Un’Infrastruttura per la Mobilità in AgentService
(Implementing Mobility in AgentService)

Christian Vecchiola, Alberto Grosso, Antonio Boccalatte
DIST – Università degli Studi di Genova
{christian, agrosso, nino}@dist.unige.it
Outline

- Introduction
- AgentService
- Mobility Services
- Load Balancing Services
- Conclusions
Introduction

Mobile Code

Definition
- the ability to move a computational unit from one site to another and continue its execution
- **strong mobility**: complete restoration of the execution state
- **weak mobility**: partial restoration of the execution state

Pros
- Flexibility
- Robustness
- Reliability

Cons
- Complex Implementation
- Interoperation
Introduction

Mobile Agents

Definition
- They have all the properties of agents and exhibit
  - communication model
  - security model
  - navigation model

Mobility Support
- FIPA guidelines...
  - Move operation
  - Transit state
  - Execute operation
- not available in all agent frameworks and always implemented among agent platform of the same type.
- Examples: Aglets, Jade, Sumatra, ... AgentService
Implementing Mobility in AgentService

Christian Vecchiola, Alberto Grosso, Antonio Boccalatte

Mobility AgentService

- We need to...
  - ... integrate it into a pre-existing infrastructure
  - ... provide an interface easy to use by the agents (... they just have to say.. “move me”)  
  - ... provide a reliable and secure service
  - ... provide an infrastructure implementation independent

- What we gain...
  - basic (optional) block on which build other services like...
    load balancing
  - short development time

Introduction | AgentService | Mobility | Load Balancing | Conclusions
Implementing Mobility in AgentService

Christian Vecchiola, Alberto Grosso, Antonio Boccalatte

11/18/2005

AgentService

Creation of the Application Domain

Agent Instance

AgentTemplate

template

Knowledge types and default instances

Behaviour types and default instances

Runtime stub

User request: create an agent of type X

All the assemblies needed to create agent defined by the template X are retrieved

Upload of assemblies containing:
- AgentTemplate type
- Knowledge and Behaviour types

Agent Template

AMS

DF

MTS

Registration

Introduction

AgentService

Mobility

Load Balancing

Conclusions
Implementing Mobility in AgentService

Chris Vecchiola, Alberto Grosso, Antonio Boccalatte

Introduction

AgentService

Mobility

Load Balancing

Conclusions
Mobility Requirements

- We need to...
  - stop the agent activity
  - persist the data constituting its state
  - create an instance of the same type on the target platform
  - restore its execution state

- We can observe that
  - moving an agent among platforms is similar to resuming it from a persistent storage
  - additional infos (sometimes maintained by modules) have to be moved too: i.e. conversations
**Mobility Services**

- **MobileAgent**
  - **AgentProfile**
    - Provides all the information to move an agent
      - AID
      - TypeDescriptors about knowledge and behaviour types
    - Used by the mobility service to setup the transfer of the agent
  - **TransportAgent**
    - Maintains all the data related to
      - Knowledge objects
      - Behaviour objects
      - Messages
Mobility Services

- Mobility Module
  - Core module
  - Controlled by AMSs of the source and target platforms
  - Defined by the `IMobilityModule` interface
  - Exposes the following operations
    - `MoveAgent(MobileAgent agent, PlatformDescription target);`
    - `ReceiveAgent(MobileAgent agent, PlatformDescription source);`
    - `[event] AgentMovedHandler AgentMoved;`
Implementing Mobility in AgentService

Christian Vecchiola, Alberto Grosso, Antonio Boccalatte

11/18/2005

Mobility Services

Interaction Protocol

- Move X to B
- AMS
- Agent X
- stop
- <transit>
- storage
- mobility
- move
- recover
- Platform A

- Can I move X?
- Yes you can
- AgentProfile
- Missing Assemblies
- Assemblies, TransportAgent
- <agent moved>
- Platform B

- AMS
- Mobility
- Storage
- receive
- lookup
- new
- resume
- Agent X
- original state

Introduction
AgentService
Mobility
Load Balancing
Conclusions
Default Implementation

- Uses the FTP protocol to move agents:
  - Acts as a server when receiving an agent
  - Acts as a client when moves an agent

- Some commands have been added to the protocol to simplify the transfer management

Filesystem structure

- ftp root
- profiles
- assemblies
  - platform 1
  - platform 2
Load Balancing Services

Mobility and Load Balancing

- Mobility is a fundamental requirement
- Load Balancing requires
  - The ability to move agents
  - The ability to detect overload conditions
  - The application of strategies to dispatch agents between nodes
- Can we do this in AgentService?
  - Mobility module
  - Rich event model
  - Load Balancing Module
Federations

- Platform join together in a *Federation*
- Federation
  - constitutes a single virtual platform
  - defines the boundaries into which load balancing policies are executed
  - dynamic structure (join)
- Master-Slave model
  - one platform node (master) controls the status of the entire virtual platform
  - slave nodes notifies the master about their status
  - master applies load balancing policies and instruct slaves to move agents
Load Balancing Module

- Uses the information about the status of the platform:
  - number of agents in the node
  - number of messages exchanged among agents
- Listens to the events originated by the platform:
  - creation / disposal of an agent
  - message sent / received
- Interacts with other modules to perform its activity
- Uses a plug-in model to implement custom policies
Inhibiting Mobility in AgentService

Christian Vecchiola, Alberto Grosso, Antonio Boccalatte

11/18/2005

Load Balancing Services

- **Interface ILoadBalancingPolicy (ILBP)**
  - Defines the services required by each load balancing strategy
  - Decouples the activity of the module from the strategy used to apply load balancing
  - Used by the master node of the federation

```csharp
interface ILBP
{
    PolicyDescription GetDescription();
    void ConsumeEvent(PlatformEvent event);
    void getNextAgentToMove(out AID aid, out PlatformDescription target);
    void AddProfile();
    void RemoveProfile();
    void UpdateProfile();
}
```
Load Balancing Services

Interaction Protocol

- .. an event occurs..
  - slave nodes send information about the event to the master node

- .. on the master node...
  - the master node forwards the event to the policy
  - if overload conditions are detected the module requires the next agent to move
  - the master node instruct slaves to move the agent

- .. on slaves / master ..
  - the module instructs the mobility service to perform the transfer
Conclusions

- **Mobility**
  - AgentService has been extended with an infrastructure to support mobile agents
  - The implementation of this infrastructure...
    - ...has been performed with low impact thanks to the modular structure of the platform
    - ...is completely transparent to programmers and agents

- **Load Balancing**
  - Requires Mobility Service
  - Simple implementation
Thanks for the Attention

http://www.lido.dist.unige.it/AgentService/