Outline

- motivation
- brief history of BioAgent
- Bioagent architecture
- Bioagent abstract model
- Preliminary results
- on-going work
Why Agent

• To support Bioscientists during the genome data analysis
  • Available data growth exponentially: many research institutes are publishing data on line

• To decentralize the coordination of local tasks processing that characterize a workflow of an experiment

• To allow the use of remote data, usually prohibitive for the complex interaction

• To allow the use of interesting experiments that might contain relevant information but may have been carried out in locations which have escaped notice

E. Merelli
Why Mobile

- Biological experimental workflow
  - not necessarily need to transfer all data
  - could only move code
  - could process its tasks by traveling over the network using remote data and services
  - ...

E. Merelli
Could be Intelligent?

- Intelligence in an agent concerns the ability to perceive the environment (context) where its mission must be achieved using a conceptual map of the workflow.
An **agent** is a program capable of acting exactingly in order to accomplish one or more tasks on behalf of its user.

A **mobile agent** is a program that can move from host to host in a network by keeping its state while travelling.

An **intelligent mobile agent** is a program that, perceiving the working context and performing its mission in a dynamic environment, can decide to move between nodes preserving its status.
Why another Mobile Agent System?

- Many mobile agent systems have been developed in both industry and research:
  - Aglets, D’Agents, Mole, Ara, NOMADS, ...

- We propose a new one starting from application point of view: Biological domain

- Biological Application requirements
  - simplicity
  - scalability
  - modularity
  - usability
History

- May 2001: Nettab 2001
- June: first prototype, we accept the challenge send to us by L. Toldo
- July: Bioagent has been announced at BOSC 2001 Copenhagen
- December: brainstorming on our idea of mobility
- January: 5 students are committed to the project

- July 2002: Nettab 2002 - first release of the system
Main choices

- All Java
- All XML + DTD + RDF + $XOL$
- All declarative

- Two Agent model
  - Service agent
  - User agent
Introduction to the system

- The BioAgent is a 4-layered architecture
  - Core Layer
    - Security management
    - Resource management
    - Mobility management
    - Agent management communication management
  - Service agent Layer
    - Broker agent
    - Web interface agent
    - Wrapper service
    - Ontology service
  - BioAgent Layer:
    - User agent
  - Workflow Layer
Core Layer

security manager intercepts any agent's request and verifies the caller's credentials for execution of the requested operation.

resource manager: provides all basic primitives with access stored data

mobility manager: supports agents migration

communication manager: allows hierarchical communication between agent parent and agent children. Communication is independent of the location where agents are active.

Bioagent manager: supports creation and cloning of agents
Service agent Layer

**Broker agent**: keeps track of each running service agent and provides any new agent with a list of locally accessible services

**Web service**: manages interaction between users and platform, by this service a user can directly interact with location services

**Wrapper**: provides access to local resources by abstracting from the nature, providing a uniform representation using XML

**Ontology service**: manages ontologies using integration tools.
BioAgent layer

BioAgent Layer is populated by user agents, it concerns with management of user agents.

The BioAgent layer consists of all those functions related to the “intelligence” agent planning, coordination, perception ...
Workflow Layer

In the BioAgente context:

- the term *workflow* means the coordinated execution of multiple tasks or activities. Thus a
- Biological experiment can be seen as a workflow

The Workflow Layer provides BioAgent-l a suitable language and tools suitable for defining and managing workflow
System architecture

BioScientists / Applications

Workflow Definition

BioAgents

Service Agents

Core
- Resources Mng
- Security Mng
- Communication Mng
- Mobility Mng
- Agent Mng

Ontology

Service

Broker

Agent

Web Interface

Wrappers

User Agent

E. Merelli
Distributed system

EMBL

NCBI

Expasy

Private Research Institute

E. Merelli
Service agents directly access the local resources.
User agents access the local resources by interacting with service agents i.e. wrappers.
Messaging

System

Cloning

permit
denied
Communication Model

Pool coordinated by A

Pool coordinated by B

Pool coordinated by C

Pool coordinated by D

E. Merelli
Agent Hierarchy

Abstract

ServiceAgent

UserAgent

Agent

Istance

E. Merelli
Cloning

Cloning and Moving

Clone()  Go(port, method)

E. Merelli
Protocol between UserAgent and ServiceAgent

- User knows protocol from service
- User uses instance for query
- User uses only instance DOM for all service

$\text{XML}_{\text{in}}$

$\text{XML}_{\text{out}}$

E. Merelli
Implementation Hints

- BioAgent is a platform completely developed in Java based on simplicity, modularity and ease-of-handling.
- Simplicity is provided by:
  - The unique abstract class *Agent* which assigns the basic features to each agent.
  - There are two usable extensions of *agent*: *UserAgent* and *ServiceAgent*.
  - The security manager defines constraints for each instance of *ServiceAgent* and *UserAgent*.
- Modularity is based on:
  - Services are plug-in.
  - Platforms interaction via *serialization* – *reflection* not RMI.
- Ease-of-handling:
  - Web interface for common user agent.

E. Merelli
BioAgent abstract model

- **Ontological level – domain**
  - It models the application domain using Common ontology
    - \( \rightarrow \) workflow (conceptual map)

- **Environmental level – context**
  - It models the environment in which the agent is going to work using
    - \( \rightarrow \) user and service agent
Preliminary model v.1
On-going works

- Complete the specification and development of the two architectural highest layers functionalities and to prove the system in a complex real applications.
- Compare efficiency and effectiveness of our mobile system with those not mobile but proposed to support automated genomic annotation problem
- Development of Darwin service to support BioAgent –/l
- Development EnsEmbl,
- Development of BioAgent- l compile
- Try to become FIPA compliant
Acknowledgements

many thanks to Luca Toldo
for useful discussions
Many thanks to students

Diego Bonura
Ezio Bartocci
Davide Lorenzoli
Francesca Piersigilli
Gloria Rossi