

MobilityTools

**A Toolbox for Agent Mobility and
Interoperability Based on OMG
Standards**

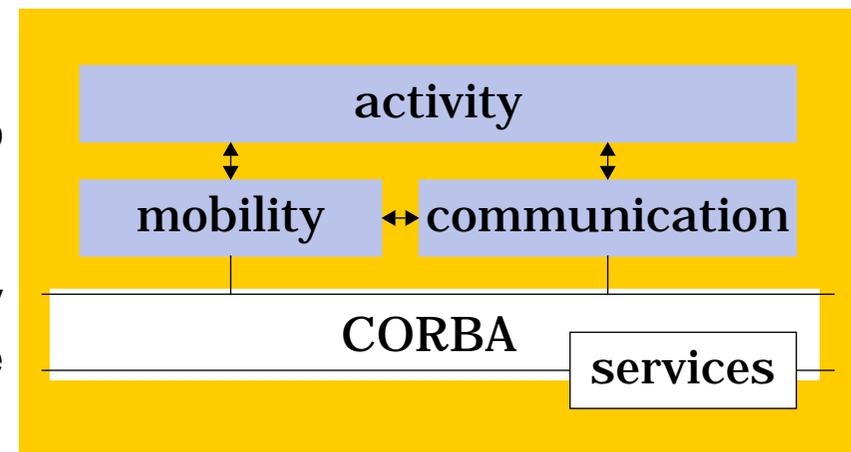
Why MobiliTools?

Usual drawbacks of mobile agent platforms

- Mobile agent platforms usually enforce an agent framework with a dedicated mix of activity model and communication model;
- Unfortunately, there is no universal agent framework meeting every requirement (scalability, reactivity, robustness, reliability, pro-activity, adaptability...);
- Standards are still lacking for making heterogeneous agent inter-operate (at both model and implementation levels).

MobiliTools specific approach

- use distributed system standards (OMG) to help interoperability;
- separate mobility, communication and activity support into independent customizable components to provide a flexible toolbox.



SMI, the Simple MASIF Implementation (1)

SMI implements OMG's Mobile Agent System Interoperability Facilities.

- class `Agency` implements MASIF's `MAFAgentSystem` CORBA interface and provides a customizable Java based environment for agent execution and management;
- class `Finder` implements `MAFFinder` interface and provides advanced region management facilities.

GUI for Agency

The GUI for Agency consists of two windows. The main window, titled "SMI agency Asama (dillense), region demo", contains buttons for "update", "create", and "shutdown" at the top, and "view", "suspend", "resume", "move", and "terminate" at the bottom. A list of agents is displayed in the center, showing "Elmer (dillense)" and "Mike (dillense)". A dialog window titled "Move agent Elmer (dillense)" is open, showing fields for "destination for agent Elmer (dillense)", "region" (set to "demo"), "agency" (set to "Zurich"), and "place". It has "move" and "stay" buttons.

GUI for Finder

The GUI for Finder consists of two windows. The main window, titled "SMI Finder for region demo", contains a table with columns "Name" and "Location". The table lists:

Name	Location
agency Asama (dillense)	CosNaming:MAF/demo/agency/Asama
agent Elmer (dillense)	CosNaming:MAF/demo/agency/Asama
agency Zurich (dillense)	CosNaming:MAF/demo/agency/Zurich
agent Mike (dillense)	CosNaming:MAF/demo/agency/Zurich

 Below the table are buttons for "new agent" and "terminate". A dialog window titled "agent Elmer (dillense)" is open, showing fields for "Name: Elmer", "Authority: dillense", "Type: 10", "Location: CosNaming:MAF/demo/agency/Asama", and "Activity: running". It has "move", "suspend", and "terminate" buttons.

SMI, the Simple MASIF Implementation (2)

No dedicated agent framework but MASIF's concepts and a generic life cycle.

- Agents must implement interface `MobileObject` to be managed by SMI agencies;
- `MobileObject` call-backs: `afterBirth()`, `beforeMove()`, `afterMove()`, `afterMoveFailed()`, `beforeShutdown()`, `beforeDeath()`, `beforeSuspend()`, `beforeResume()`.

What about agent activity?

- Agent activity has to be handled through the lifecycle defined by `MobileObject`;
- passive agent, pro-active agent (one thread per agent), and reactive agent (see synchronous programming languages) models have been developed.

What about agent communication?

- whatever socket, CORBA, Java RMI, JavaSpace...;
- why don't try ACTS and its personalities?

ACTS, the Agent Communication Transport Service (1)

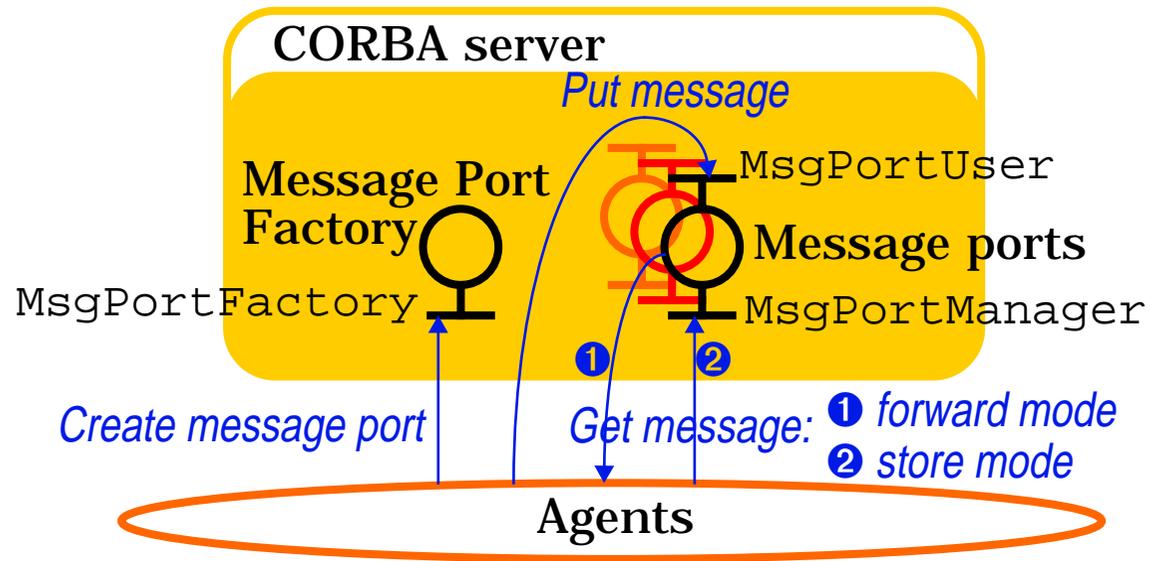
Why a CORBA service for transporting messages?

Issue	ACTS answer
MASIF doesn't address agent communication, while agent interoperability is a problem.	ACTS defines interfaces to transport communication between heterogeneous agents via CORBA, taking mobility into account.
In some cases, agent mobility makes communication delivery dramatically complex.	ACTS's pragmatic approach is to spread an infrastructure of static <i>message ports</i> to store messages. Mobility awareness makes it reliable.
There is a variety of agent communication styles.	ACTS is highly versatile and customizable: "any" type messages, with both store and forward delivery modes. Our Mailbox and FIPA personalities show advanced utilizations of ACTS.
Should agents using ACTS be CORBA objects?	Agents need not be CORBA objects to get messages from a message port, but agents that are CORBA objects can have their messages forwarded to them on the fly (forward mode).

ACTS, the Agent Communication Transport Service (2)

How does it work?

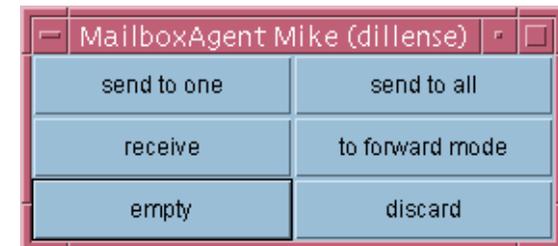
- static message ports store (MsgPortUser interface) and deliver (MsgPortManager interface) messages;
- message port factories create message ports on demand (MsgPortFactory interface).



Use case: ACTS Mailbox personality

Mailbox personality wraps message ports and CORBA interfaces into a user-friendly Java utility, managing high-level addresses (Email-like, inline with MASIF's region concept), multicast and unicast, forwarding and mobility.

Mailbox GUI



7

Availability

Requirements

Java 2; CORBA-compliant ORB: e.g. Java2, Jonathan/David (<http://www.objectweb.org/>)

Licence

MobiliTools belongs to France Télécom R&D. It is available free of charge for research purpose on signed official demand (form available).

Contact

Bruno Dillenseger, France Télécom R&D, BP98, F-38243 Meylan CEDEX.
bruno.dillenseger@rd.francetelecom.fr

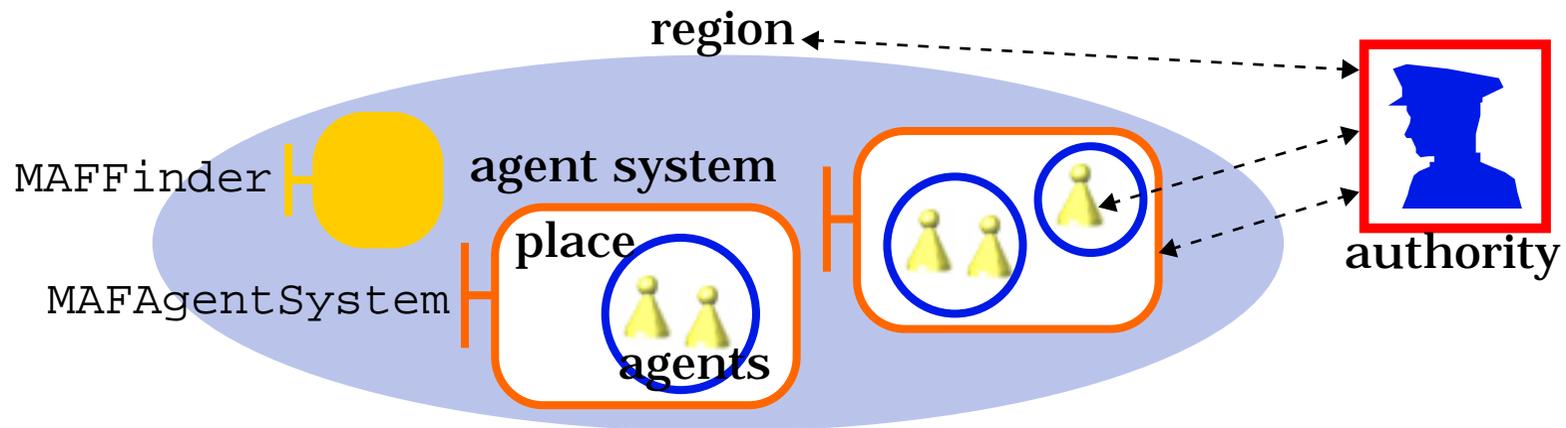
Future plans

ACTS enhancements, agent activity models and engines for SMI, MASIF interoperability investigations...

MASIF in short...

The Mobile Agent System Interoperability Facilities specifications were adopted by the Object Management Group in February 1998.

MASIF defines a conceptual framework...



... and two CORBA interfaces

- MAFAgentSystem for managing an agent execution environment and its agents;
- MAFFinder to register and lookup agents.