

Behavior Model of Mobile Agent Systems

Authors : **Pr. Serge Chaumette**
: **Dr. Pierre Vignéras**

Team : Distributed Systems and Objects
Laboratory : LaBRI – Bordeaux, France
Head : Pr. Serge Chaumette

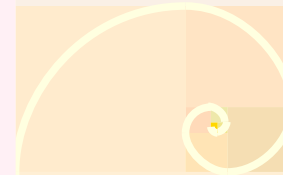
The 2005 International Conference
on Foundations of Computer Science

Monte Carlo Resort,
Las Vegas, Nevada,
USA

June 27-30, 2005

Outline

- I. Mobile Code Paradigms
- II. Mobile Agents
- III. Active Containers
- IV. Conclusion





I. Mobile Code Paradigms

Mobile Code Paradigms (1/5)

[Fugetta et al. 1998]

Paradigm	Before		After	
	Site A	Site B	Site A	Site B
Client/Server <i>non mobile</i> (HTTP, X)	A	code B ressource	A	code B ressource
Remote Evaluation (rsh, PostScript)	code A	ressource B	A	<i>code</i> B ressource
Code on Demand (Java Applet, Flash)	ressource A	code B	<i>code</i> A ressource	B
Mobile Agent	code A	ressource	code A ressource	<i>code</i> A ressource

A, B: execution entities (*thread, process*)

code: move

C: executor

Mobile Code Paradigms (2/5)

[Fugetta et al. 1998]

Paradigm	Before		After	
	Site A	Site B	Site A	Site B
Client/Server <i>non mobile</i> (HTTP, X)	A	code B ressource	A	code B ressource
Remote Evaluation (rsh, PostScript)	code A	ressource B	A	code B ressource
Code on Demand (Java Applet, Flash)	ressource A	code B	code A ressource	B
Mobile Agent	code A	ressource	code A ressource	code A ressource

A, B: execution entities (*thread, process*)

code: move

C: executor

Mobile Code Paradigms (3/5)

[Fugetta et al. 1998]

Paradigm	Before		After	
	Site A	Site B	Site A	Site B
Client/Server <i>non mobile</i> (HTTP, X)	A	code B ressource	A	code B ressource
Remote Evaluation (rsh, PostScript)	code A	ressource B	A	code B ressource
Code on Demand (Java Applet, Flash)	ressource A	code B	code A ressource	B
Mobile Agent	code A	ressource	code A ressource	code A ressource

A, B: execution entities (*thread, process*)

code: move

C: executor

Mobile Code Paradigms (4/5)

[Fugetta et al. 1998]

Paradigm	Before		After	
	Site A	Site B	Site A	Site B
Client/Server <i>non mobile</i> (HTTP, X)	A	code B ressource	A	code B ressource
Remote Evaluation (rsh, PostScript)	code A	ressource B	A	<i>code</i> B ressource
Code on Demand (Java Applet, Flash)	ressource A	code B	<i>code</i> A ressource	B
Mobile Agent	code A	ressource	code A ressource	<i>code</i> A ressource

A, B: execution entities (*thread, process*)

code: move

C: executor

Mobile Code Paradigms (5/5)


[Fugetta et al. 1998]

Paradigm	Before		After	
	Site A	Site B	Site A	Site B
Client/Server <i>non mobile</i> (HTTP, X)	A	code B ressource	A	code B ressource
Remote Evaluation (rsh, PostScript)	code A	ressource B	A	code B ressource
Code on Demand (Java Applet, Flash)	ressource A	code B	code A ressource	B
Mobile Agent	code A	ressource	code A ressource	code A ressource

A, B: execution entities (*thread, process*)

code: move

C: executor



II. Mobile Agents

Mobile Agents: Definition

- An agent is a program which is both **autonomous** and **independant**
 - **autonomous**: it controls its own execution
 - **independant**: it holds its own thread
- An agent is **mobile** if it can move from one execution environment (server) to another.
 - *strong migration*: resumes at the next instruction (requires saving the instruction pointer)
 - *weak migration*: resumes using an event mechanism

Mobile Agent Frameworks

- Precursor in 1996: *Telescript* -- General Magic
 - Special language
- Nexts are essentially Java-based frameworks
 - Odyssey (General Magic, successor of Telescript), Voyager (ObjectSpace now Recursion Software), Concordia (Mitsubishi), Aglets (IBM), ...
- Interoperability
 - Standard MASIF (OMG): implemented by Grasshopper

Modelisation (π -calculus)

Mobile Agent
System Model
(Aglets-inspired)

equivalence
(bisimulation)?

Mobile Agent
System Model
(Aglets-inspired)

Security problems
underscored
(system ressources
access by agents)

Open Question!

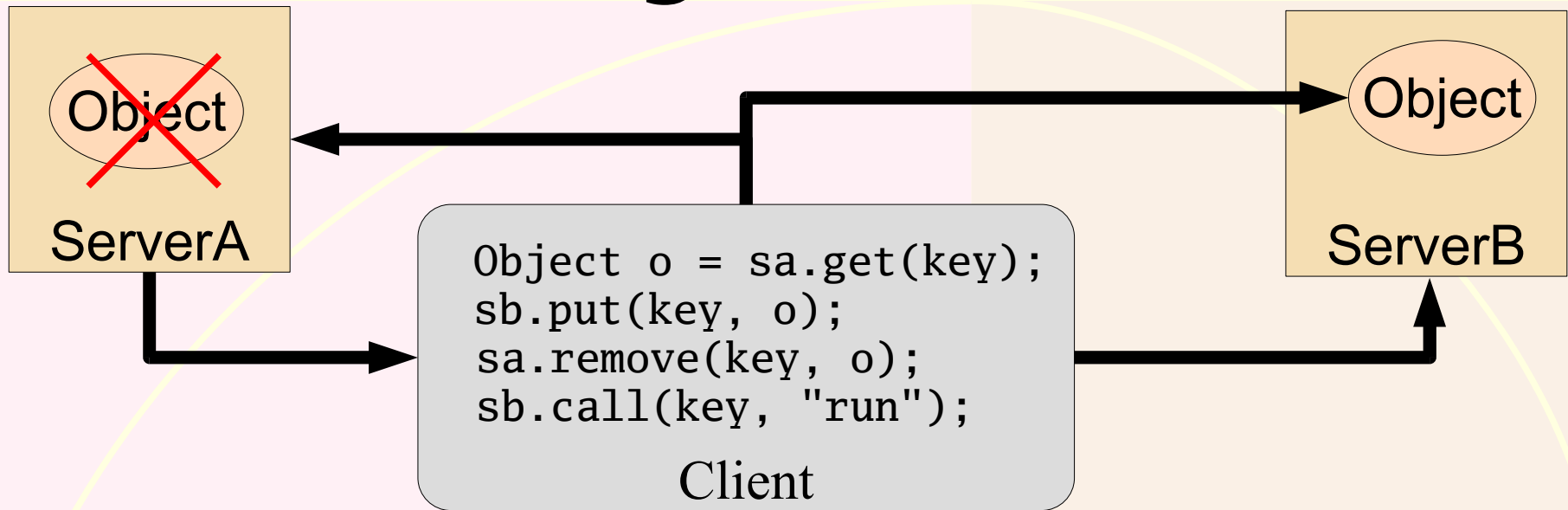
Active Container
Model

- Other calculus may be used for the proof
 - Ambients, join-calculus (Chemical Abstract Machine)
- Proof tools (HOL, Pict)

The slide features a decorative background with a golden spiral on the right side, set against a light blue and white grid. A large, light blue rounded rectangle with a dark blue border and a drop shadow is centered on the slide. Inside this rectangle, the text "III. Active Containers" is written in a bold, black, sans-serif font.

III. Active Containers

Active Container: a Base Mechanism for Code Migration



When Client == Object,
we have an agent!

These four primitives
do exist in almost all systems,
but they are not publicly available
→ **JACOb: Java implementation**

Container: put(), get(), remove()
Activity in the container: call()
asynchronous call: **parallelism**

Dynamism

Use Cases of the Active Container Model

[chaumette, vignéras PARCO'2003]

- Memory Model
 - put(): create a *stored object* (new operator)
 - remove(): remove a *stored object* (garbage collector)
 - get(): retrieve a copy of a *stored object* (clone())
 - call(): invoke a method of a *stored object* (o.m())
- Application Deployment
 - Deployment of components of a parallel application
 - Software network installer
- Plugins for applications
- **Asynchronous Remote Method Invocation**



JACOb: a Java implementation of active containers (1/2)

III. Active Containers

Defines the active container

Familiar to developers

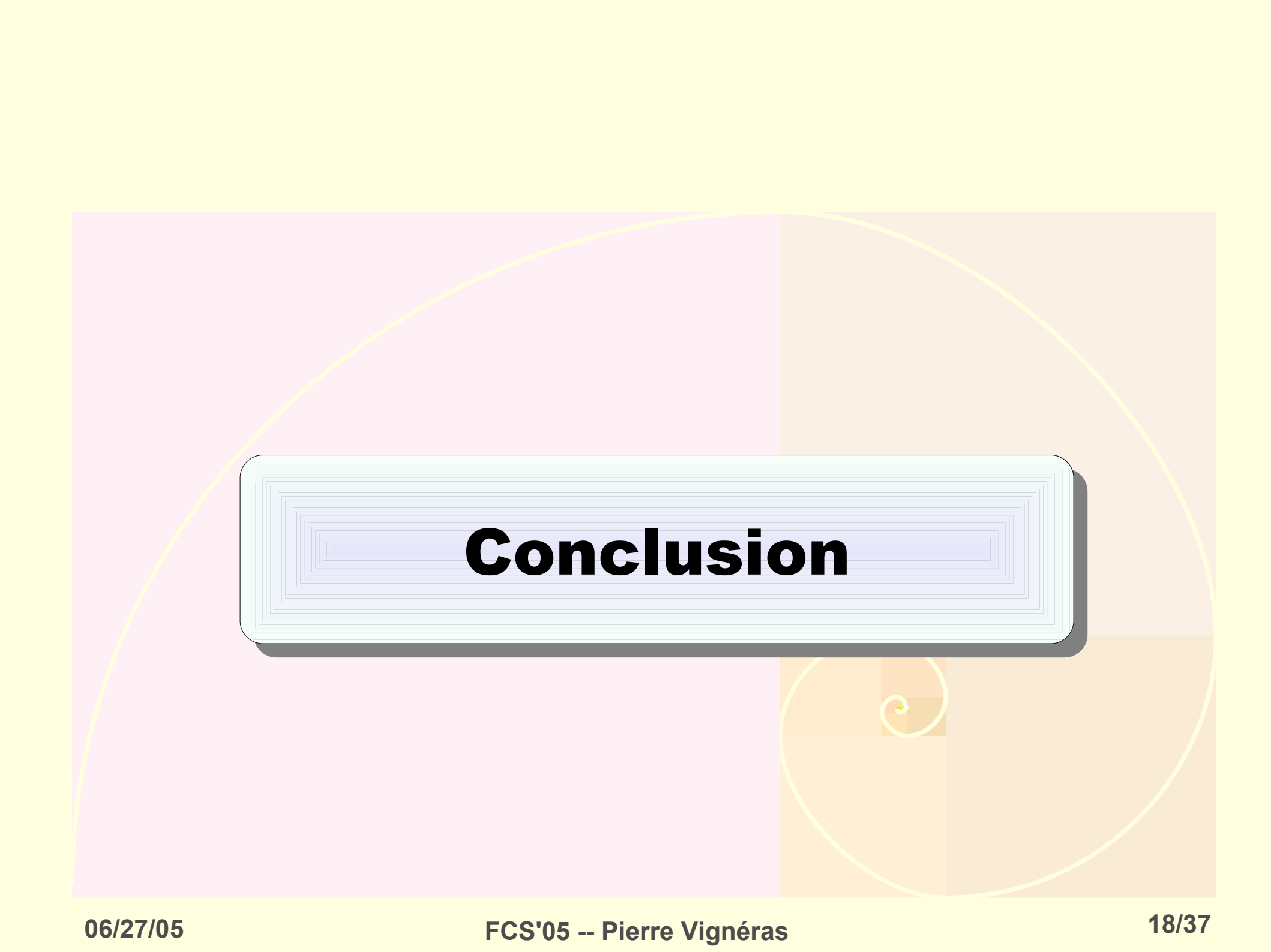
```
public interface ActiveMap extends java.util.Map {  
    void call(Object key,  
              java.lang.reflect.Method method,  
              Object[] args,  
              Future future);  
}
```

Use of reflection
(*dynamism*)

Asynchronous Method:
early reply

JACOb: a Java implementation of active containers (2/2)

- Local implementation
 - Business (container) and remote layer separation
- Remote implementation
 - Global, event-based remote exception handling
 - still allows the traditionnal (by invocation) RMI exception handling
 - Multi-protocols
 - RMI, TCP and UDP available
 - No limit (Myrinet, SOAP or **JToe** possible)

The background features a golden spiral on the right side, set against a light blue and white grid. A large, light blue rounded rectangle with a dark blue shadow is centered on the page, containing the word "Conclusion" in bold black text.

Conclusion

Mobile Agents: a Good Idea?

- In [Harrison and al. 1995] (IBM)
 - Cons: it is always possible to write an equivalent application *without the use of the mobile agent paradigm* (no "killer application"!)
 - Pros: this paradigm allows the development of many applications that would require *many distinct paradigms*
 - Ease of design, development and deployment
 - Used for system and network management (SNM) [Reuter and Baude 2002]
- IBM starts the design of the Aglets system

Mobile Agents: reasons of a "failure" (1/2)

- Technical reasons
 - Internal security problems
 - protection of an agent from its server
 - maximal risk in case of bugs for a server (*worms*)
 - disk, memory, CPU pollution
 - ease of infection due to the *mobile* aspect
 - Interoperability
 - migration type (strong/weak, proactive/reactive)
 - language used
 - security management, ...
- Too many choices, too many risks

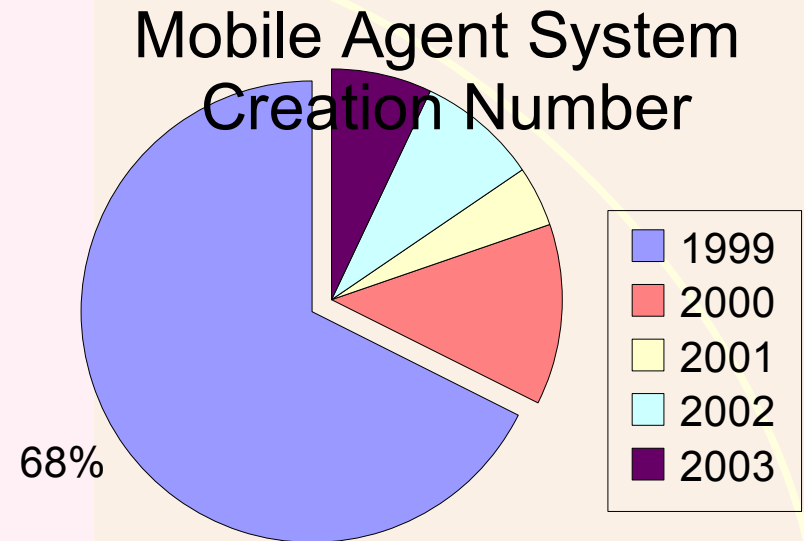
Mobile Agents: reasons of a "failure" (2/2)

- Non-technical Reasons
 - Modes of Remuneration
 - Enterprises: advertisement-based *business model*
 - a mobile agent is not a user, it is not sensible to adverts!
 - Individuals: no *killer* application
 - Quality of Service
 - Service provider distinct from the agent provider
 - perception of the service by the user depends on the agent (common case on the web: display depends on navigators)
 - Unknown Paradigm in the business world

Facts: from Mobile Agents to Asynchronous Execution Frameworks

Mobile agent list

- Last update: 01/2004
- 71 mobile agent systems
- 54 % links are invalids today
- Most standards are no more supported (Concordia, Odyssey, AgentOS, ...)



Growing success of distributed frameworks based on communicating (possibly asynchronous) components (possibly mobile)

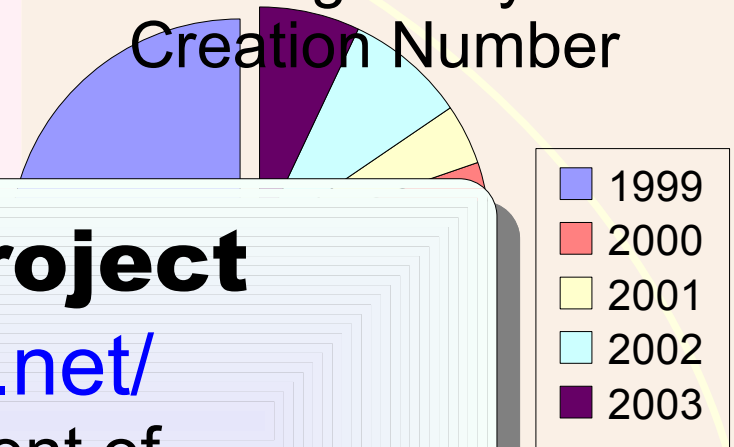


Facts: from Mobile Agents to Asynchronous Execution Frameworks

Mobile agent list

- Last update: 01/2004
- 71 mobile agent systems
- 5
- M

Mobile Agent System Creation Number



The Mandala Project

<http://mandala.sf.net/>

Eases the development of concurrent and/or distributed applications

Growth

frameworks based on communicating (eventually asynchronously) components (eventually mobile)

CORBA
.NET
EJB