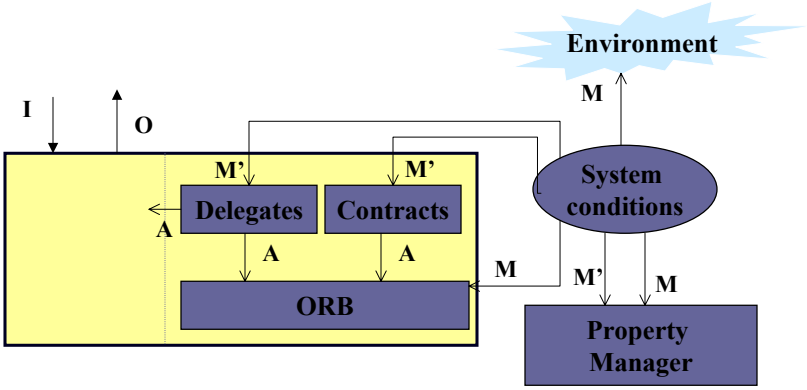
Reflective Middleware

- Application Domains
  - Distributed Systems, ubiquitous systems, mobile systems
- Mechanisms
  - Self-representation (application architecture + infrastructure + resources)
- Goals:
  - Achieve system quality attributes such as performance, reliability, etc.
- Monitor:
  - Use Interceptor
- Detection:
  - Application specific
- Resolution:
  - Application specific
- Adaptation:
  - Adaptation at meta-level propagate to implementation
  - Allow adaptation in application module(add/remove/replace/change constraints), resources and ORB level

Quality Objects Middleware

- Application Domains
  - Distributed Systems
    - Wide-area Systems
    - Embedded Systems
- Mechanisms
  - Contracts, Delegate, System Condition
- Goals:
  - Achieve required **System** quality attributes such as performance, reliability, etc.
- Monitor:
  - QoS variables explicitly specified in Contracts
- Detection:
  - QoS variables change and are evaluaed by
    - In-bound – By Delegates
    - Out-of-bound – By Contracts and System conditions
- Resolution:
  - Determine suitable behavior in Delegate or Contracts
- Adaptation
  - Change of Delegate behavior, action in Contracts

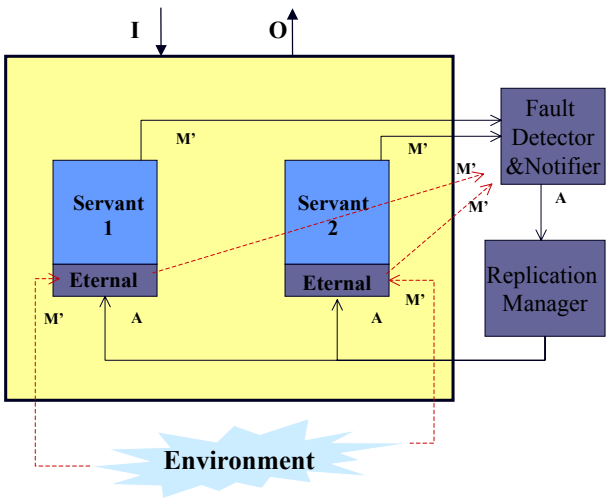
Quality Objects Middleware



Eternal System

- Application Domains
  - High reliability distributed Systems
- Mechanisms
  - Interceptors, replication management
- Goals:
  - Transparently fault tolerant
- Monitor:
  - Push/pull heart-beat
  - External Interceptor
- Detection:
  - Fail to get heart-beat
- Resolution:
  - Fault report is sent to relevant subscriber for suitable action
- Adaptation:
  - Recovery actions are executed

Eternal


I
O
Fault
Detector
&Notifier
M'
M'
M'
M'
A
Servant
1
Servant
2
Eternal
Eternal
Replication
Manager
M'
A
A
M'
Environment