MadKit is a versatile agent platform, designed to support heterogeneous agent and communication models, and host multiple distributed applications.

The platform architecture is based on a minimalist agent kernel decoupled from specific agent or communication models. Services like distributed message passing, migration or monitoring are provided by individual agents for maximal flexibility.

A componential graphical interface model enables variations in platform look&feel and classes of usage.

**Architecture**

The MadKit kernel is a small agent engine that only manages the most basic functions in the platform: messaging, global structuration and agent lifecycle. It is completely decoupled from specific agent models and graphical user interface.

Its small size and adaptability makes it compatible with the smallest devices (such as the Java Platform Micro Edition running on a Palm PDA).

MadKit makes it easy to add a specific graphic layer or to embed a multi-agent system in a legacy application. Any agent may define an associated GUI (usually a standard Java bean) that will be managed by the graphic host application. For instance, changing the host application from the development environment to the final application does not require any agent or GUI code modification.

Platform services are designed as individual agents that extend the kernel functionality. Any system agent can be replaced (for instance from a socket-based to a CORBA communication agent), to tailor the platform to specific application requirements.

There is no “MadKit agent architecture”: the agent model is intentionaly weak to ease integration of various classic agent models, while providing a group/role methodology for application design.

Specific agent libraries have been built on this generic architecture, from legacy-wrapper agents directly coded in Java to Scheme or CLIPS-rules agents. A synchronous engine permits custom scheduling and observations tasks for applications where thousands of agents are necessary.

**Some applications**

- Multi-robot simulation framework
- Knowledge Management Tools
- Federated search engines
- Social networks research
- Reactive/cognitive hybrid architectures for robot control
SEdit - A Structure Editor

SEdit is a powerful agent design environment based on the MadKit platform. It provides graphical tools for the various models that appear in agent-oriented application design.

The agent developer chooses a model, or defines it himself in a XML file. This model description is then interpreted by SEdit to automatically setup a custom graphical editor. These editors can be extended with Java classes to define “active” formalisms (petri nets with simulation, code generators, ...). Structures edited in different models can also be modularized and combined to ease reuse, or directly executed as agents in the runtime environment.

As SEdit is itself a MadKit application, editors (which have been developed as a set of specialized agents) are fully integrated in the development environment and may be embedded in any agent application.

Platform highlights

- Monitoring agents: kernel status, organization views, group analyzers
- Distributed modes through sockets or CORBA communications
- Synchronous agents with probes and schedulers, and libraries of generic analysis tools.
- Agent configuration and organization descriptions defined through XML DTD.
- Bindings to external languages and library: Scheme, CLIPS rules
- Extensible message descriptions
- Kernel decoupled from AWT
- Customizable graphical development environment
- MadKit versions for non standard Java environment: MacOS X, PalmOS KVM.
- Agent-based security through agent groups

Requirements

MadKit kernel
Java Platform Micro Edition or higher
Environement & standard agents:
JDK 1.1.x or later

For more information

<http://www.madkit.org>
Olivier Gutknecht <olg@madkit.org> - Jacques Ferber <ferber@madkit.org>

Laboratoire d’Informatique, Robotique et Micro-Electronique de Montpellier (LIRMM)
161, rue Ada- 34392 Montpellier Cedex 05 - France