

A Reflective Framework for Self-Management of Applications and Middleware

Reza Haydarlou
Benno Overeinder
Frances Brazier

Intelligent Interactive Distributed Systems Group
Department of Computer Science
Vrije Universiteit Amsterdam

Problem:

Increasingly complex systems

- composed of a variety of components
- operating in large-scale distributed heterogeneous environments
- require more and more human skills to install, configure, tune, and maintain

Approach:

Self-managing systems which are self-aware:

- determine when and where an error state occurs
- analyse the error state in its specific context
- devise a strategy to acquire more information or to solve the problem
- use the knowledge they have of the processes at hand and (partial) solutions

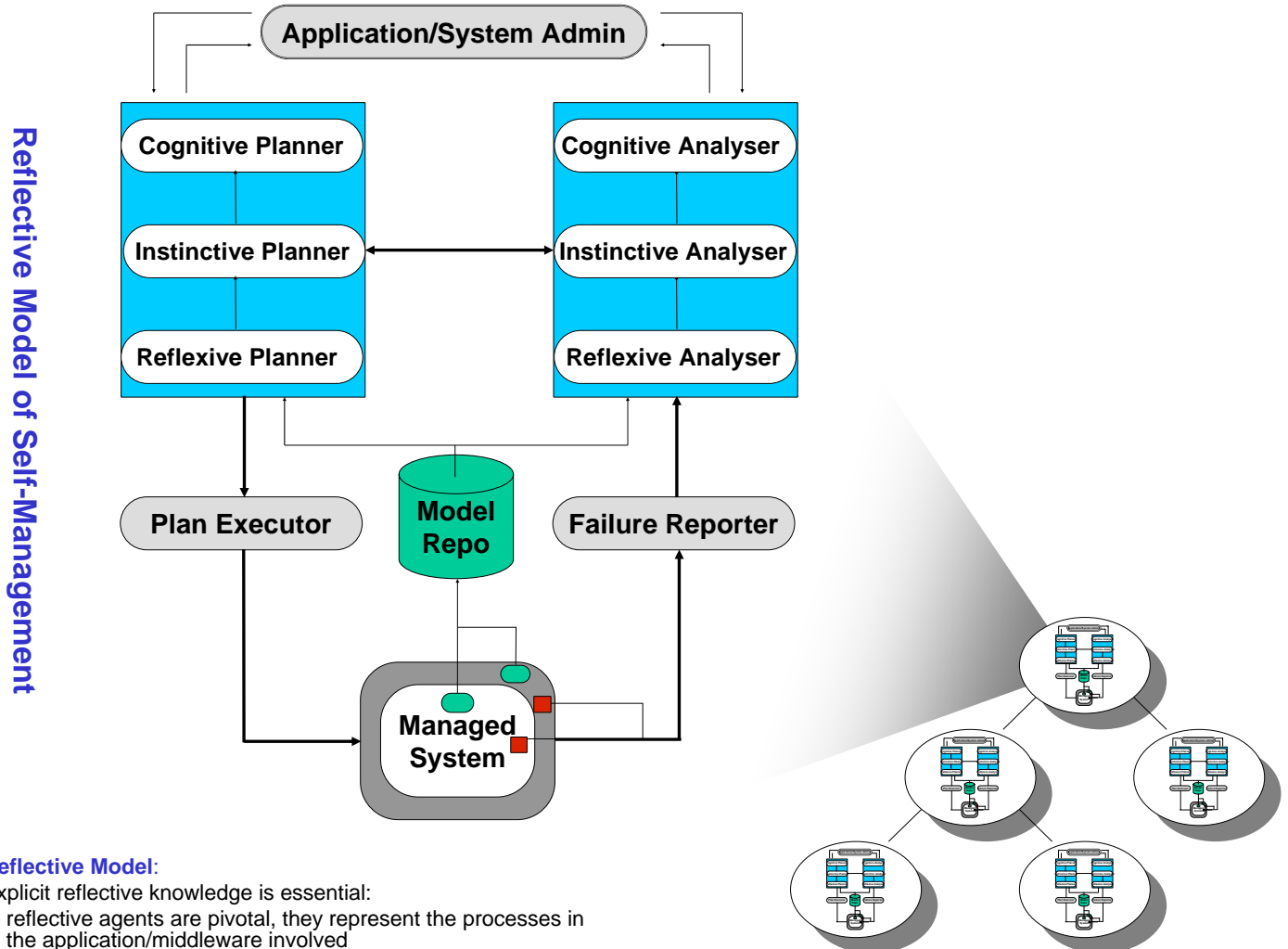
Self-Managing Tasks:

- Self-configuration
- Self-healing
- Self-optimization
- Self-protection
- Self-*

Diagnosis:

Determining the root cause:

- runtime failures, configuration errors, or reduced performance and scalability
- hindered by the lack of appropriate diagnostic feedback to human users



Reflective Model:

Explicit reflective knowledge is essential:

- reflective agents are pivotal, they represent the processes in the application/middleware involved
- reflective agents are capable of reasoning not only about their external environment, but also about their own behavior and other agents behavior at different meta-levels:
 - cognitive analysis and planning
 - instinctive analysis and planning
 - reflexive analysis and planning
- reflective agents are associated with middleware services, applications, middleware/application components, and constituent classes for diagnosis and repair at the appropriate level

Self-Management Architecture:

Managed elements and their compositional organization. E.g., top level is web server farm composed of a number of servers (hosts), each running a number of services.

Next levels can be broken down into components of the services, and the classes of the components.