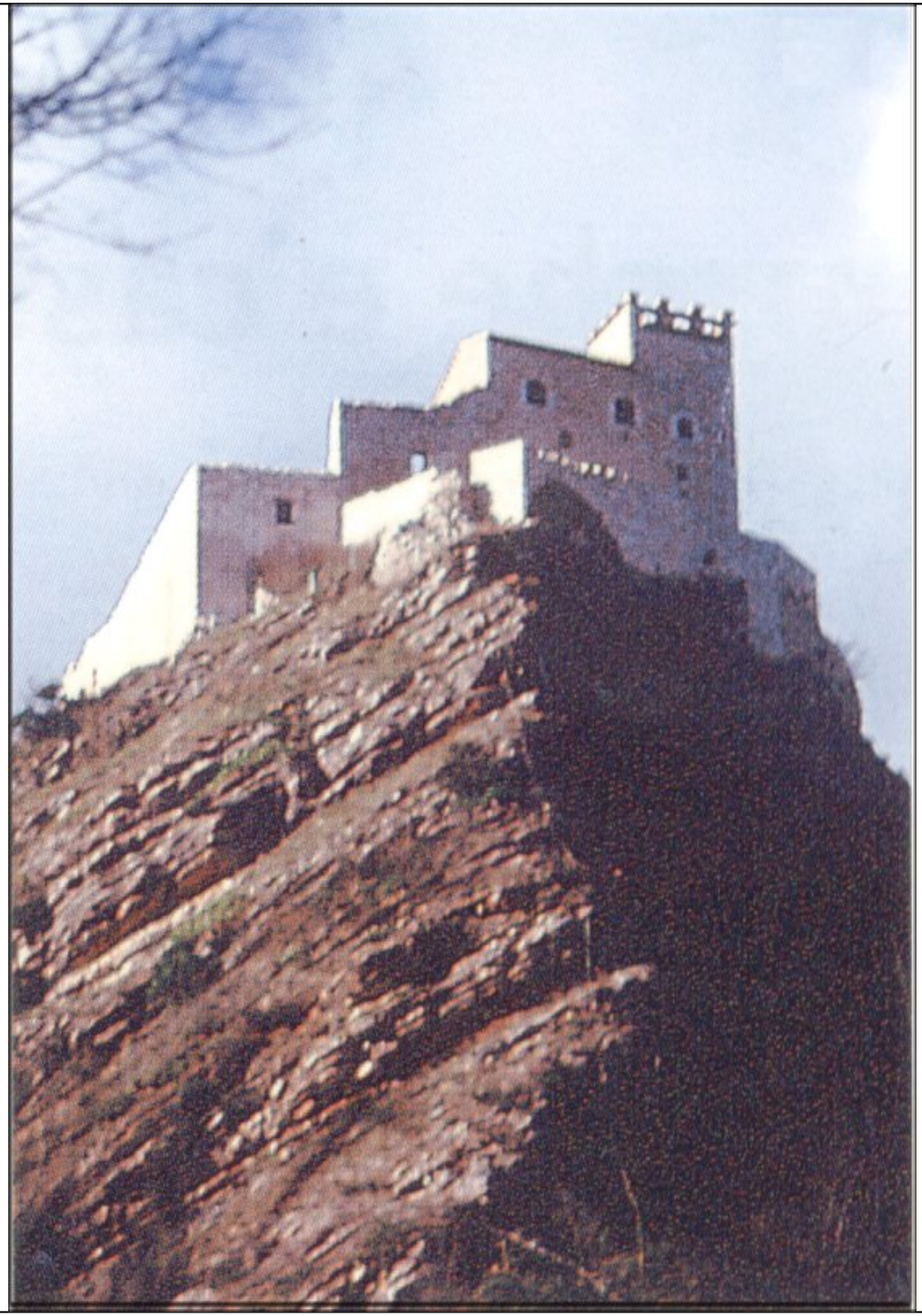
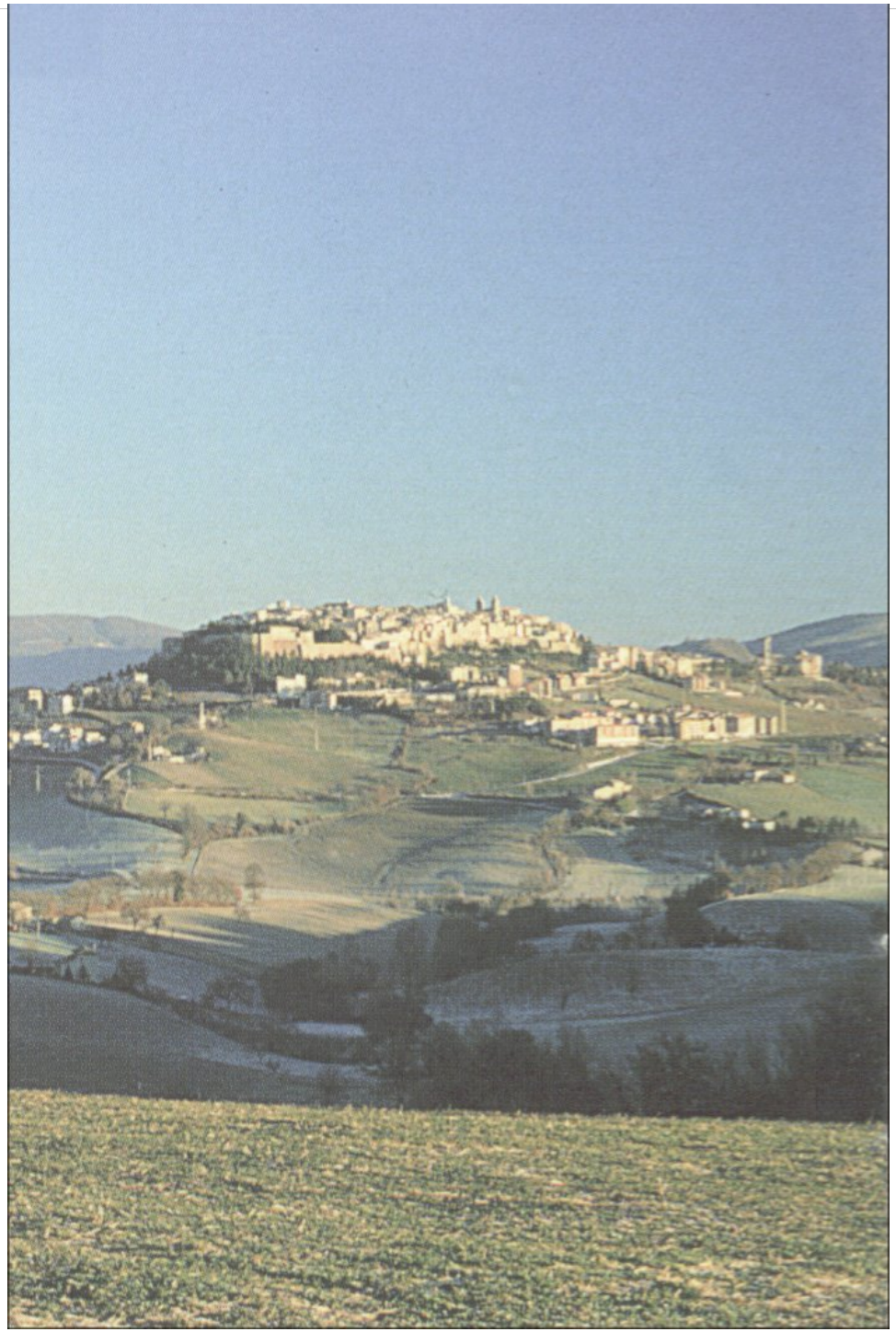




# CAMERINO

Monumento in Piazza

# CAMERINO

San Venanzio

# CAMERINO

Teatro


BUS

Pinacoteca e Museo civici, *Annunciazione*

Pinacoteca e Museo civici,
*Santa Caterina*

Duomo, *Madonna della Misericordia*, 1450

Teatro Filippo Marchetti, 1856


# Agents in Bioinformatics

**Emanuela Merelli**
**Dipartimento di Matematica e Informatica**
**Università di Camerino**
**Italia**

**emanuela.merelli@unicam.it**

**Stanford 2 February 2005**

University of Camerino


A14
AUTOSTRADE
E SUPERSTRADE
ALTRE STRADE
FERROVIE
CAMERINO
SEDI UNICAM
ANCONA
FABRIANO
PERUGIA
MATELICA
CASTELRAIMONDO
CIVITANOVA M.
FIRENZE
ROMA
FOLIGNO
CAMERINO
S.BENEDETTO d.T
ASCOLI
A14

# Agents in Bioinformatics

AgentLink III -- http://www. agentlink.org
Technical Forum Group on Agents in Bioinformatics
(http://www.bioagent.net/TFG-bioagents)
1st meeting in Rome July 2004
Next in Camerino September 2005

Workshop on Agents in Bioinformatics:
1st 2001 in Bologna - http://www.nettab.org/2001
2nd 2004 in Camerino – http://www.nettab.org/2004
Next
Workshop 25 July in Utrecht- Neatherland - http://www.diee.unica.it/biomed05

# Agents in Bioinformatics

Agents may be useful for applications that imply: repetitive and time-consuming activities; knowledge management, such as integration of different knowledge sources; and modelling of complex, dynamic systems.

- Genomic analysis
  - Geneweaver UK (Bryson et al., 2001)
  - DECAF in the US (Decker et al., 2001)
- Systems biology
  - intracellular signalling pathways (Cellulat in Gonzalez, 2003)
  - cell modelling (CellAK in Webb and White, 2004)

# Outline

- What is an agent and a multi-agent system?
- The Bioagent project
- What we mean for agents in bioinformatics?

- The activity-base programmin environment
- The Hermes middleware for mobile computing
- One case study
  - "Oxidation of carbohydrate" simulation process

- What are the interesting BioMedical domains for agents?

# What is an agent?

- A computer system capable of flexible, autonomous (problem-solving) action, situated in dynamic, unpredictable and typically multi-agent environment.

# What is an agent?

- A computer system capable of flexible, autonomous (problem-solving) action, situated in dynamic,, unpredictable and typically multi-agent environment.

- control over internal state and over own behaviour

# What is an agent?

- A computer system capable of flexible, autonomous (problem-solving) action, situated in dynamic, unpredictable and typically multi-agent environment.

- experiences environment through sensors and acts through effectors

# What is an agent?

- A computer system capable of flexible, autonomous (problem-solving) action, situated in dynamic, unpredictable and typically multi-agent enviornment.

- reactive: respond in timely fashion to environmental change
- proactive: act in anticipation of future goals

# Our interest is on mobile agent computation

An agent is a computer system capable of acting in order to accomplish tasks on behalf of its user

A mobile agent is a computer system that can move from one execution environmnet to another, if in a network (or virtual distributed space) from host to host by keeping its execution state while travelling

A proactive agent is computer system able to achieve a goal by performing actions in anticipation and reacting to events in the dynamic execution environment

In our context, a proactive mobile agent is a computer system that can move between execution environments by preserving the status and managing remote information by using its knowledge base so to perform its goal

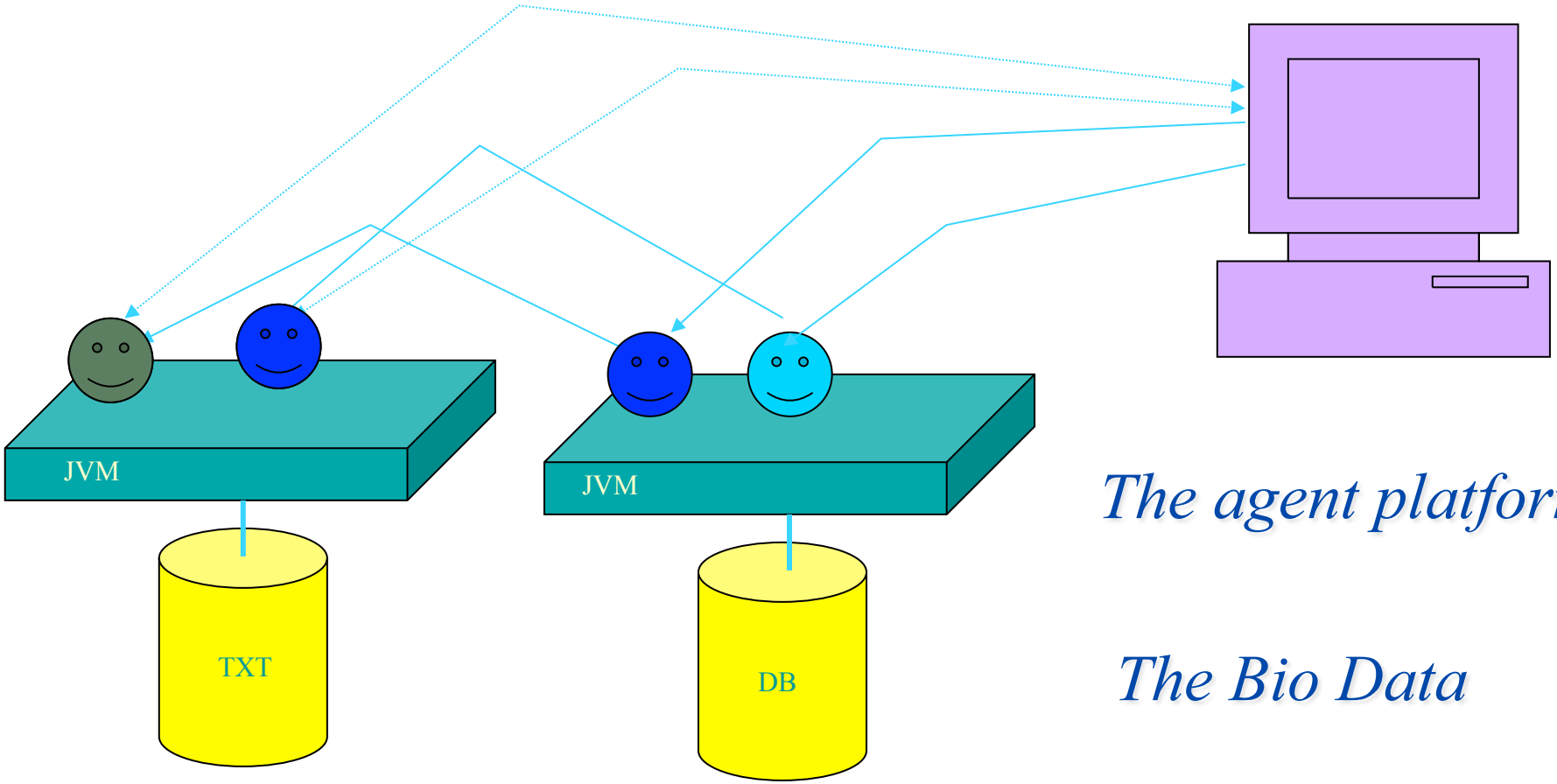
Any agent is characterized by the following features

*Moving, Communication, Cloning, State-preserving, Knowledge-management capability*

# How it works …

Single agent

*The People*




*The agent platform*

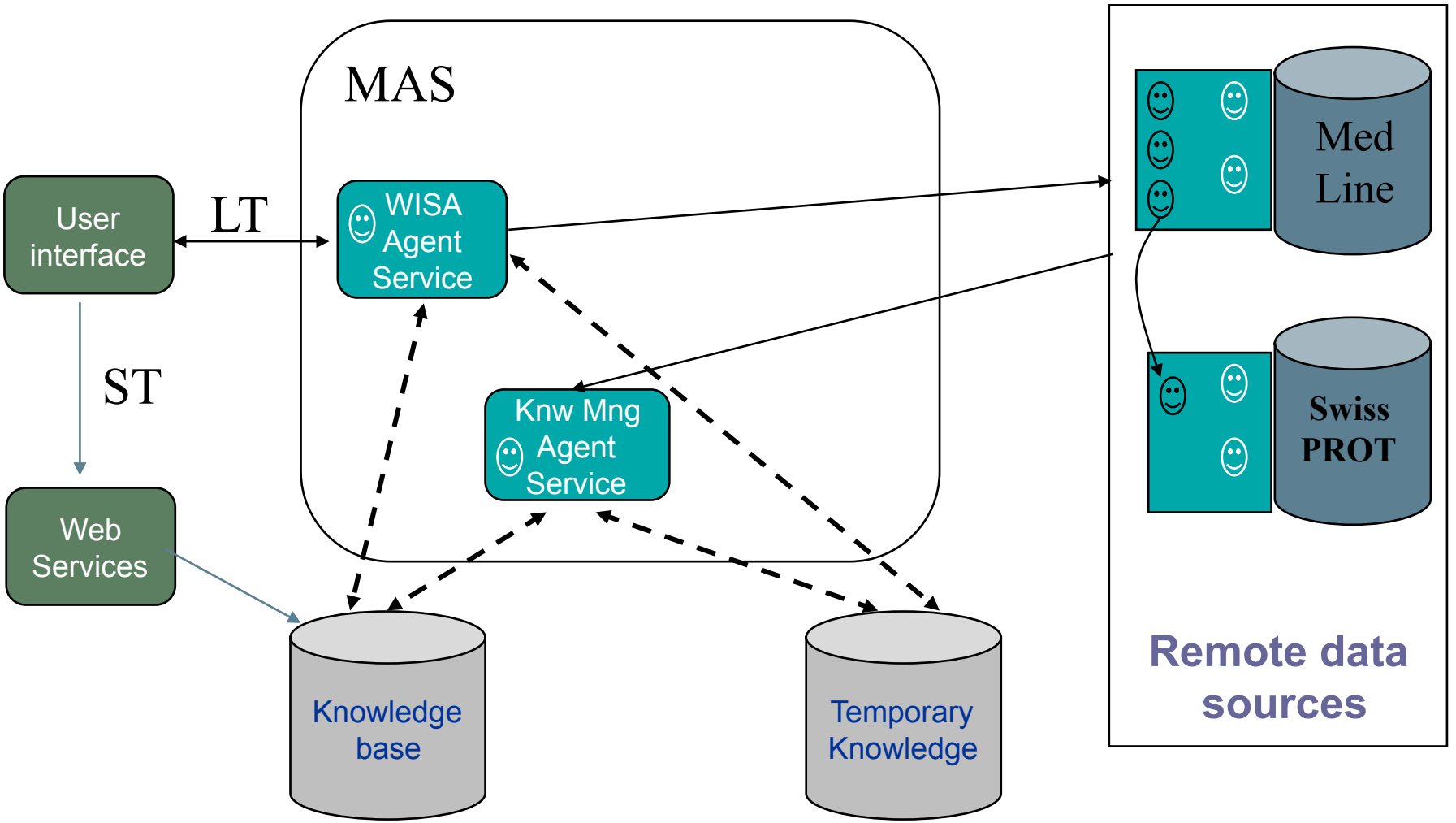
*The Bio Data*

# Problems and issues

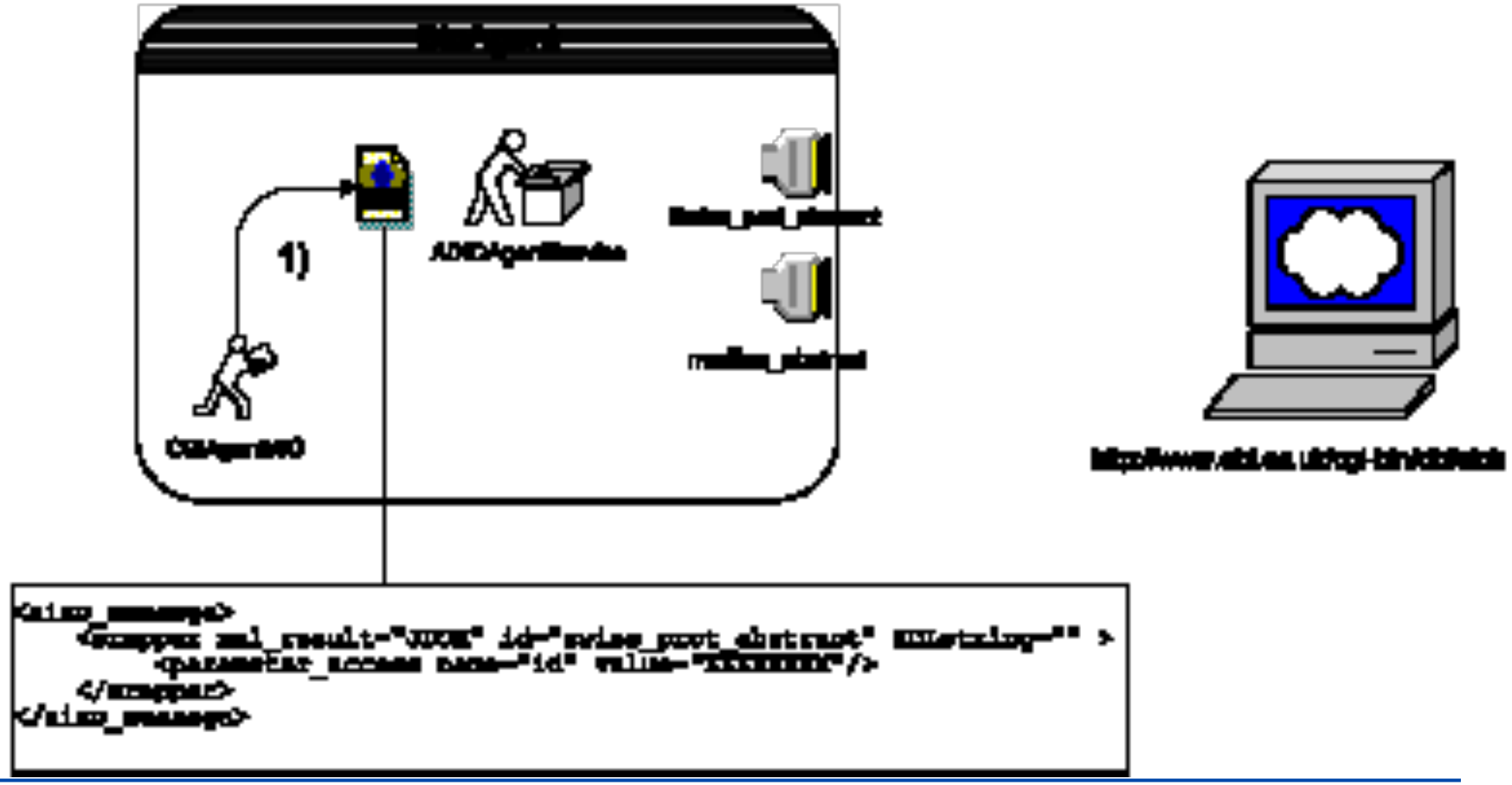
- Agent has to
  - speak multiple languages
  - Overcome firewalls
  - Be easily programmable

- Platform has to be
  - Flexible
  - Expandable
  - Modular
  - Secure and
  - it has to support mobiliby

# BioOnco --- Oncology over internet

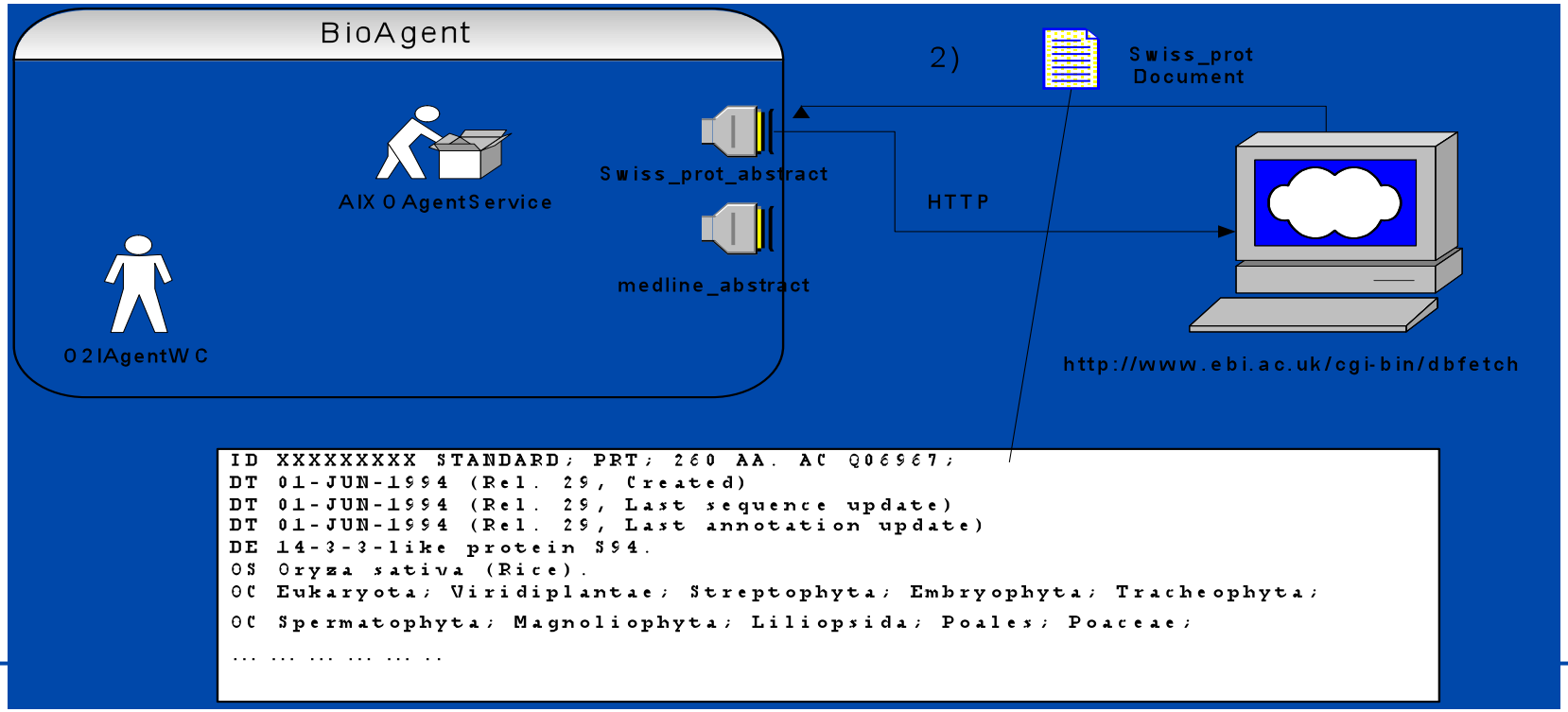


# Esempio - step1

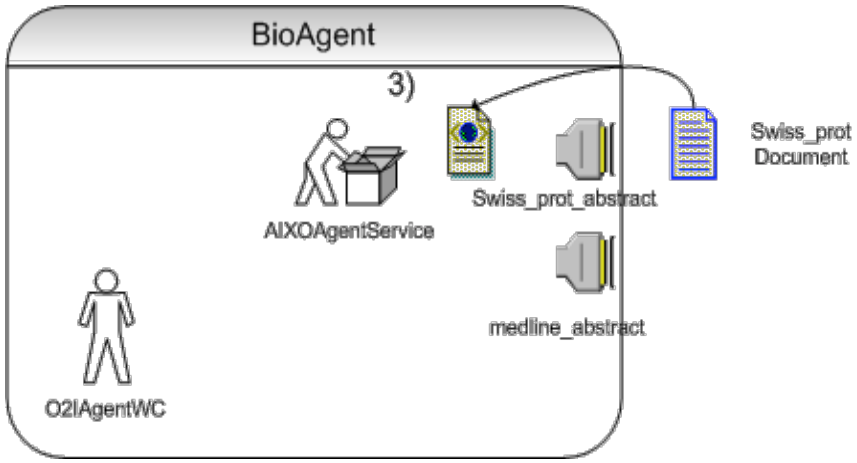


# step 2

| Proteina | Computer | KBytes | Rete | http access time(sec.) |
|---|---|---|---|---|
| **P53_HUMAN** | **MyNotebook** | **49.6** | **Rete Ateneo** | **4.89** |
| **FOS_HUMAN** | **MyNotebook** | **4.72** | **Rete Ateneo** | **3.62** |
| **1433_ORYSA** | **MyNotebook** | **2.51** | **ADSL 640 Kbits/s** | **1.76** |
| **P53_HUMAN** | **Giasone** | **49.6** | **Rete Ateneo** | **2.37** |
| **FOS_HUMAN** | **Giasone** | **4.72** | **Rete Ateneo** | **1.63** |
| **1433_ORYSA** | **Giasone** | **2.51** | **Rete Ateneo** | **2.36** |

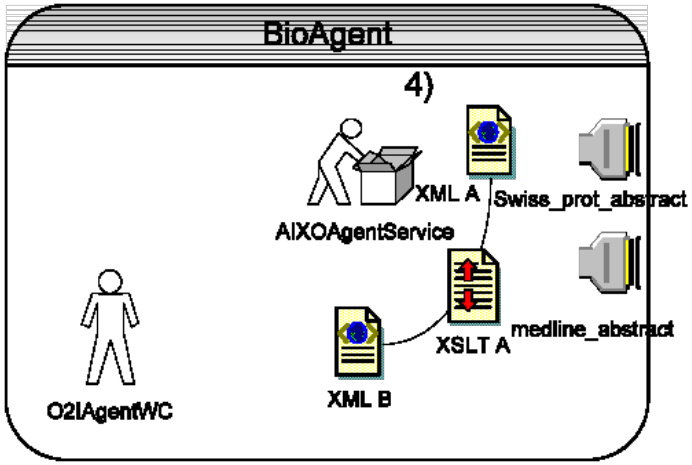


# step 3



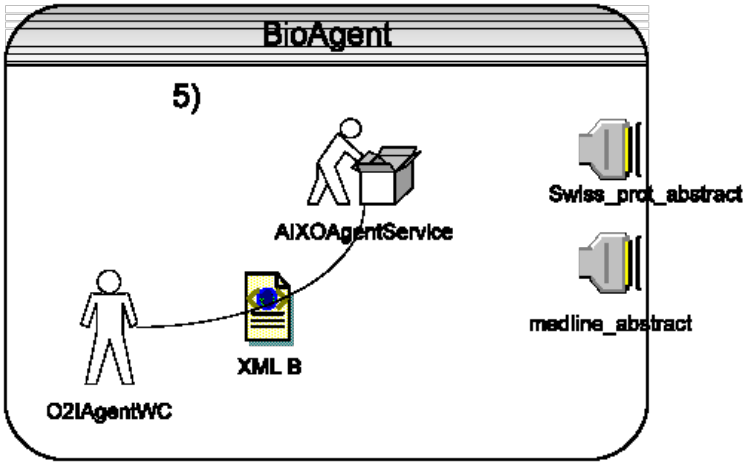

| Proteina | Computer | KBytes | XMLTranslation time (sec.) |
|---|---|---|---|
| **P53_HUMAN** | **MyNotebook** | **49.6** | **1.21** |
| **FOS_HUMAN** | **MyNotebook** | **4.72** | **0.11** |
| **1433_ORYSA** | **MyNotebook** | **2.51** | **0.06** |
| **P53_HUMAN** | **Giasone** | **49.6** | **1.11** |
| **FOS_HUMAN** | **Giasone** | **4.72** | **0.04** |
| **1433_ORYSA** | **Giasone** | **2.51** | **0.36** |

# step 4



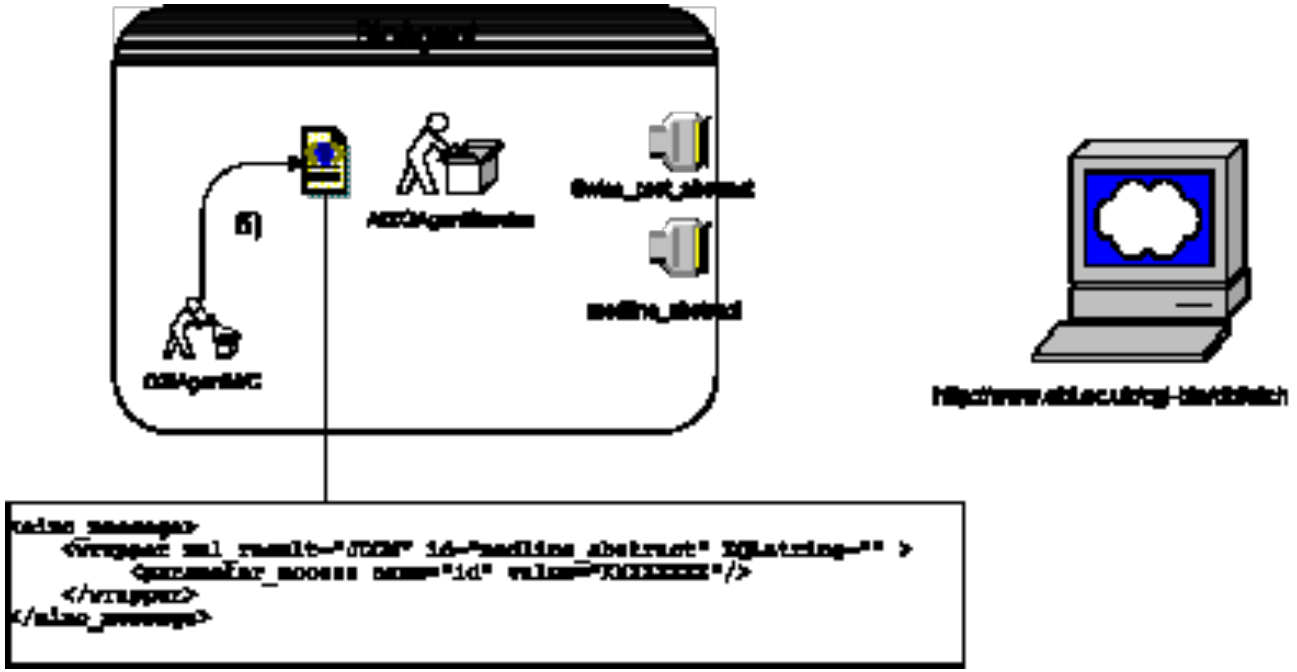

| Proteina | Computer | XSLT time (sec.) |
|---|---|---|
| **P53_HUMAN** | **MyNotebook** | **1.42** |
| **FOS_HUMAN** | **MyNotebook** | **1.1** |
| **1433_ORYSA** | **MyNotebook** | **0.6** |
| **P53_HUMAN** | **Giasone** | **0.7** |
| **FOS_HUMAN** | **Giasone** | **0.48** |
| **1433_ORYSA** | **Giasone** | **0.428** |

# step 5




| Proteina | Computer | Message go and go back time (sec.) |
|---|---|---|
| **P53_HUMAN** | **MyNotebook** | **1.87** |
| **FOS_HUMAN** | **MyNotebook** | **3.29** |
| **1433_ORYSA** | **MyNotebook** | **1.09** |
| **P53_HUMAN** | **Giasone** | **0.8** |
| **FOS_HUMAN** | **Giasone** | **0.9** |
| **1433_ORYSA** | **Giasone** | **0.68** |

# step 6



http://ensembl.unicam.it:8080/bioagent/

# Intermediate Result

```
<?xml version="1.0" standalone="no" ?>
- <SwissProt_entries>
- <SwissProt_Entry>
  <ID entry_name="P53_HUMAN" data_class="STANDARD" molecule_type="PRT" sequence_lenght="393" />
  <AC number="P04637" />
  <AC number="Q16848" />
  <AC number="Q9UBI2" />
  <DT date="13-AUG-1987" release="05" type="Created" />
  <DT date="01-MAR-1989" release="10" type="Last sequence update" />
  <DT date="28-FEB-2003" release="41" type="Last annotation update" />
  <DE>Cellular tumor antigen p53 (Tumor suppressor p53) (Phosphoprotein p53) (Antigen NY-CO-13).</DE>
  <GN>TP53 OR P53.</GN>
  <OS>Homo sapiens (Human).</OS>
  <OC>Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.</OC>
  <OX>NCBI_TaxID=9606;</OX>
- <RN num="1">
  <RP>SEQUENCE FROM N.A.</RP>
  <RX Bibliographic_db="MEDLINE" id="85230577" />
  <RX Bibliographic_db="PUBMED" id="4006916" />
  <RA>Zakut-Houri R., Bienz-Tadmor B., Givol D., Oren M.;</RA>
  <RT>"Human p53 cellular tumor antigen: cDNA sequence and expression in COS cells.";</RT>
  <RL>EMBO J. 4:1251-1255(1985).</RL>
  </RN>
- <RN num="2">
  <RP>SEQUENCE FROM N.A.</RP>
  <RX Bibliographic_db="MEDLINE" id="87064416" />
  <RX Bibliographic_db="PUBMED" id="2946935" />
  <RA>Lamb P., Crawford L.;</RA>
  <RT>"Characterization of the human p53 gene.";</RT>
  <RL>Mol. Cell. Biol. 6:1379-1385(1986).</RL>
  </RN>
…
```

# Final Result

- <?xml version="1.0" encoding="UTF-8" ?>
- - <Abstracts>
- - <MedlineCitation>
- <MedlineID>85230577</MedlineID>
- <AbstractText>A 2.5-kb cDNA clone for human p53 tumor antigen has been isolated. This clone contains the entire coding region including 135 bp upstream of the first ATG. Comparison of the nucleotide sequence of human p53 and mouse p53 demonstrates that the first ATG in human p53 corresponds to the second ATG (codon No. 4) in mouse p53. The human p53 comprises 393 residues and is longer than the mouse p53 due to six additional codons present at the region corresponding to exon 4 of the mouse p53 gene. The DNA sequence homology between the coding regions of mouse and human p53 is 81% and the conservation of homology is not equally distributed along the molecule. When inserted into SV40-based expression vectors the human p53 cDNA successfully directs the production of a polypeptide with an apparent mol. wt. of 55 kd which can be precipitated by monoclonal antibodies to p53.</AbstractText>
- </MedlineCitation>
- - <MedlineCitation>
- <MedlineID>87064416</MedlineID>
- <AbstractText>Cosmid and lambda clones containing the human p53 gene were isolated and characterized in detail. The gene is 20 kilobases (kb) long and has 11 exons, the first and second exons being separated by an intron of 10 kb. Restriction fragments upstream of sequences known to be within the first identified exon were tested for promoter activity by cloning them in front of the chloramphenicol acetyltransferase gene and transfecting the resulting constructs into HeLa cells. A 0.35-kb DNA fragment was identified that had promoter activity. Results of primer extension experiments indicated that the mRNA cap site falls within this fragment, as expected. Analysis of the sequence upstream of the presumptive cap site indicated that the human p53 promoter may be of an unusual type.</AbstractText>
- </MedlineCitation>
- - <MedlineCitation>
- <MedlineID>85267676</MedlineID>
- <AbstractText>Three clones for the human tumor antigen p53 were isolated from a cDNA library prepared from A431 cells. One of these clones, pR4-2, contains the entire coding region for human p53. This clone directs the synthesis of a polypeptide with the correct molecular weight and immunological epitopes of an authentic p53 molecule in an in vitro transcription-translation reaction. Although the pR4-2 clone contains the coding region for p53, it is not a full-length copy of the human p53 mRNA. Northern analysis showed that the p53 mRNA is approximately 2,500 nucleotides long, whereas the pR4-2 insert is only 1,760 base pairs in length. Analysis of the DNA sequence of this clone suggests that the human p53 polypeptide has 393 amino acids. We compared the predicted amino acid sequence of the pR4-2 clone with similar clones for the mouse p53 and found long regions of amino acid homology between these two molecules.</AbstractText>
- </MedlineCitation>
- …

AIXO: Any Input XML Output
E. Bartocci, L. Mariani, 2003

Download:
http://www.bioagent.net

# What is a Multiple Agent System (MAS)

In most cases, single agent is insufficient

- multiple agents are the norm, to model complex real systems (business, manufacturing, telecommuniation, biology..
  - natural decentralisation
  - multiple loci of control
  - multiple perspectives
  - competing interests

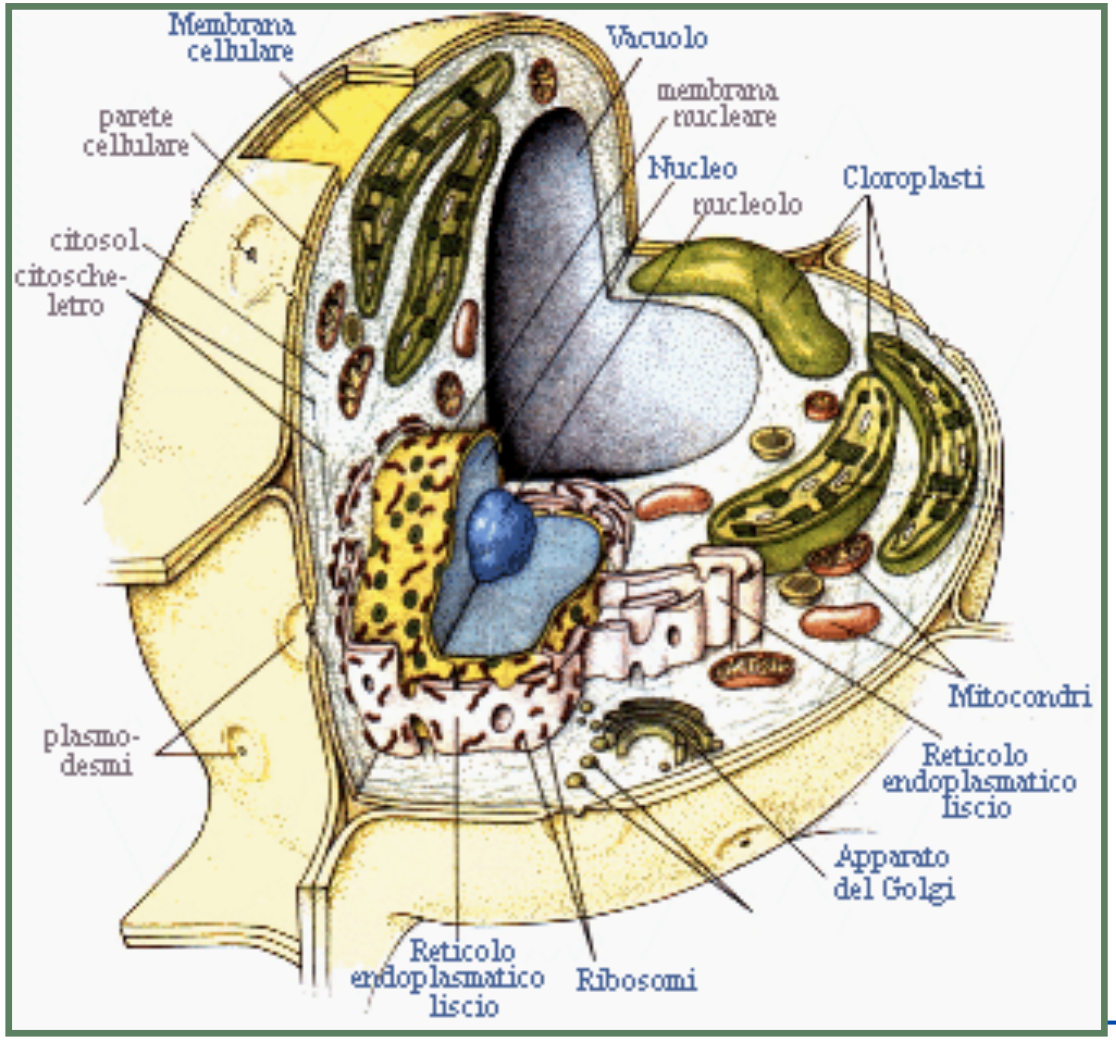
- Multi-agent system asks for coordination models

# Agents … what for?

To support next generation computing through facilitating agent technologies

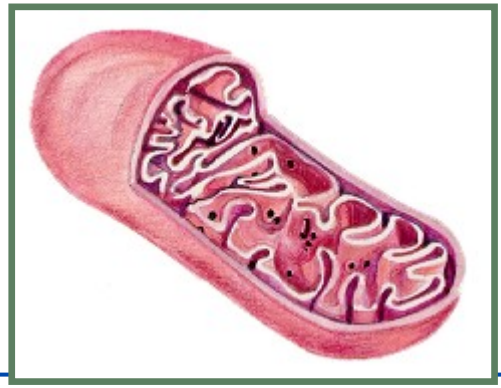
- As a metaphor for the design of complex, distributed computational systems
- As a source of technologies (e.g. mobile agents)
- As simulation models of complex real-world systems, such as in biology and economics

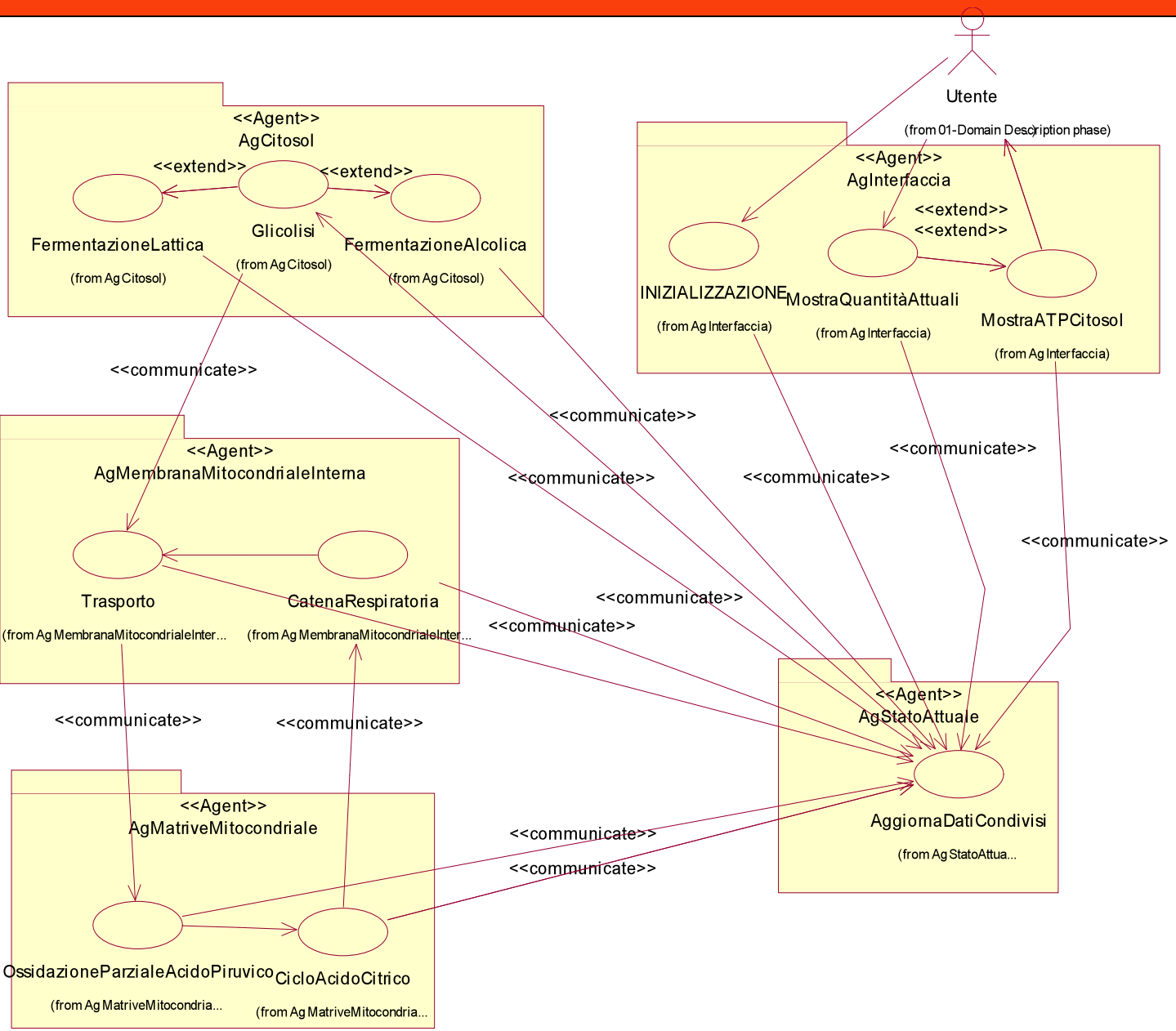
# La Struttura Cellulare



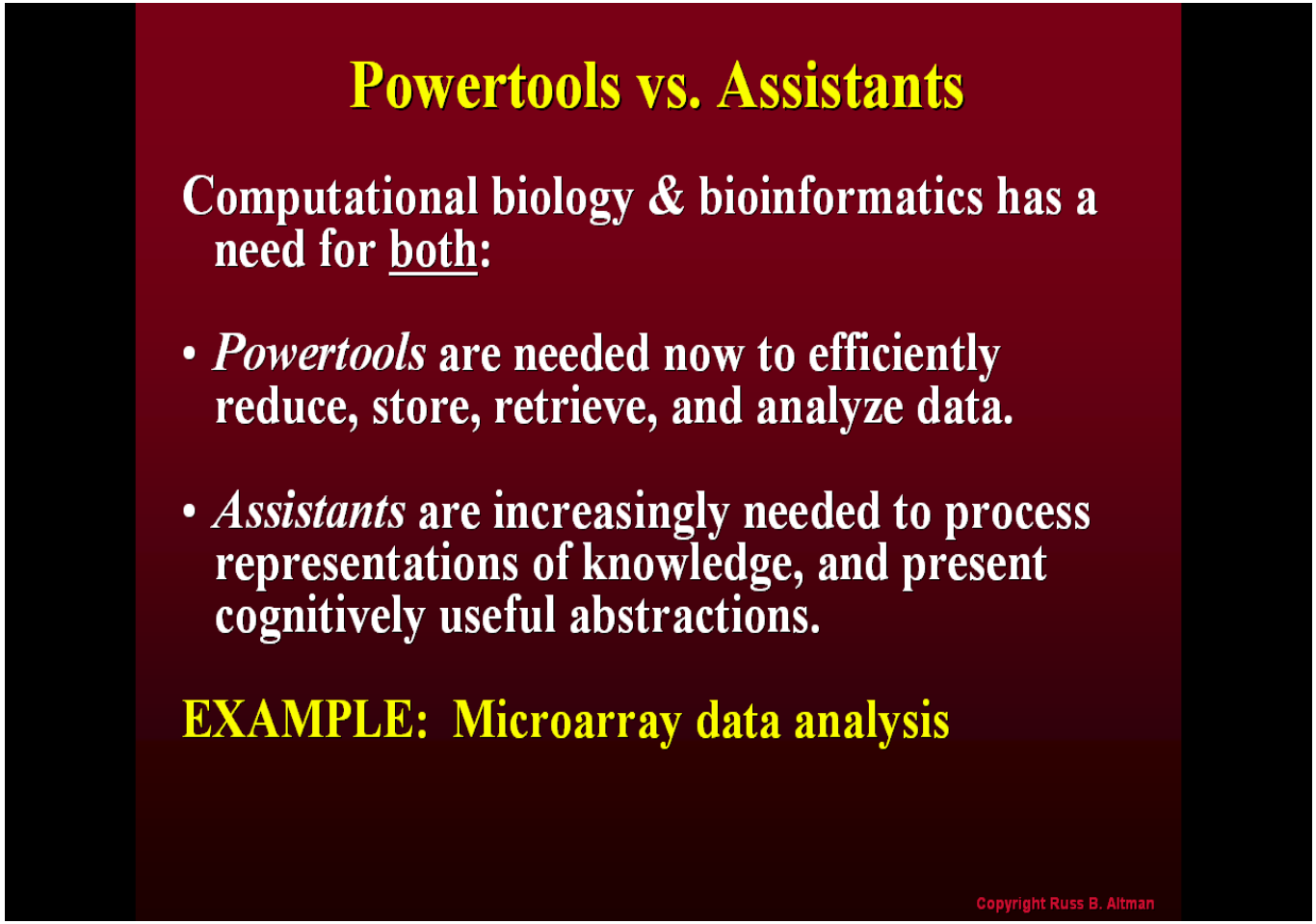

## Componenti Coinvolte

- **Citoplasma**
  - **Mitocondrio**
    - **Matrice Mitocondriale**
    - **Membrana Mitocondriale Interna**




Utente
(from 01-Domain Description phase)
<<Agent>>
AgCitosol
<<extend>>
<<extend>>
FermentazioneLattica
(from AgCitosol)
Glicolisi
(from AgCitosol)
FermentazioneAlcolica
(from AgCitosol)
<<Agent>>
AgInterfaccia
<<extend>>
<<extend>>
INIZIALIZZAZIONE
(from AgInterfaccia)
MostraQuantitàAttuali
(from AgInterfaccia)
MostraATPCitosol
(from AgInterfaccia)
<<communicate>>
<<communicate>>
<<communicate>>
<<communicate>>
<<communicate>>
<<Agent>>
AgMembranaMitocondrialeInterna
Trasporto
(from AgMembranaMitocondrialeInter...
CatenaRespiratoria
(from AgMembranaMitocondrialeInter...
<<communicate>>
<<communicate>>
<<communicate>>
<<communicate>>
<<Agent>>
AgStatoAttuale
AggiornaDatiCondivisi
(from AgStatoAttua...
<<Agent>>
AgMatriveMitocondriale
<<communicate>>
<<communicate>>
OssidazioneParzialeAcidoPiruvico
(from AgMatriveMitocondria...
CicloAcidoCitrico
(from AgMatriveMitocondria...

# Agents in Bioinformatics, what for?




2001

# Bioagent: a mobile agent platform for Computational Analysis of Biological Data

# Motivations (2001)


λ Electronic diagnostic tool = diagnosis

- Significant amount of data (ORFs, Experiments, …) disseminated and duplicated in a myriad of different dbs and repositories
- Different access modes
- Several actors (researchers, physicians, lab-technicians …)
- …

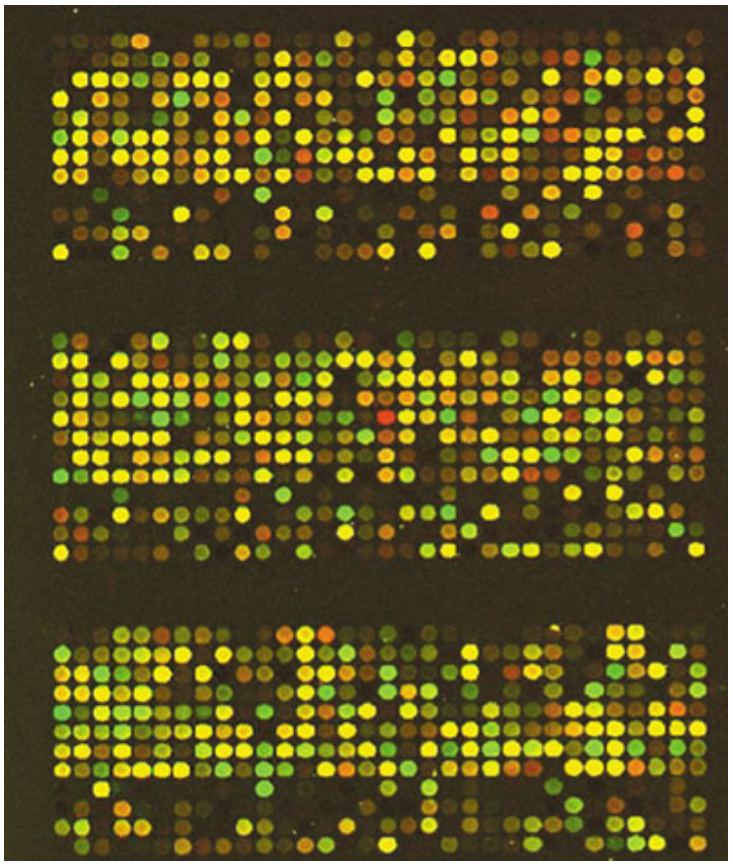
λ Our-goal

- define a general platform for mobile computing Bioagent (*Intelligent* mobile agent platform to process biological data) to support "genetic data analysis"
- define a declarative language BioAL to specify agents


# Application: analysis of gene expression

# Scenario: Microarray experiments

"...normally an experiment should include a set of hybridisations which are inter-related and performed in a limited period of time."



*MIAME (Minimal Information About Microarray Experiments) document by MGED*

*(Microarray Gene Expression Database group, UK)*

- Each hybridisation is constituted by a collection of experimental data (spots) usually one spot for each ORF (Open Reading Frame).
- The intensity of each spot quantifies the expression of the related ORF under the chosen experimental conditions

# MicroArray Markup Languages

## MAML "DTD" structure

proposed by the European Molecular Biology Lab (EMBL) and the European Bioinformatic Institute (EBI) and recently submitted to OMG (Object Management Group)

- Experimental design: the set of the hybridization experiments as a whole;
- Array design: each used array and each element (spot) on the array;
- Samples: used samples, the extract preparation and labeling;
- Hybridizations: procedures and parameters;
- Measurements: **images**, quantitation, specifications;
- Controls: types, values, specifications.

# Major goals pursued during the analysis of hybridization data: Clustering

- Clustering using experiments euc. distance
  - λ (ORF "guilty-by-association")
  - λ Present use: discover gene function
- Clustering using ORFs euc. distance
  - λ (transcriptional fingerprint)
  - λ Future use: diagnostic tool

# The BioAgent model

Data: Biological data related to an "organism"

- set of inter-related hybridisations
- Database source (ExpressDB Aach et al. 2000)
  - 147 distinct hybridisation experiments
  - 6053 ORFs

Goal: analysis of hybridisation data

- Clustering by experiments
- Clustering by ORF

Instruments: Algoritm and methods:

- Kohonen self-organizing map
  - Alborithm in C
  - 9 clusters
  - Convergens in 1000 cycles

Basic Knowledge: MAML "DTD"

Exteded Knowledge: Kohonen map


# Preliminary results

[M. Angeletti, R. Culmone, 2001]

# Demo for single agent platform

- We simulate four different sites by four DOS windows. Three will be the place with the data repository and the fourth will be the starting and ending point for an agent

- Open 4 DOS windows
- For each window, in C:\bioagent run setenv
- I window: in C:\bioagent\esempi\place run

  java exResourcePlace1
- II window: in C:\bioagent\esempi\place run

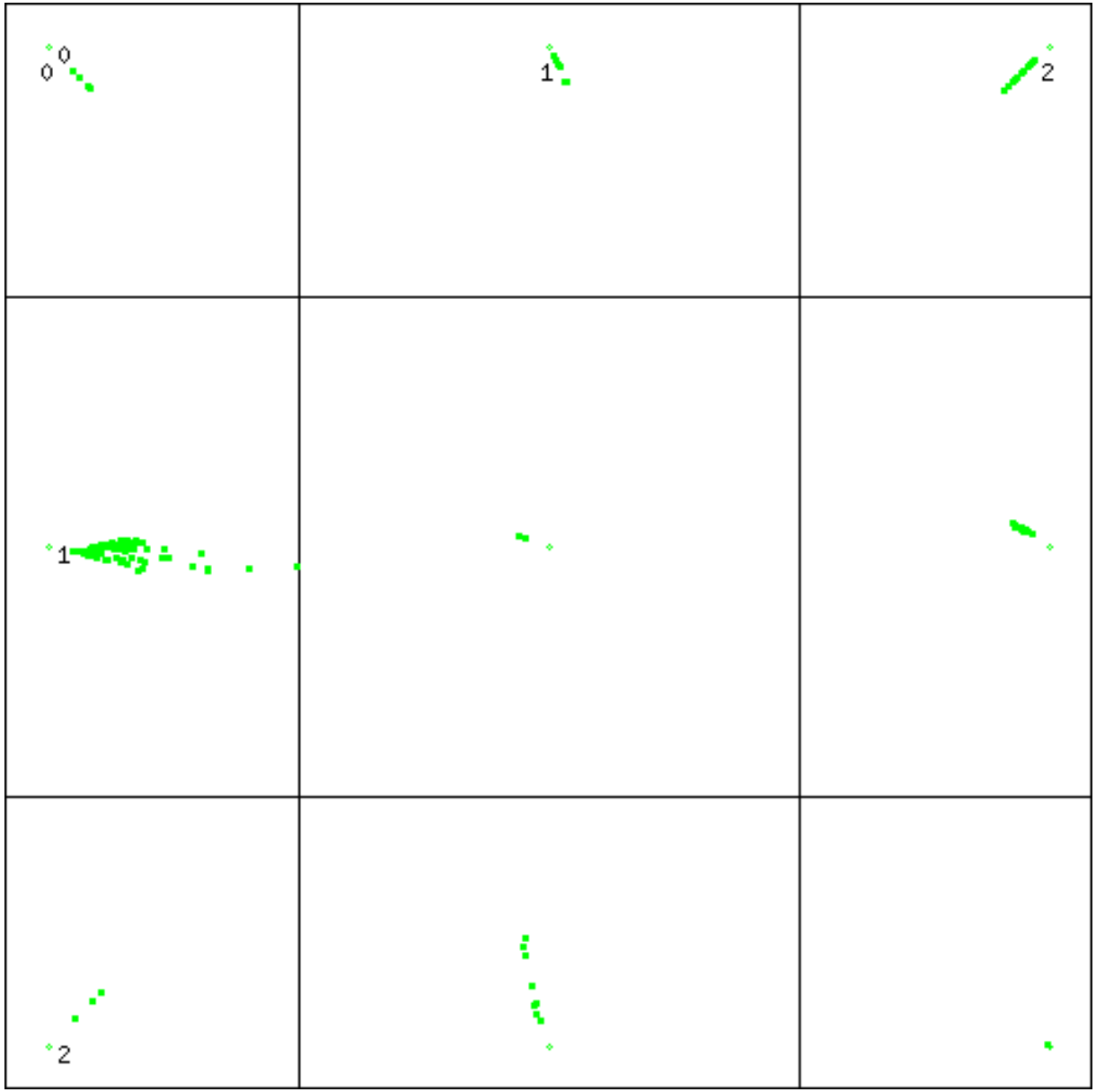
  java exResourcePlace2
- III window: in C:\bioagent\esempi\place run

  java exResourcePlace3
- IV window: in C:\bioagent\esempi\place run

  java RunAgent localhost
- The result is in C:\hermes\esempi\risultato.res

# The output

## Kohonen map




10000


# Back to agent programming environment

# Who will program agents?

- bioinformatics,
- biochemical researchers,
- physicians,
- biologists
- lab-technicians,
- …

are they able to program in Java?
must they learn it?
What kind of applications would they like to automize?

What kind of agent-oriented language?

# BioAL: Agents specification language

## Agent's Activities

- Bio data search
- Bio data integration
- Bio data clustering
- Bio data extraction
- Bio pattern recognition
- Bio knowledge discovering
- Bio data prediction
- …

## Agent's Instruments

- Optimization
  - Combinatorial algorithms
  - Heuristics algorithms
  - CLP: Constraints Logic Programming
  - …
- Classification
  - Neural nets
  - Kohonen self-organizing map
  - …
- Knowledge Discovering
  - Data mining
  - Multidimensional analysis
  -

## Agent Knowledge

- Basic knowledge (behaviours description; cömmon, local, extended Ontologies)

# Bioagent and BioAL (still in 2001)

BioAL must be "easy to use"

- could it be a declarative language?
- could it be hide to the end user?
- could it be ontology based?

Example: how to specify a new environment

let <envname>

be *VirtualEnv*

with *Instruments, Servicies, Events, Objects, Languges …*



. . .

Example: how to specify a new agent in a certain environment

create <agent> like <tipo> in <environment>

with

goals (m1,m2 …mn)

rules (r1,r2, … rn)

contraints (c1,c2, … ck)

instruments (s1,s2, … sn)

ontologies (o1,o2, … on)

agent-status (s1,s2, … sn)

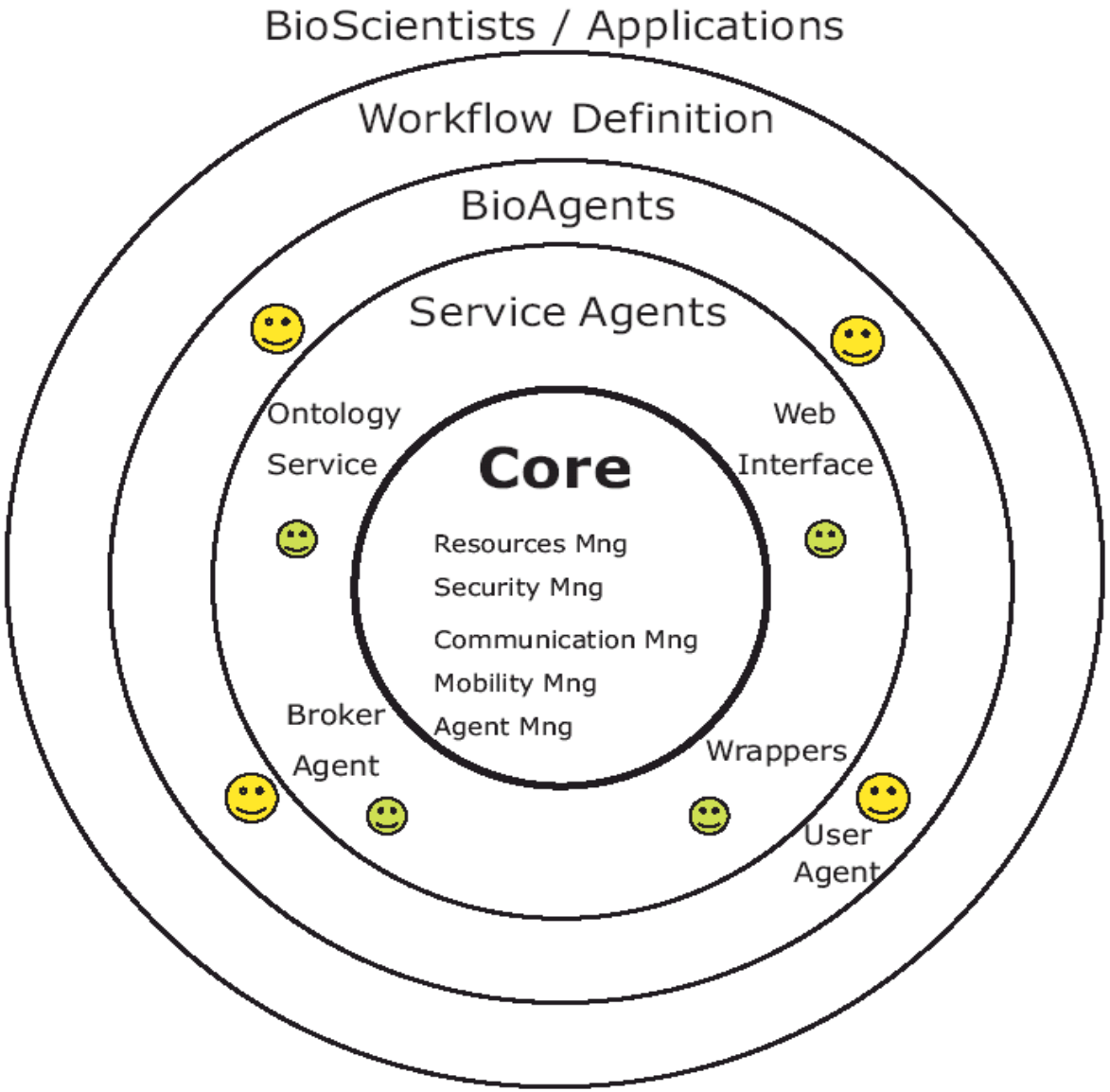
# What requirements the mobile agent platform has to satify?

- Agent execution
  - Agent creation, execution, deletion, execution control, task control, agent cloning,
- Agent mobility
  - Remote execution, migration, naming, addressing, location, domains
- Agent communication
  - Inter-agent, user/agent, naming, addressing, location of peers
- Agent and agent system security
  - System protection (agent = virus?), agent communication

# Main Choices: Java and ontology


- Hardware independent (runs on Windows, Unix, MacOS, PDA, …)
- Object oriented, agent oriented and activity workflow oriented
- Ubiquitous (Java does everything, everyone does Java)
- Network.centric
  - Code mobility (applets, servlets, class loading, serialisability
  - Network communication (TCP/UDP sockets)
  - Internet-ready (URL classes)
  - Security (secure sockets, secure manager)

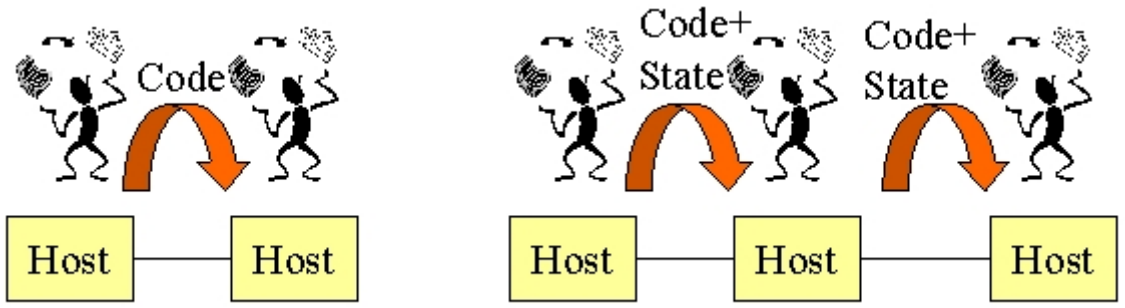
# Bioagent System architecture



# Mobile agent

A mobile agent can move from site to site in a network and keeps its execution state while traveling

- Agent while hoping from site to site over is bring a *backpack* with
  - Suitcase – long term memory (kng-base, experimental procedure, algorithms)
  - Briefing – temporary memory on the execution environment (communication langauges, ontologies, rules, protocols)




Remote execution Migration

# Benefits of mobile agents

- Reduction of network traffic and client processing
  - Automation of distributed task processing
  - Decentralized -local task processing
  - Asynchronous task execution
  - Flexibility on demand software distribution

- Robustness: reduction of dependence of network availability and client/server availability
- Load-balance optimization of execution

# Problems with mobile agents

- Security
  - Malicious host, malicious agent
- Task execution control and traciability
  - Location and status of agent may be unknown
- Fault management
  - Orphan detection, data inconsistency
- Performance
  - Interpreted code is slow
- Open Environment
  - Heterogeneity of hosts

# Mobile agent platforms

- Aglets: www.tri.ibm.co.jp/aglets
- D'agents: agents.cs.dartmputh.edu
- Grasshoper: www.ikv.de
- Voyager: www.objectspace.com
- Telescript (mobile code)
- Lime

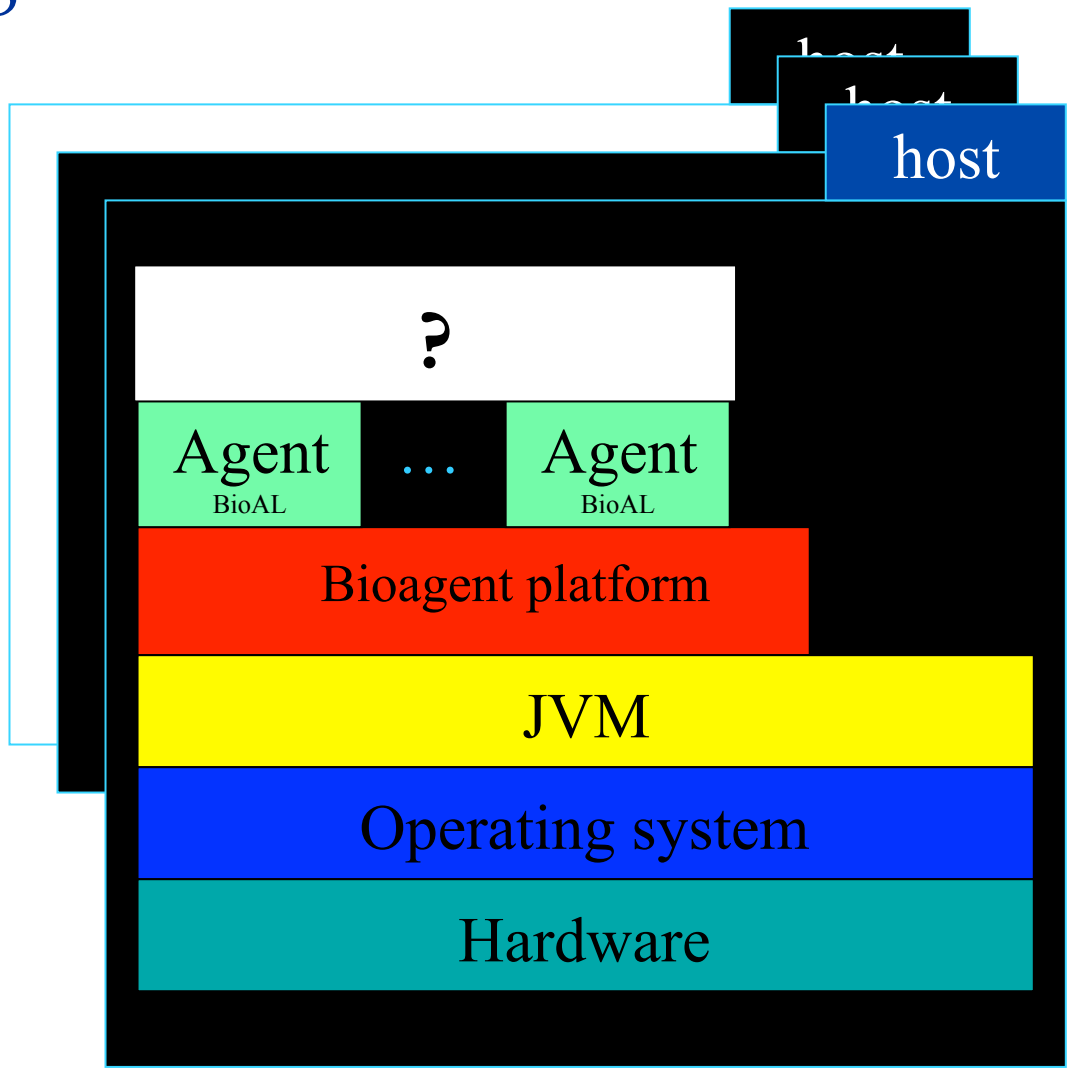
Virtual mobility

- Jade

…

Why another?
because the others were
propetary or simulating mobility

# Bioagent + BioAL



… is Bioagent a platform?
or
a middleware?


# From 2001 … up to 2005

we work to develop Hermes:
a middleware to support activity-based workflow applications over a mobile-agent computing environment

# Bioagent project
# http://www.bioagent.net


- May 2001: 1st prototype of Bioagent platform development of an agent-based application for "Agent for Microarray Data Analysis"
- July 2001: 1st prototype of Bioagent platform proposed at BOF - ISMB in Copenhagen
- 2003: 1st version of HermesV1 a component-based middleware for mobile computing
- 2004: 2nd version HermesV2 a middleware system for mobile computing
- In the future → back to Bioagent

# Tools suitcase

- Wf Management
  - Wf Editor – modelling and definition
  - Wf Checker – analyzing consistency of the model
  - Wf Compiler - translating model to executable code
  - *User interface*: Web-based GUI, Console, …
- System Management
  - Middleware platform deployment tool
  - Account management tool
  - System's Diagnosis tool
  - System maintenance tool
  - Traceability tool
- …

# Workflows in the BioScience Domain

**Definition**

The computerised facilitation or automation of a business process, in whole or part.

(from Workflow Management Coalition-Reference Model)

**Goals**

- to design and implement a data analysis process (standardized protocols - S. Hoon et al. '03)
- to simulate a high-level biological process (Peleg et al. '03, Amici et al.'04)

**Advantages for data analysis, it makes possible:**

- to reproduce the analysis
- to reuse intermediate results
- to create a transparent analysis environment
- to support a good practice
- to free the bioscientist from repetitive interactaction with the web
- to verify structure, functional and dynamic process requirements
- treacibility

# An Agent-based Approach to Tool Integration

[F. Corradini, L. Mariani, 2003, 2004]

Navigating through the Interactive Bioinformatics Tutorial

http://www.isat.jmu.edu/users/klevicca/p53/bigun.htm.

# Activity-Based Application in Bioinformatics Domain

**Problem**: To find the crystallographic structure of the 10 proteins more similar to that of a given genetic sequence, e.g., *atggag … tga*,

**Objective**: To use several Bioinformatics Software Tools available on Internet in order to find the wanted result

1. *Select the 10 proteins more similar to the given sequence atggag … tga*
   - by using BLASTn in GenBank at NCBI

   1st Activity

2. *Search for the PDB ID (crystallographic structure identifier) of each selected proteins,*
   - by using BLASTp in SWISS-PROT at EMBL-EBI
   - by retrieving from PubMed via Entrez Retrieval System at NCBI, abstracts containing PDB-ID information

   2nd Activity

   3rd Activity

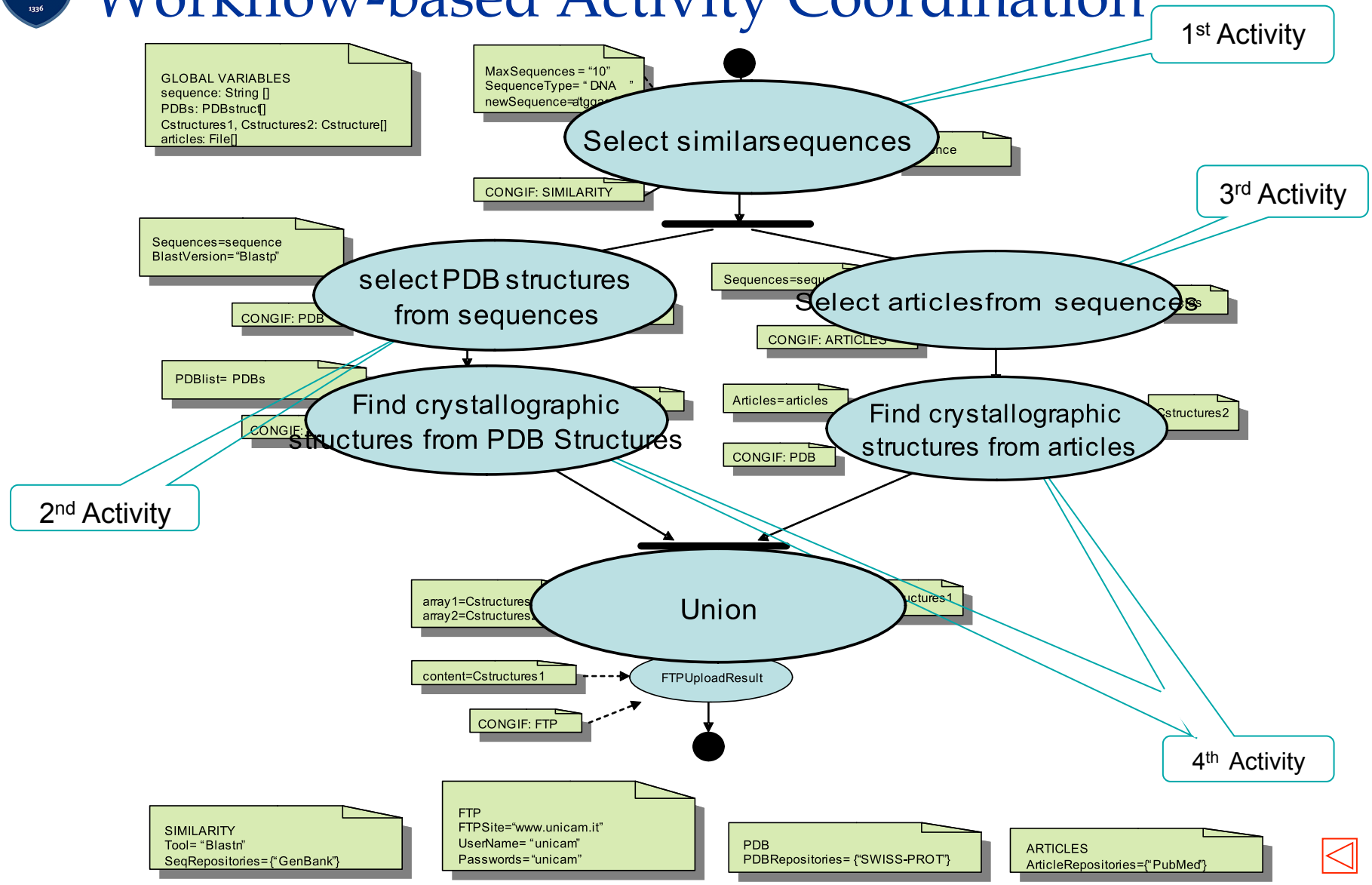
3. *Search for the Crystallographic Structure of any selected PDB ID*
   - find 3-D biological macromolecular structure in Protein DataBank repository

   4th Activity

**Aim**: the automatic execution of the activities by coordinating several Bioinformatics tools, freeing the Bioscientist from the need to know details on data and to continous interact with remote sites.

By defining an *integrated environment* supporting activity coordination data and services integration large scale distributed management system in order to automatically execute an experiment

# Workflow-based Activity Coordination



# Bioinformatics Tools

*Tool 1*:

Environment: NCBI (WebSite): html format
Data: GenBank (DB): proprietary format
Tool: BLASTn (Algorithm): Takes nucleotides sequences in FASTA format
Output: GenBank Format

*Tool 2*:

Environment: EMBL-EBI (WebSite): html format
Data: Swiss-Prot (DB): proprietary format
Tool: BLASTp (Algorithm): Takes protein sequences in FASTA format
Output: SwissProt format

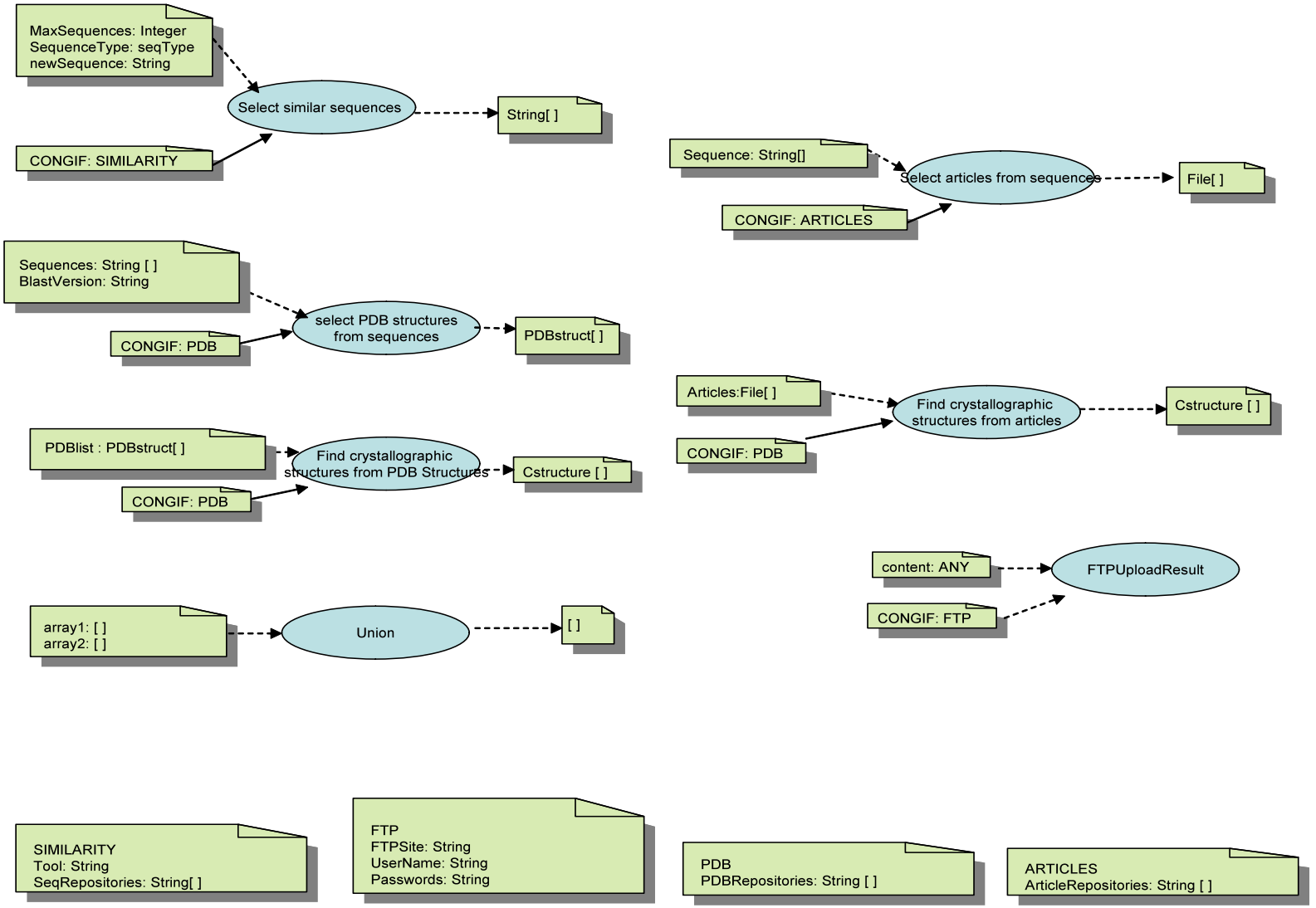
*Tool 3*:

Environment: NCBI (WebSite): html format
Data: PubMed: ANS.1 format
Tool: Entrez Retrieval System: Takes protein name
Output: XML format

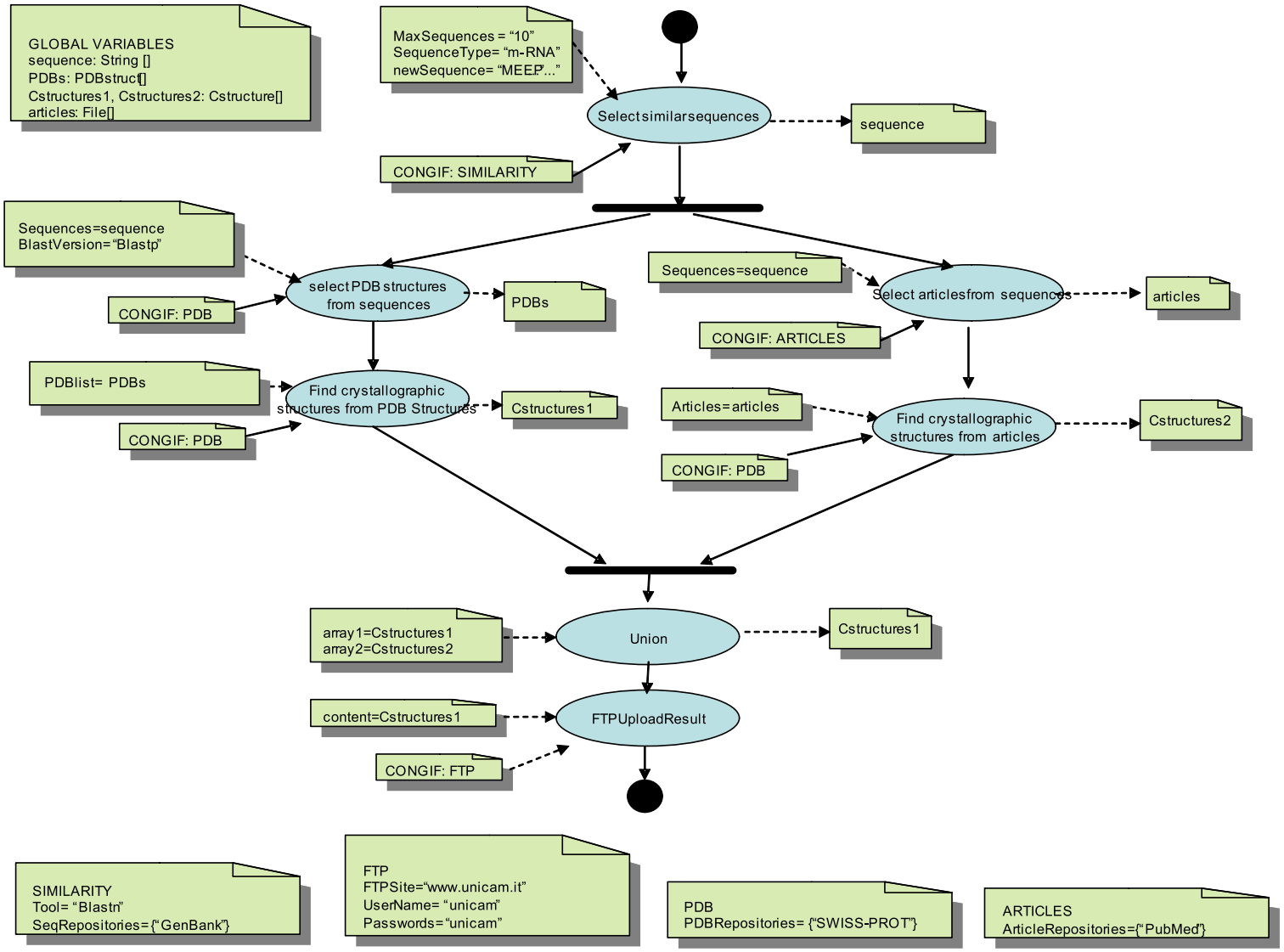
*Tool 4*:

Environment: Protein DataBank web site
Data: PDB(DB): proprietary format
Tool: FASTA (Algorithm)
Output: PDB format (3 dimensional)

# Activity Database (ADb)



# User Level Workflow



# User level to Agent level Compiler

The compiler generates a pool of cooperating agents form the workflow specification

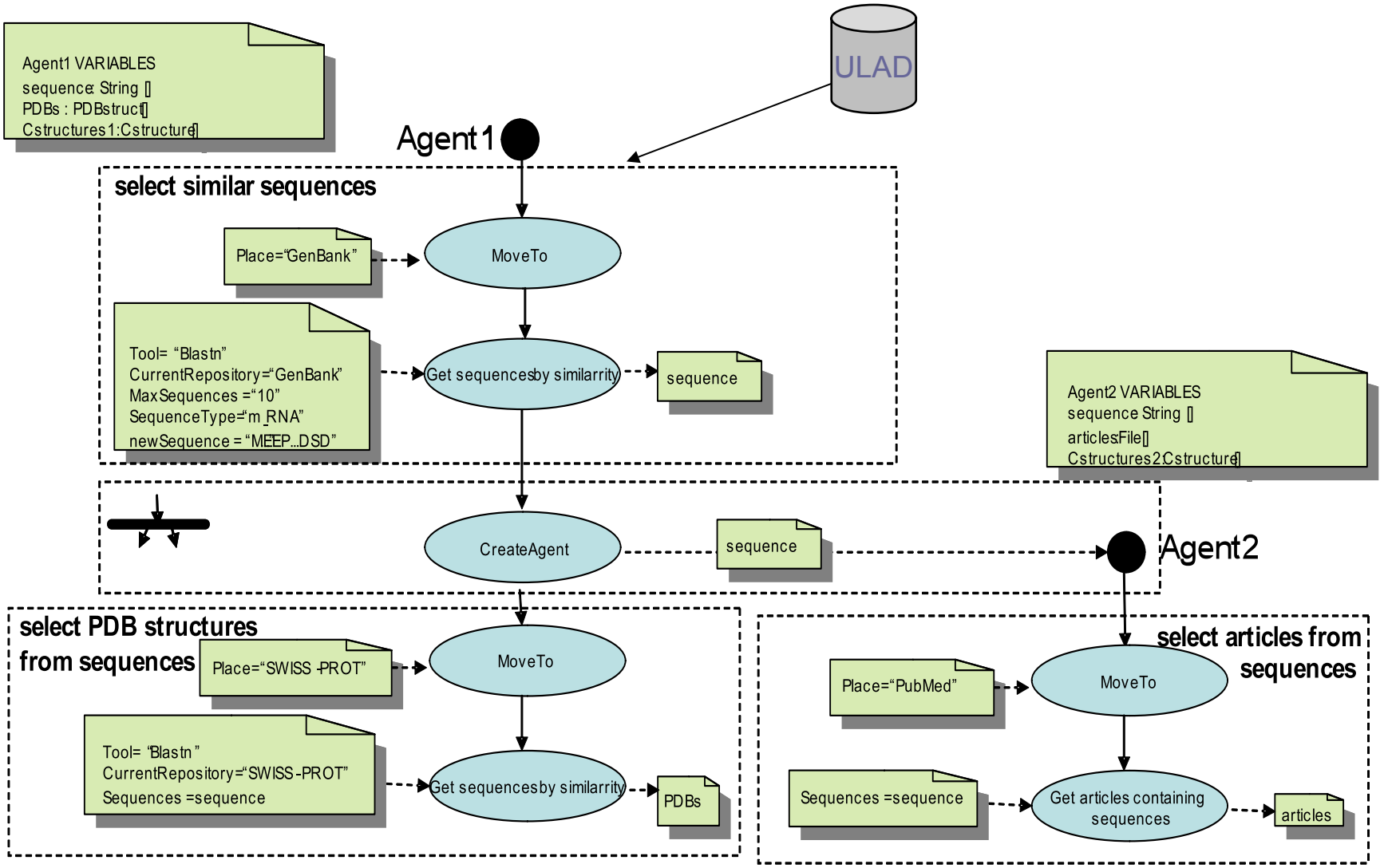
Two steps compilation:

- Step1: from user level Wf to agent level Wf
  - User Level Activity Database (ULAD)
    - contains the mapping from each user activity to an agent level Wf specification
- Step 2: agent synthesis
  - Database of Skeletons (DoS)
    - contains the "empty" implementation of an agent role (skeleton)
  - Agent-Level Activity Implementation Database (ALAID)
    - contains the implementation of each agent level activity

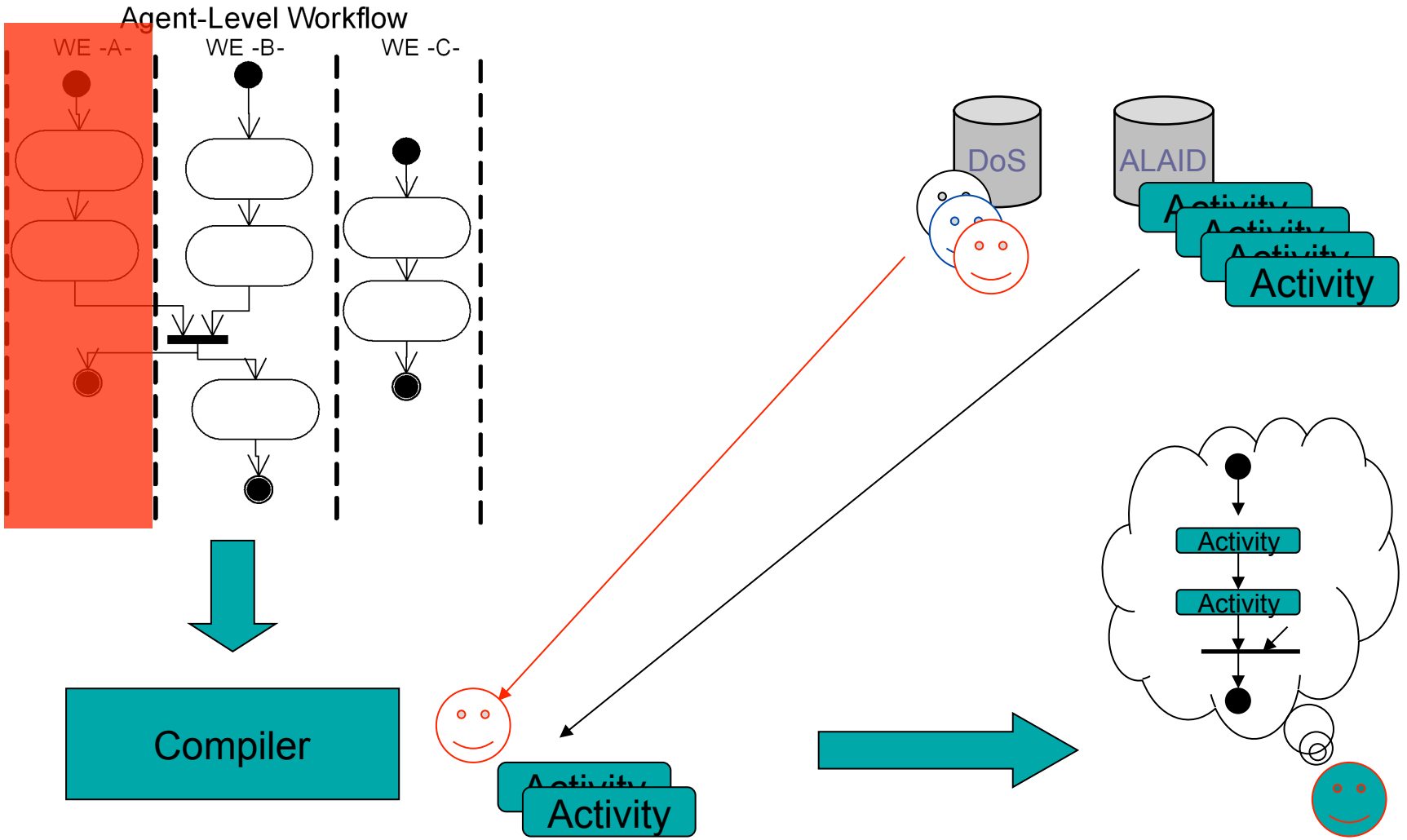
Context-awareness

The compiler takes advantage of information about the state of the global environment during the agent synthesis

# Compiler Step 1: Agent level workflow



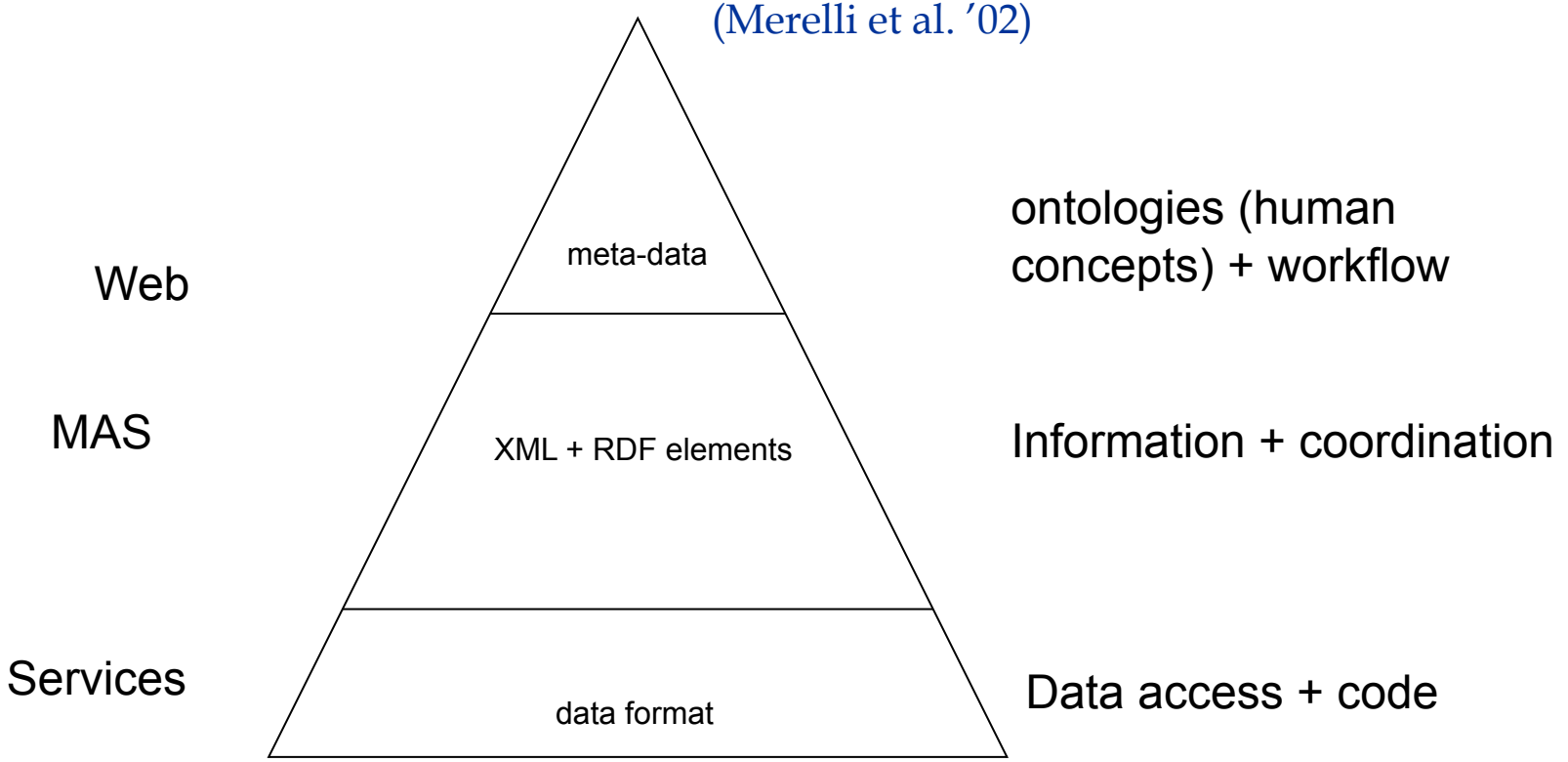
# Compiler Step 2: executable application agents




# The agent-based supporting technology

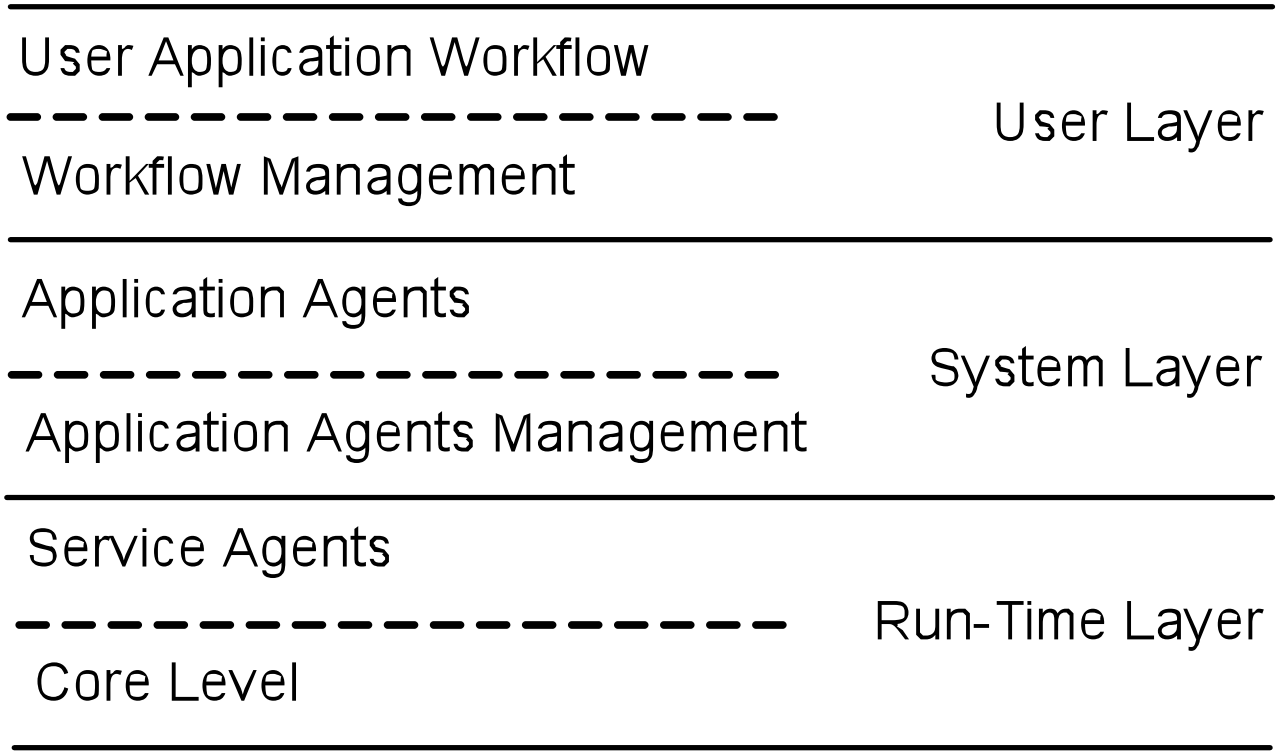
# From Data to Knowledge and vice versa

(Merelli et al. '02)

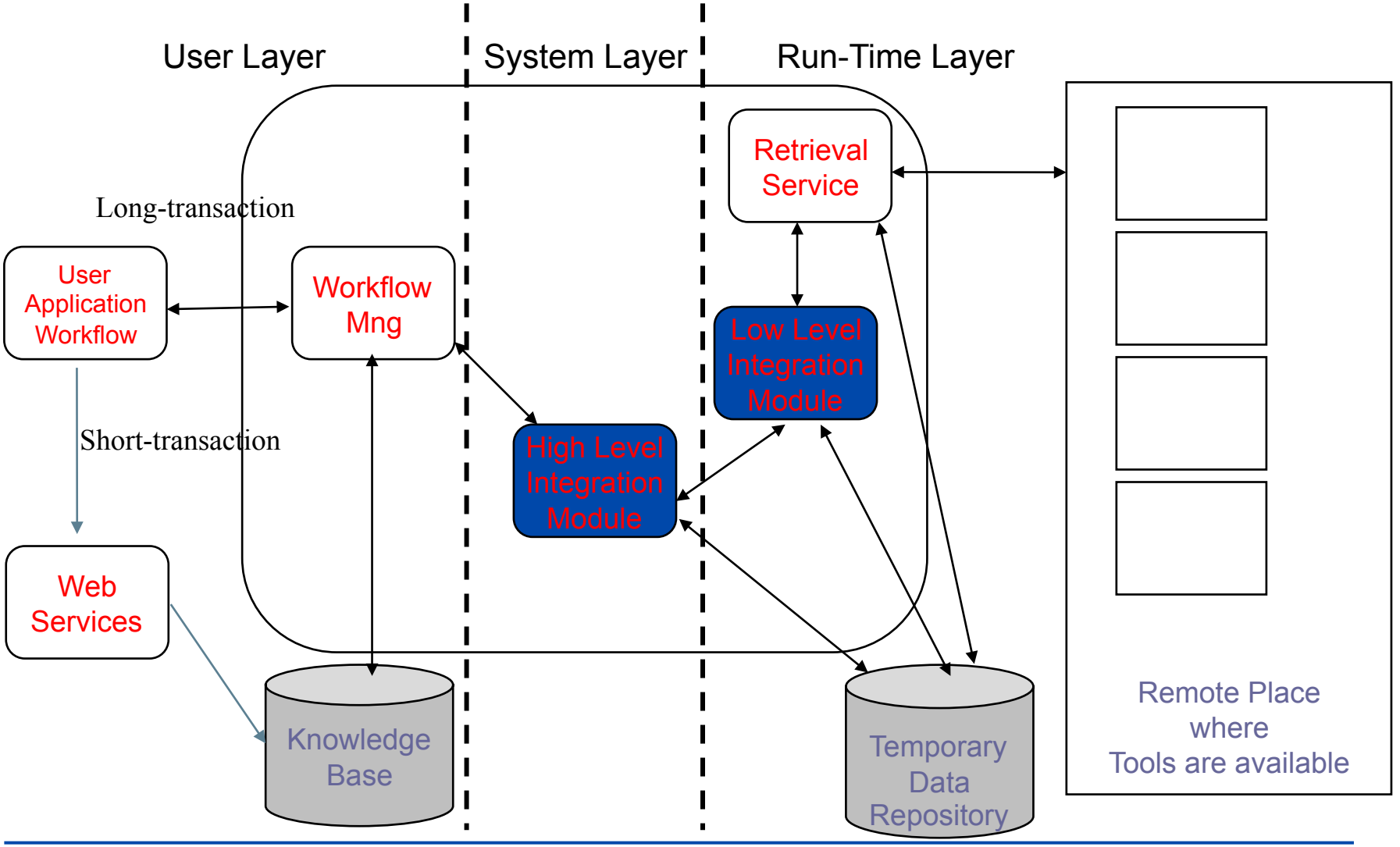


# System's software architecture

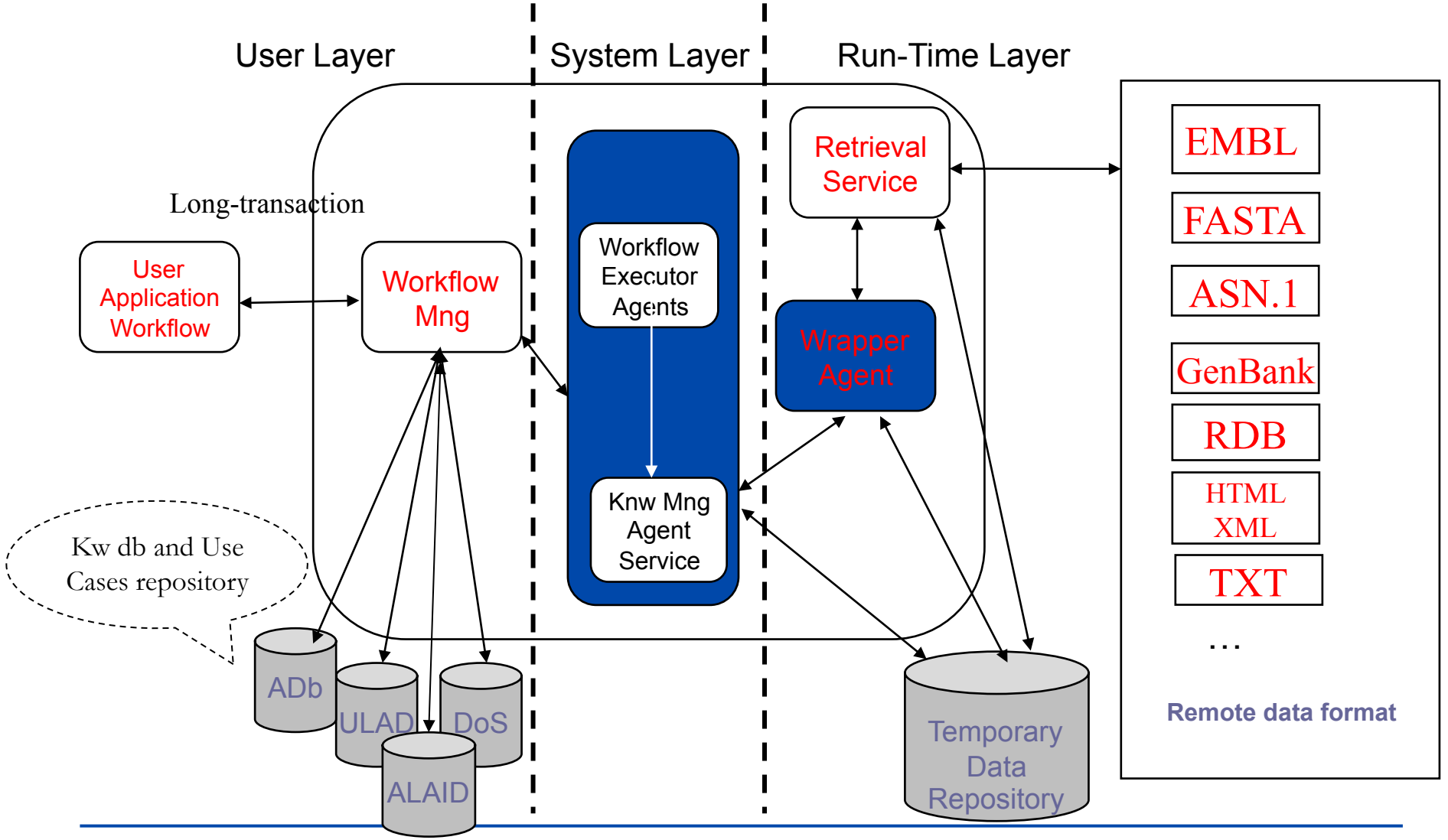
(Corradini et al '03)



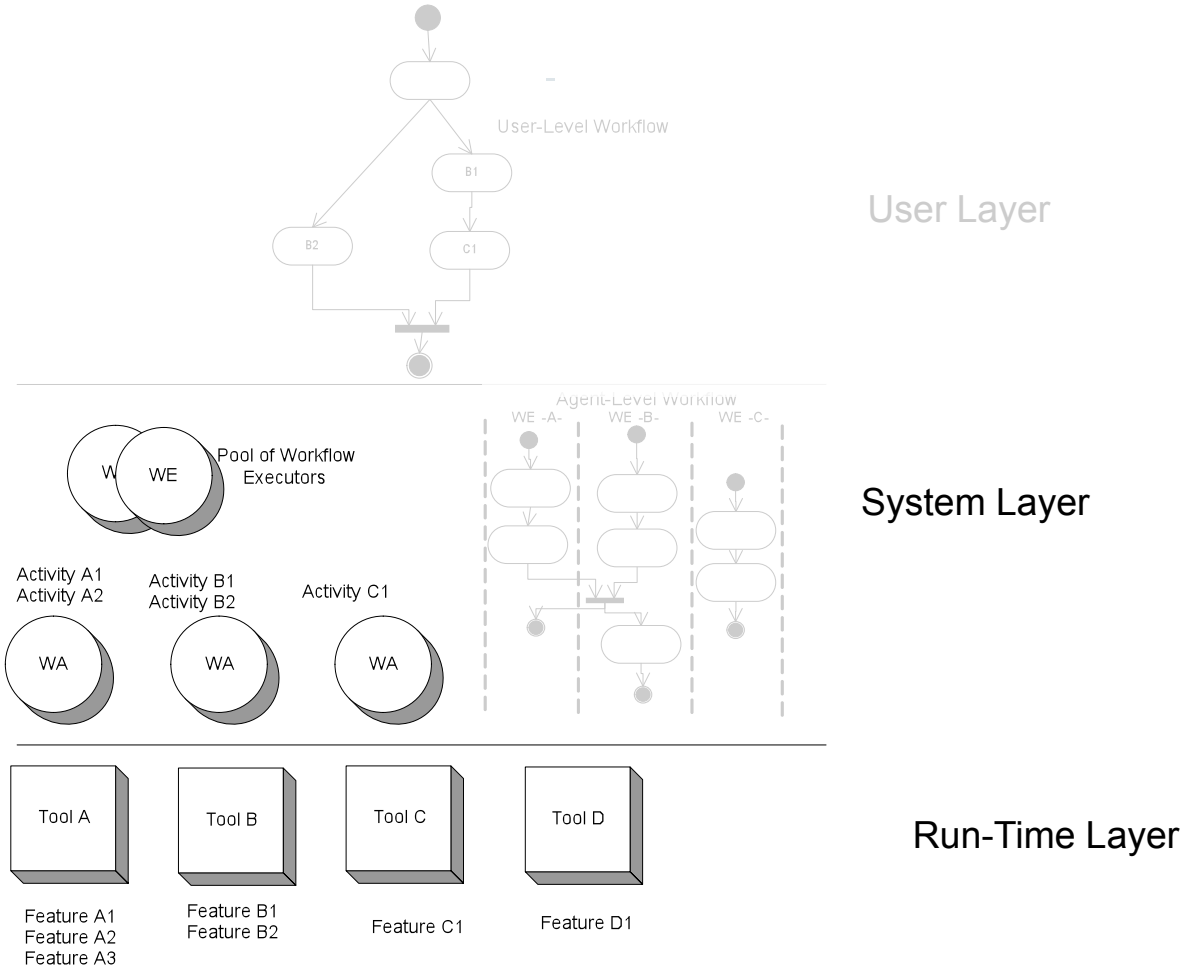
# The O2I system's architecture



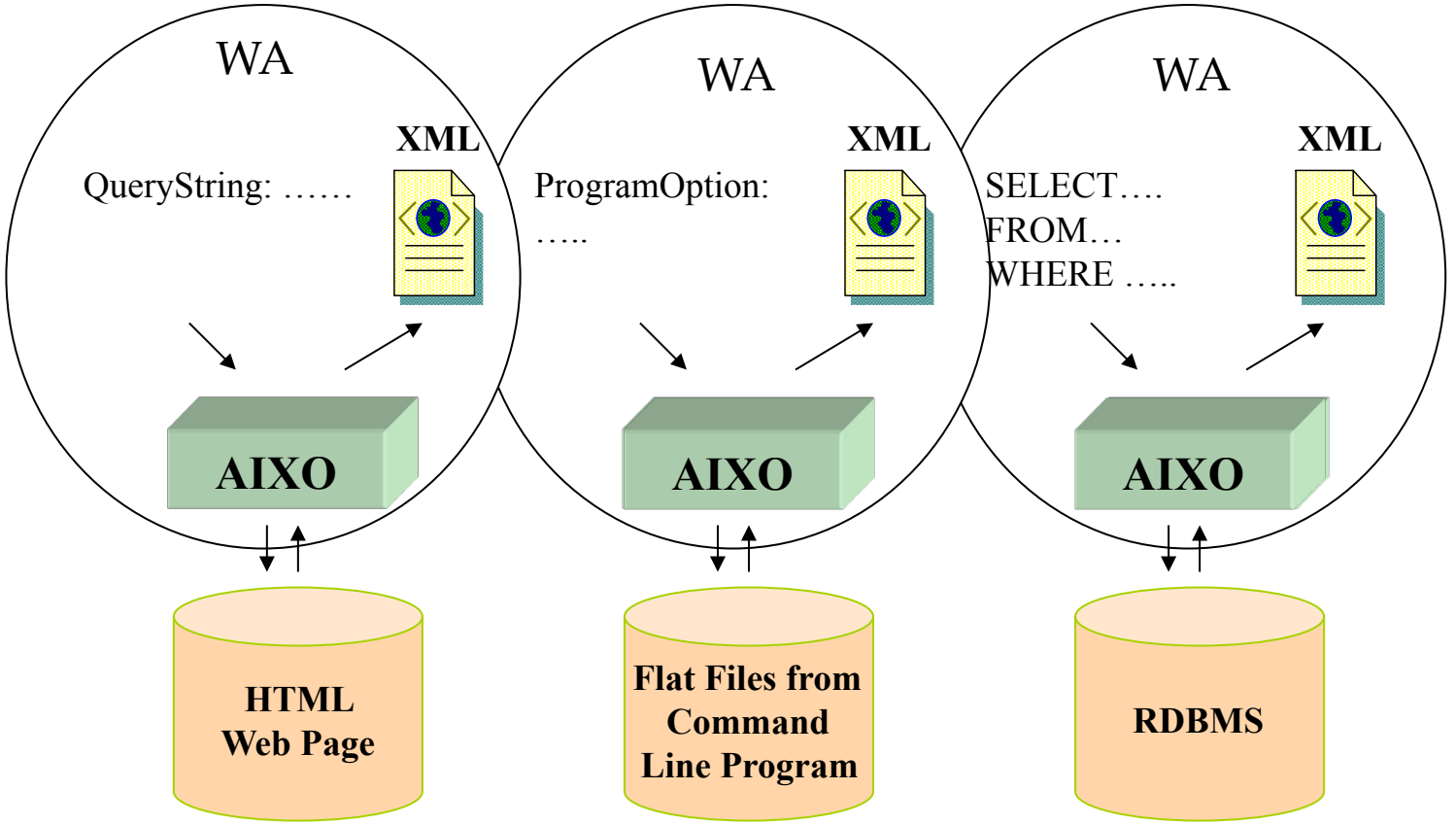
# BioAgent System Architecture for O2I



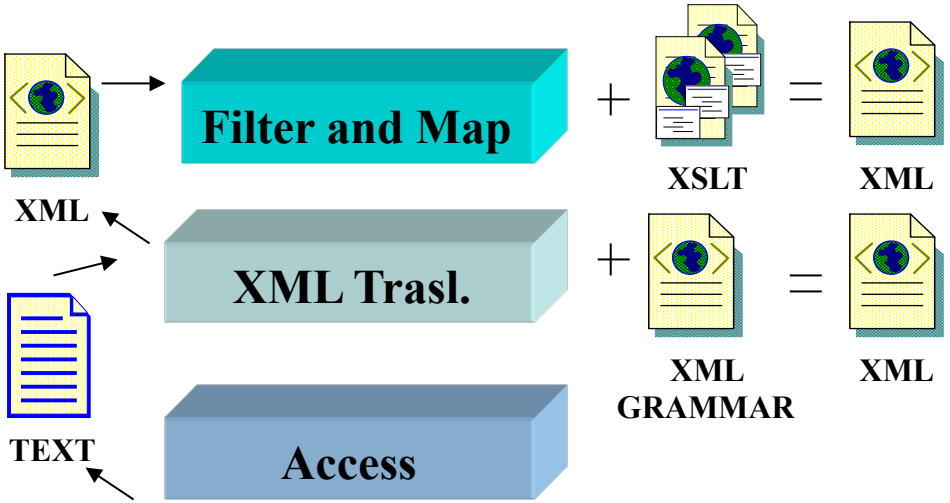
# Application-agents (WE) and Wrapper-agents (WA) cooperative environment



# Wrapper-Agent: general scenario



# Wrapper-based System: Retrieval articles about P53 protein




```
<…>
 <entry>
        <ID name="P53_HUMAN" type="STANDARD" molecule="PRT" lenght="393"/>
        <AC value="P04637"/> <AC value="Q16848"/> <AC value="Q9UBI2"/>
        <DT day="13" month="AUG"  year="1987" rel="05"/>
  </entry>
```

# Hypothetical Scenario for O2I project

**Oncology over Internet** project aims to develop a framework to support searching, retrieving and filtering information from Intenert for oncology research and clinics

**A possible scenario involving the use of biological resources**

- Biological resources (micro-organisms, cell lines …) are essential for implementing a good, reproducible experiment
- High quality biological resources are available at some specialized centers (Biological Resources Centers:ATCC, DSMZ, …) and related catalogues are available on-line
- Molecular Biology (MB) databases, e.g., sequence dbs, often include information (strain numbers, accession numbers) on the original resources
- Researchers assessing MB databases often need extended information regarding resources to finally request materials

# A simple workflow example

**Use context**: to verify a mutation experiment by reproducing

**Goal**: Retrieve abstracts from a literature db for identifying the best cell line for reproducing a human TP53 mutation experiment linked to a particular tumour-habits-sex combination

**Activities**: use Bioinformatics Services available on Internet in order to achieve the desired result

1st Activity

1. Retrieve all mutations (IDs) observed in the **7th exon** in **men** who are ex-smokers and drinkers
   - by searching p53 mutations database SRS implementation at IST, Genova

2nd Activity

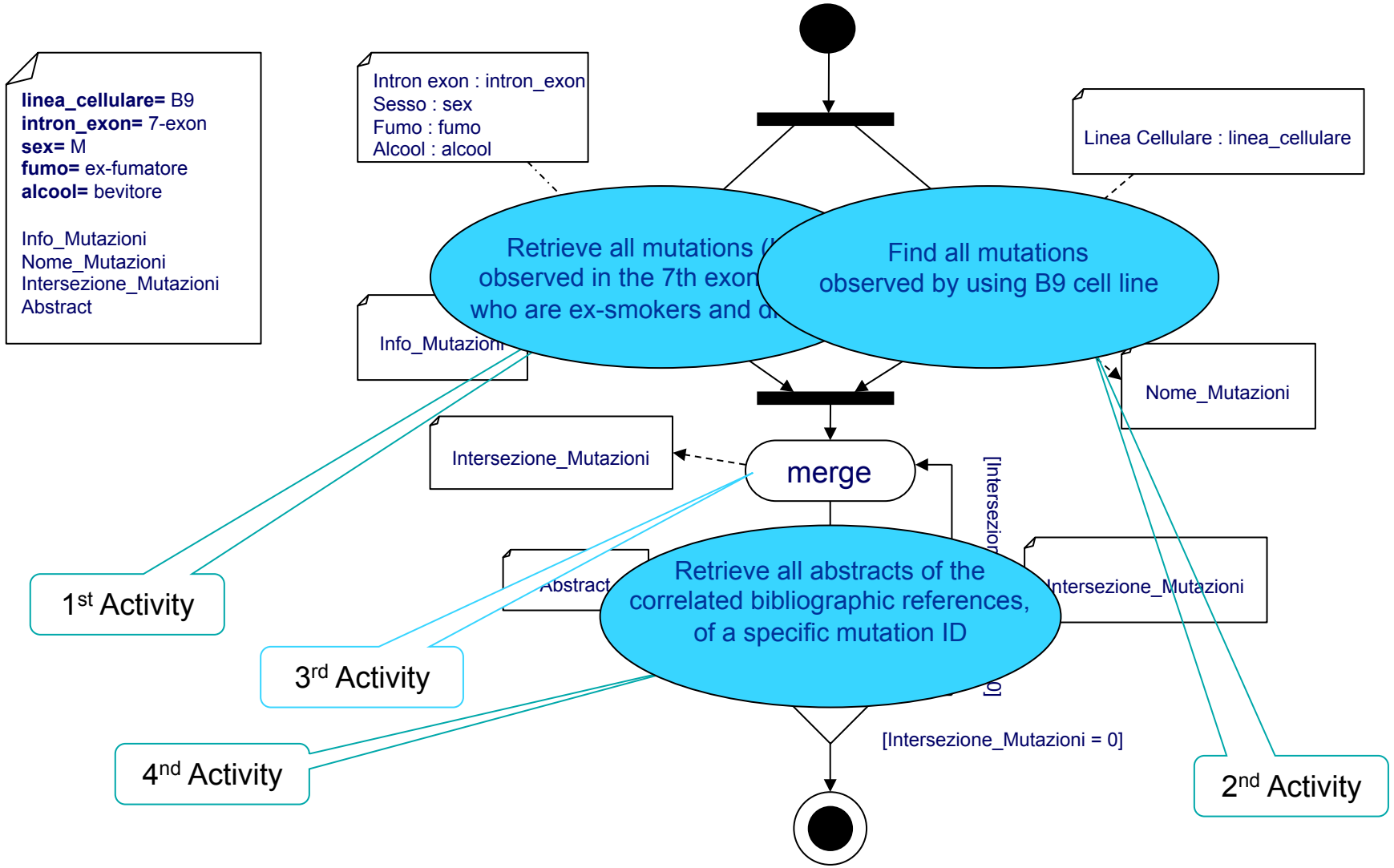
2. Retrieve all mutations (IDs) observed by using **B9 cell line** as original resource
   - by searching p53 mutations database SRS implemerntation at IST , Genova

3. Retrieve all abstracts of the correlated bibliographic references, of a specific ***mutation ID***
   - by searching Medline

3rd Activity

**Achievement**: To integrate on-line Bioinformatics data in a unique result freeing the Bioscientist from the need to personally interact with remote sites

# An example of workflow at user level



# Software Architecture

(Corradini et al. '04)

**User Application Workflow**

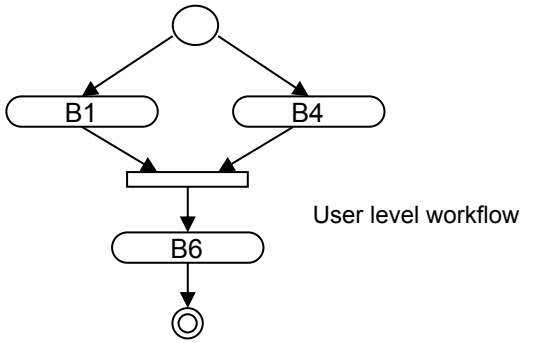
**Workflow Management**

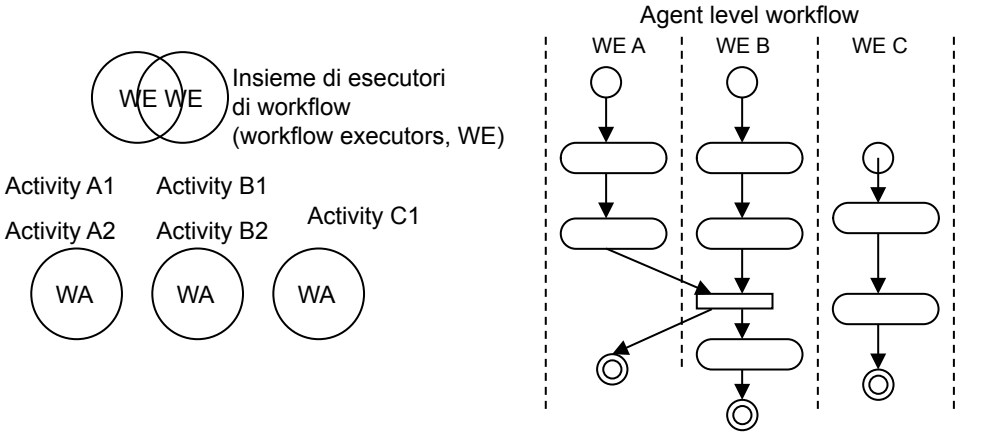
**Application Agents**

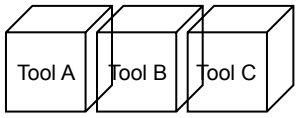
**Service Agents Management**

**Resources (data, tools and services)**

**Core**



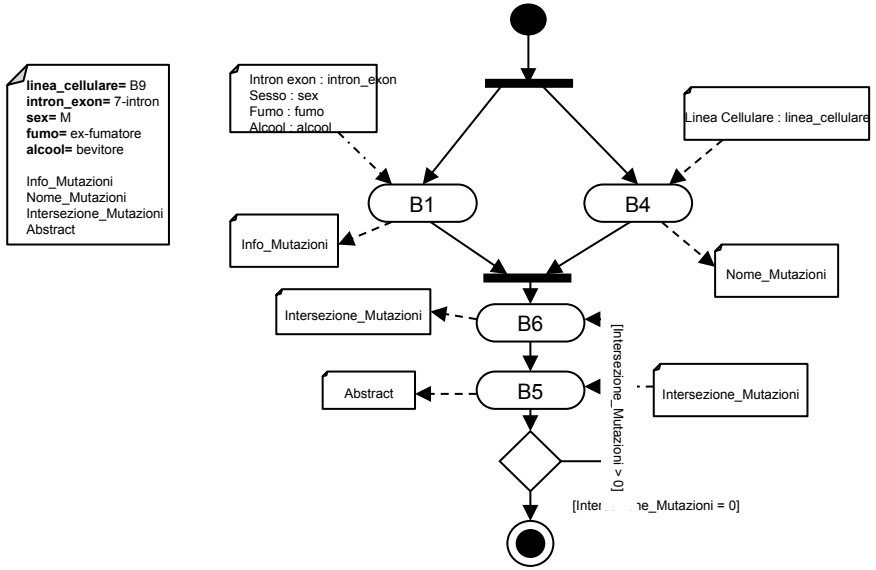

User Layer

System Layer

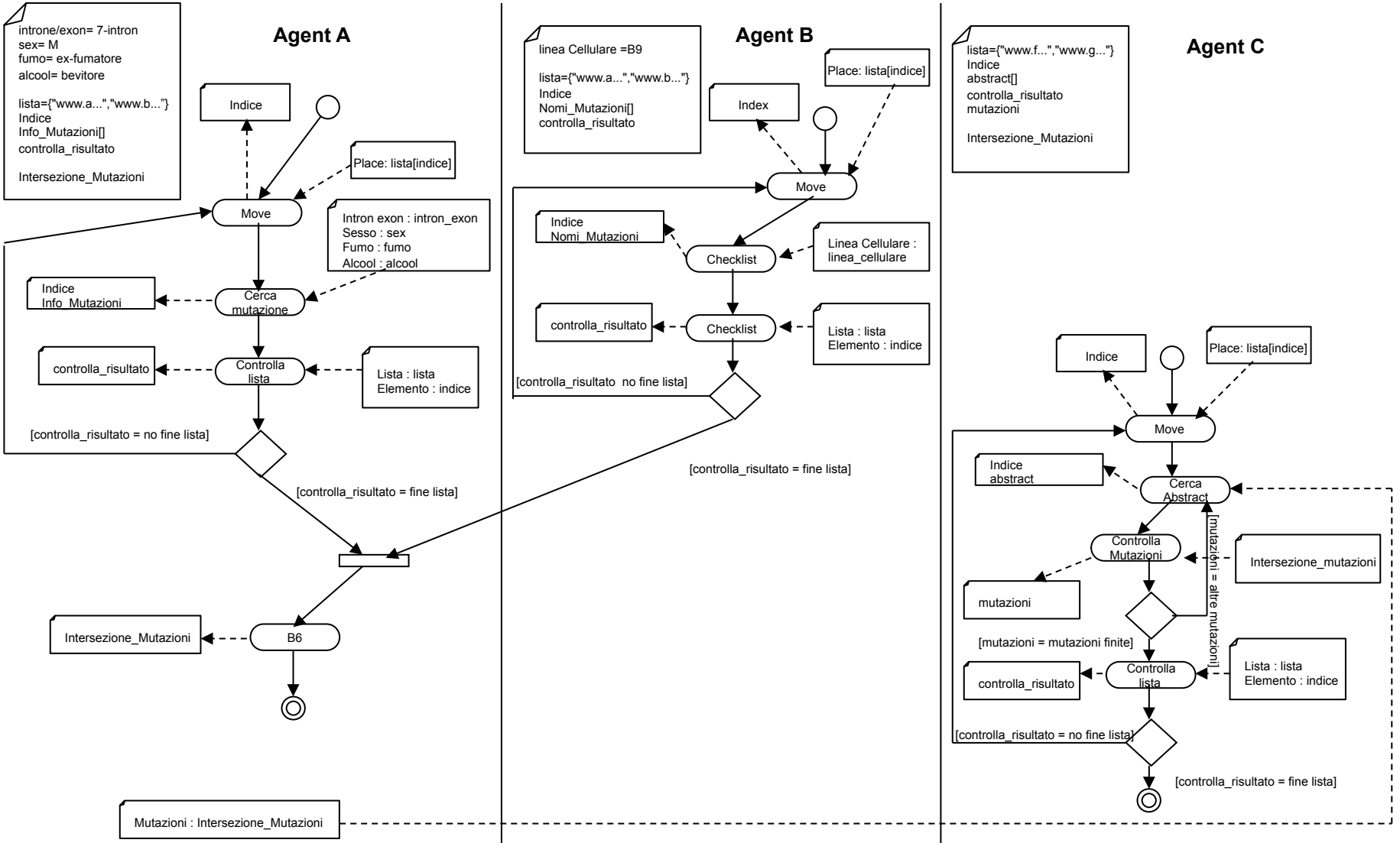
Run-Time Layer

# Workflow Management Service

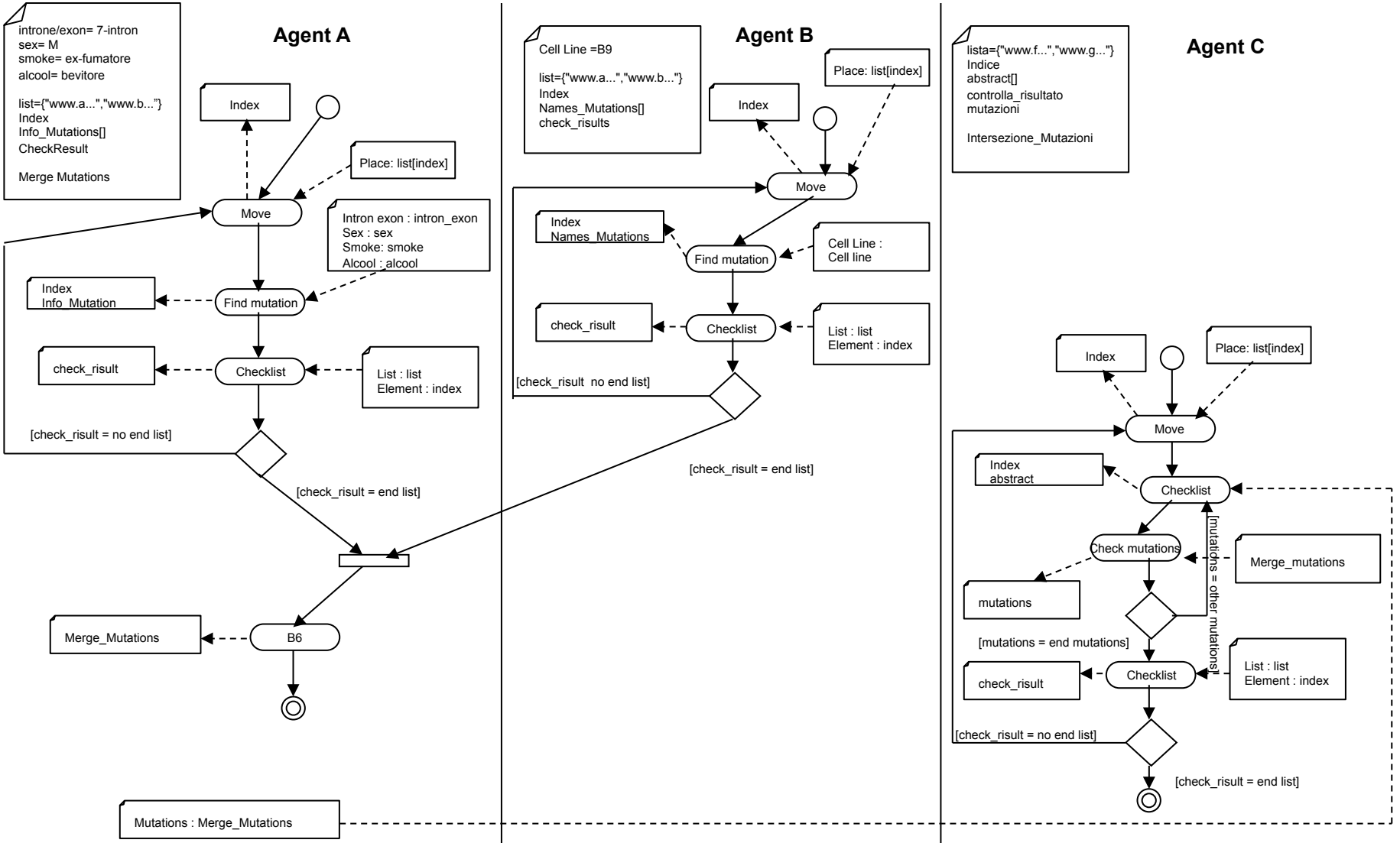
The **workflow management service** is a prototype which allows the defintion of complex queries (use case) by using workflow as a coordinaton model.
The application uses a databases to allow the user to configure use cases and manage resulting data.
The application provides a graphical interface
The application is a plug-in for the BioAgent platform



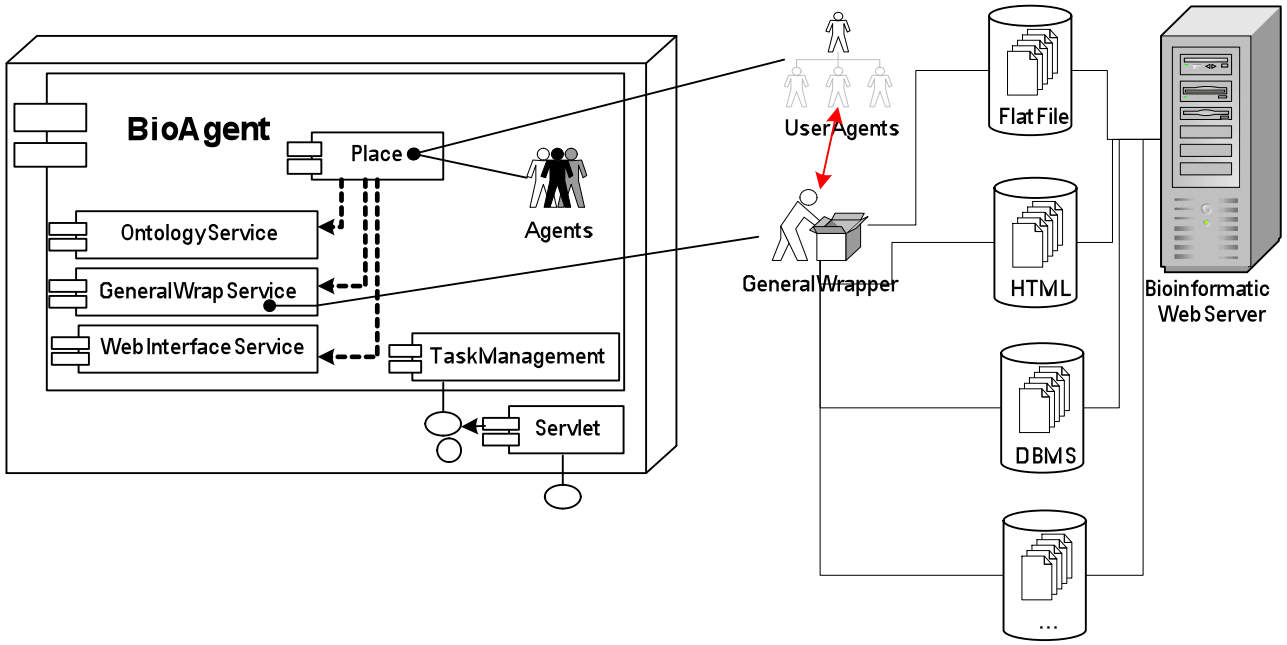
# An example of workflow at agent level



# An example of workflow at agent level



# BioAgent x O2I (Merelli et al. '02)




http://www.bioagent.net

# Use Cases

Use cases are a set of scenarios tied together by a common user goal. Where a scenario is a sequence of steps describing an interaction between a user and a system.

UML definition

Some examples:

- find all possible **mutations** that involve a given **protein** *X*
- find all possible **cell lines** related to a given **tumour** *Y*
- select all possible **abstracts** referring to a given **protein** *Z*
- select all possible **abstracts** referring to **cell line** *W*

Workflow as a composition of uses cases

# Use cases and Application Domains

## Use cases in Cell Line domain

**A1:** Find information about the cell line named x
**A2:** Find all cell lines derived from a specific tumour or pathology
**A3:** Find all Cell Lines producing a specific protein
**A4:** Given a specific Cell Line, find all related bibliographic references
**A5:** Given a specific Cell Line, find all information about produced proteins

## Use cases in Mutation domain

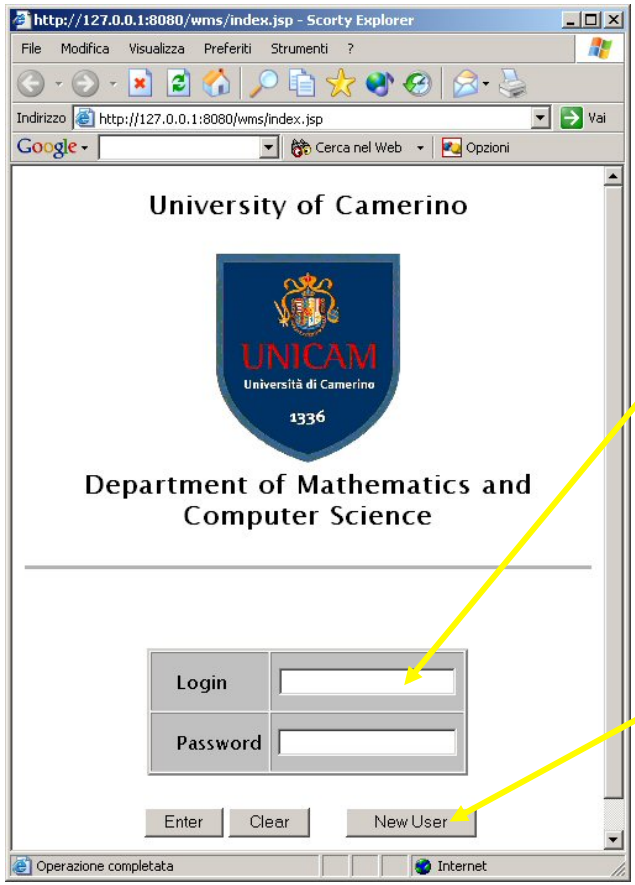
**B1:** Find all mutations observed in a specific intron/exon in subjects with specific sex and life habits (i.e. smokers/ drinkers)
**B2**: Find all mutations in subjects affected by a given pathology
**B3**: Find all subjects affected by a tumoural pathology and with a given protein mutation
**B4**: Find all mutations observed by using a given cell line
**B5**: Given a specific mutation, find all abstracts of the correlated bibliographic references

## Use cases in Bibliographic database domain

**C1:** Select all abstracts of bibliographic references, whose text includes a given term
**C2:** ……..

# User accounts managements

This page is offered by the **BioAgent** platform site. It is possible to access to the system services

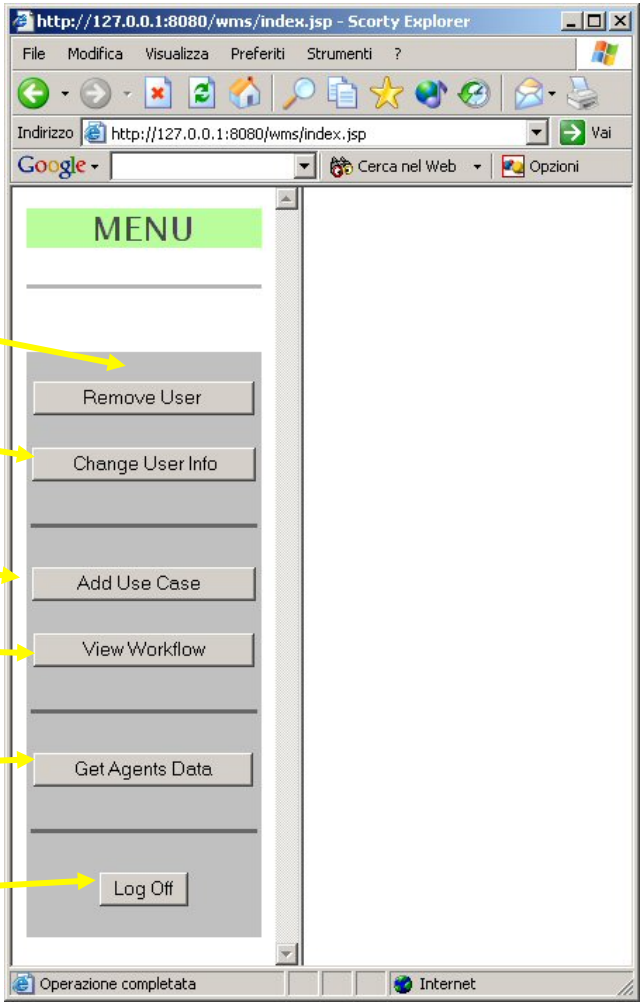



If you have already a login and password, *Submit*

Through the *New User* botton you can register

# The offered services

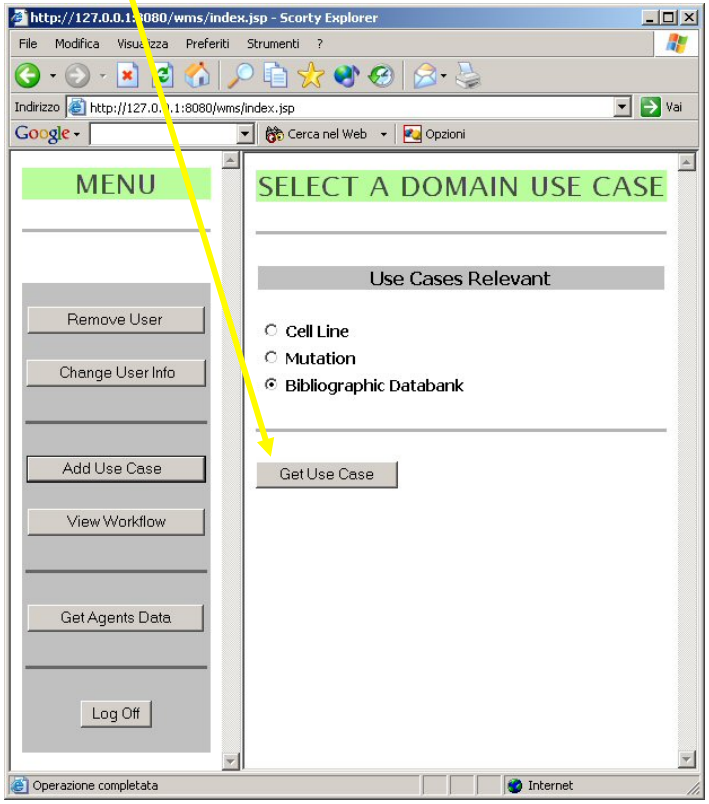
- Remove a user
- Change user inforation
- Add a new use case
- View the available workflows
- Get “agent” data
- Exit



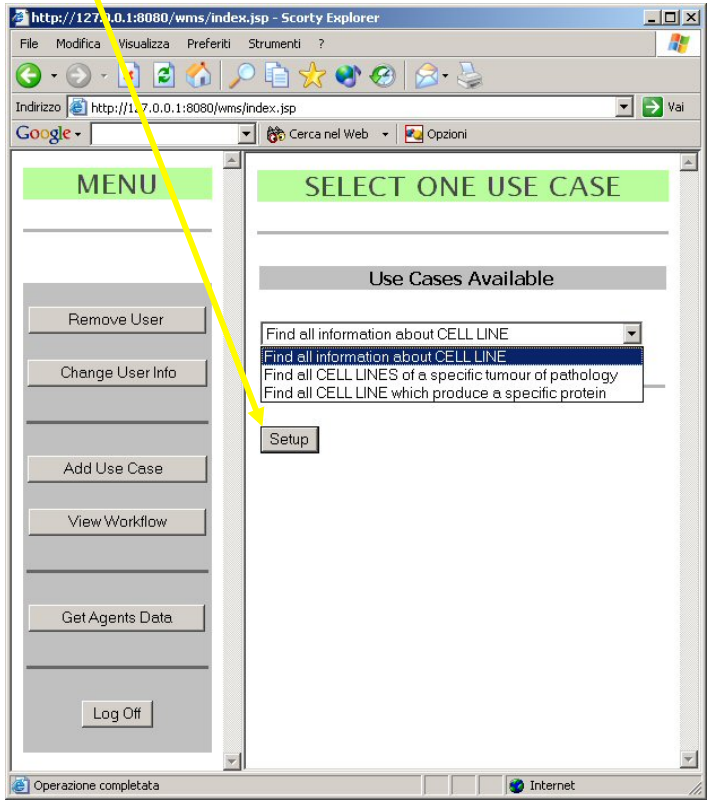
# Use Cases management

The *Get Use Case* botton allows to configure a new use case

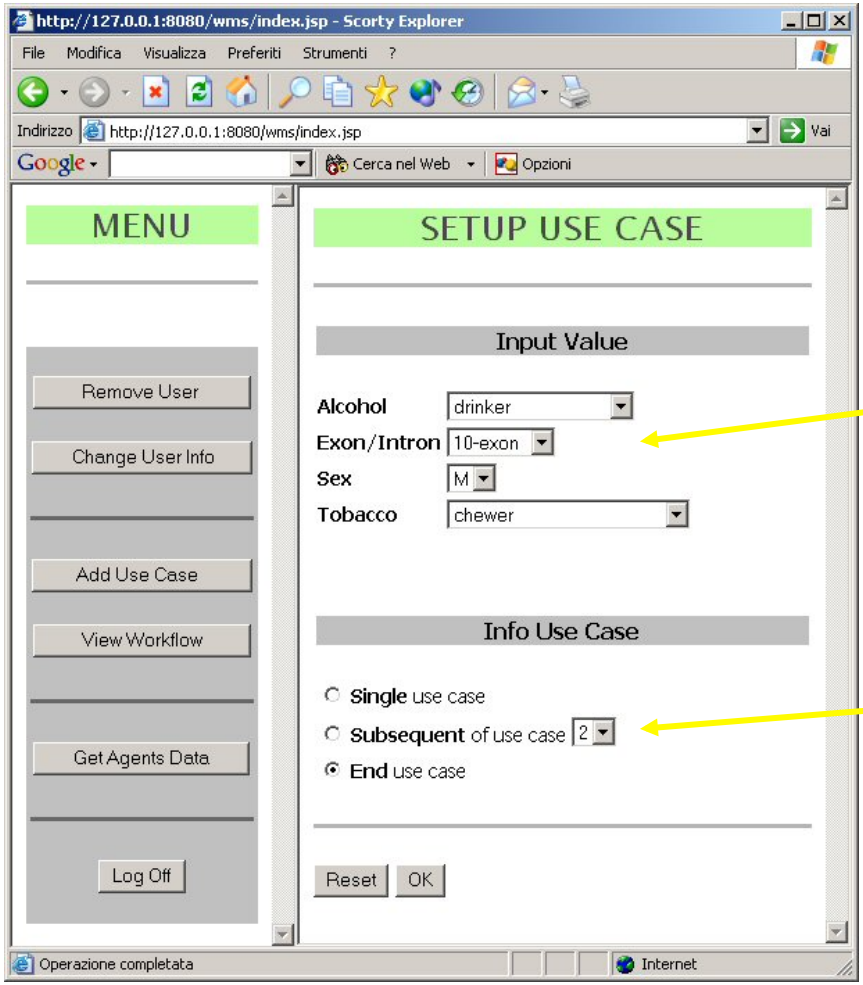
*1*- Choose the application domain




*2*- Choose one of the available use cases



# Use Cases configuration



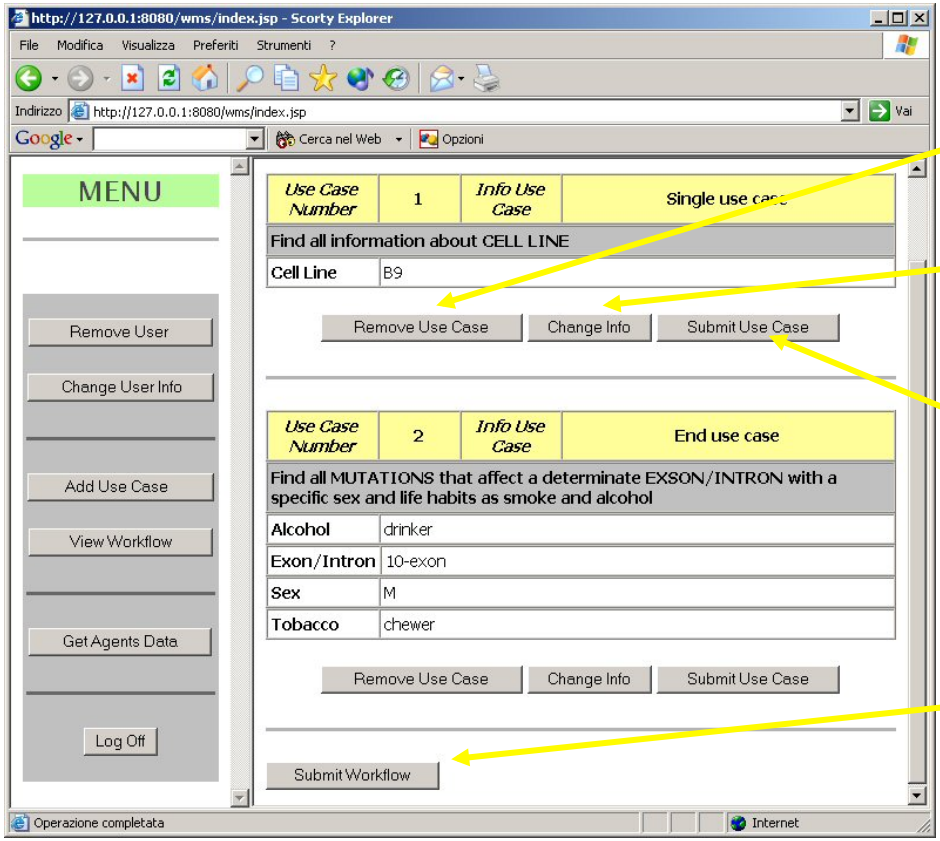

**B1:** Find all mutations observed in a specific intron/exon in subjects with specific sex and life habits (i.e. smokers/ drinkers)

Input parameters

Choose the coordination operator

# The Workflow and Use cases

Global textual view of the workflow. *Submit Workflow*




Remove the use case

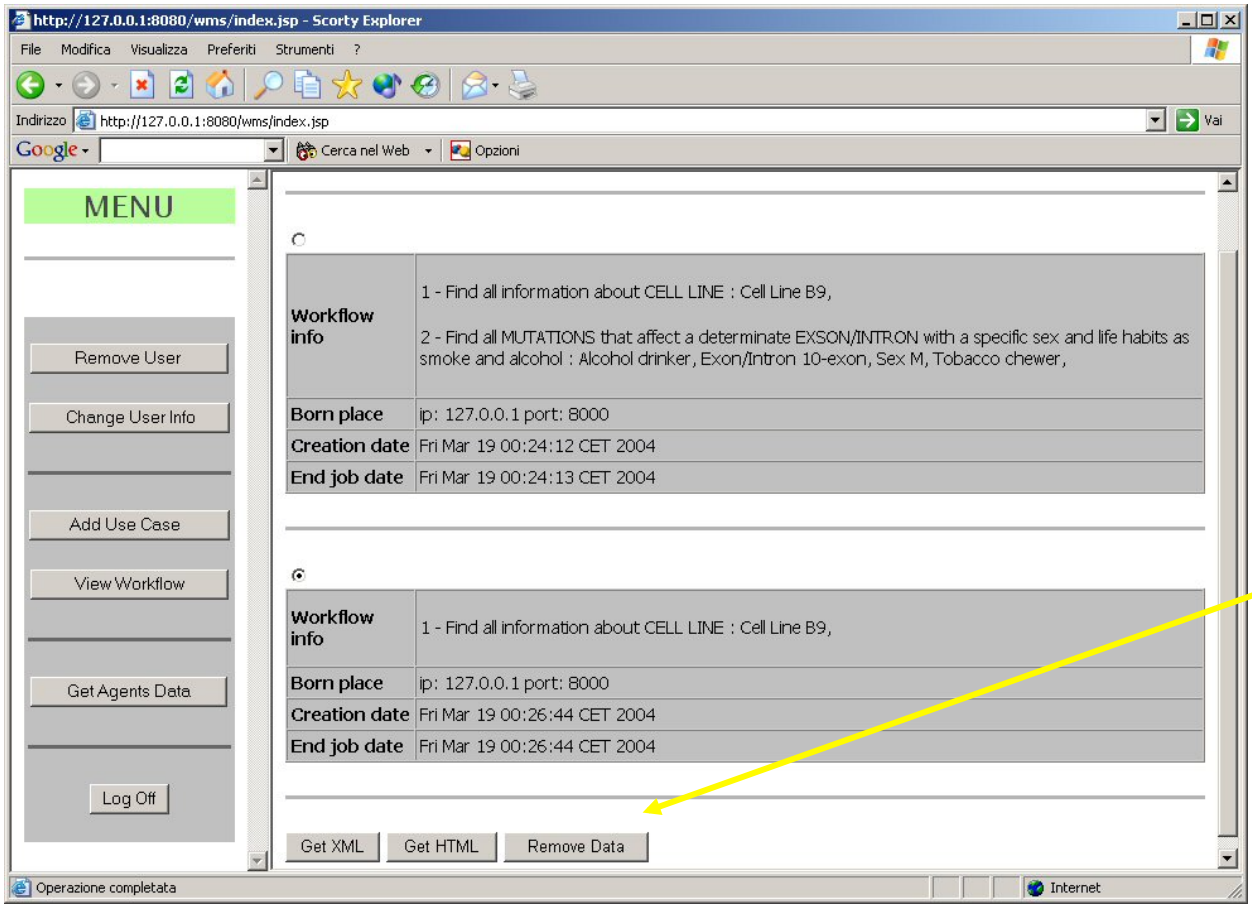
Change configuration of a use case

Submit a single use case

Submit the workflow

# Data management

The *Get Agents Data* allows to manage data resulting from the workflow submission

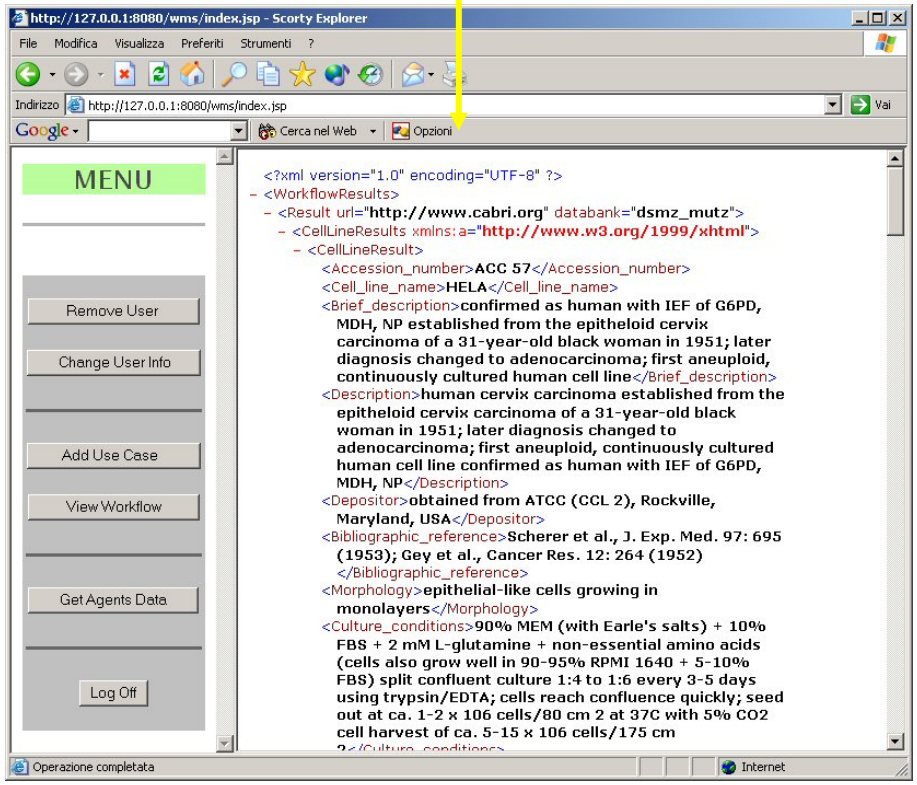


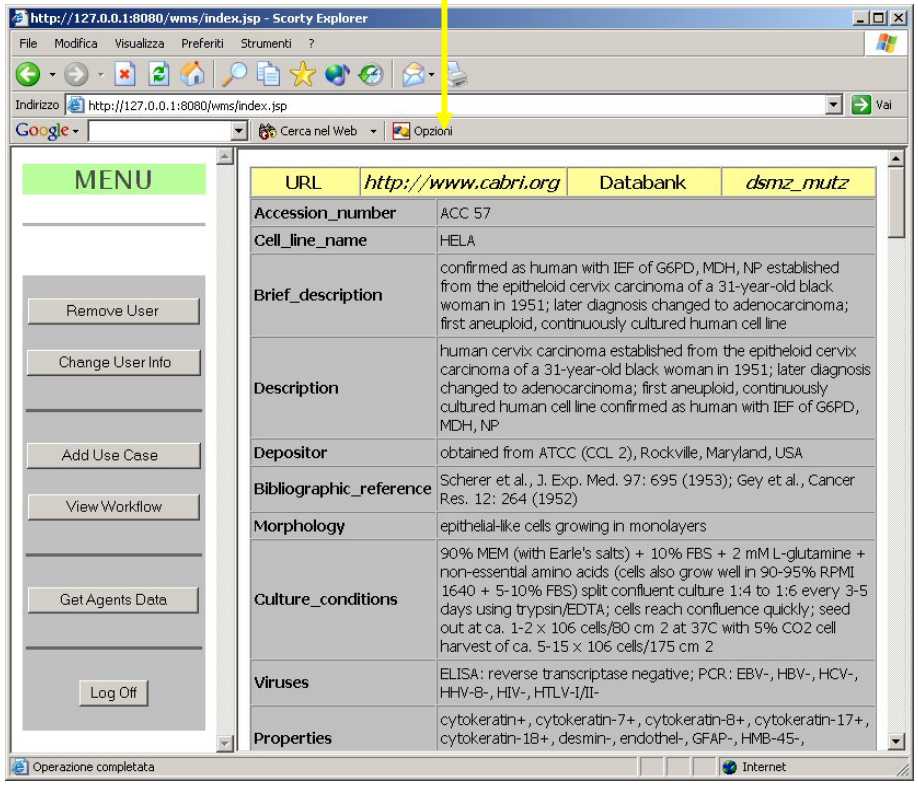

Once we have selected data of interest, we can remove or view in XML or HTML format

# Data view

XML data format

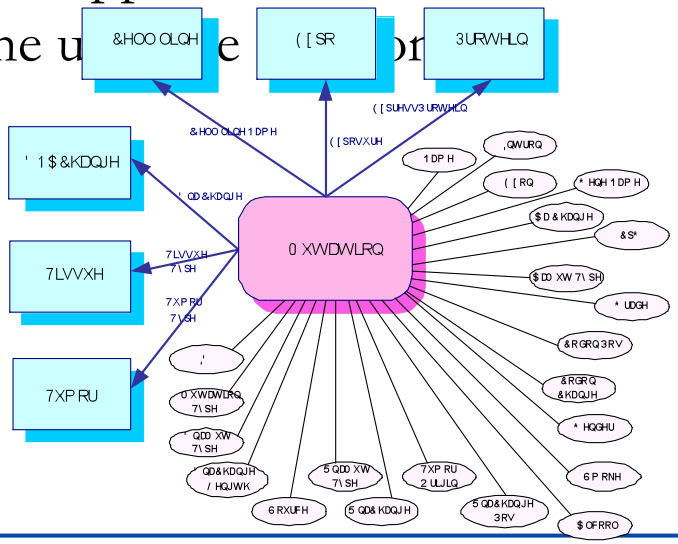
XHTML data format

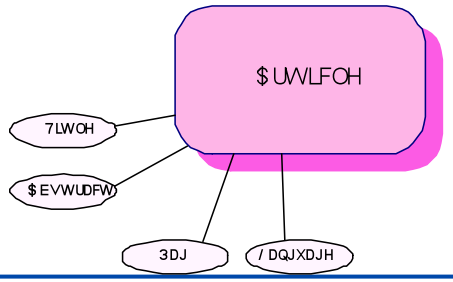


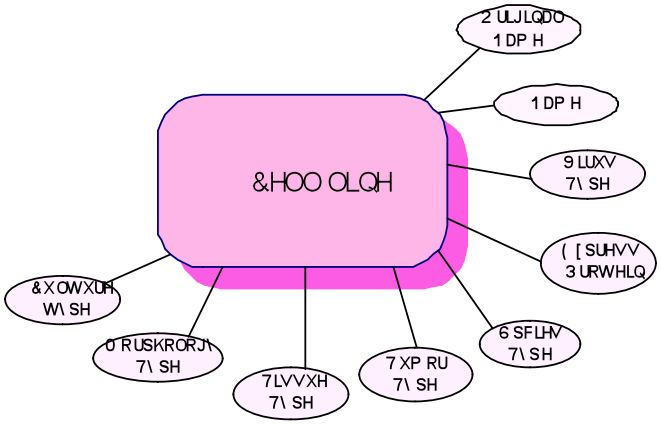



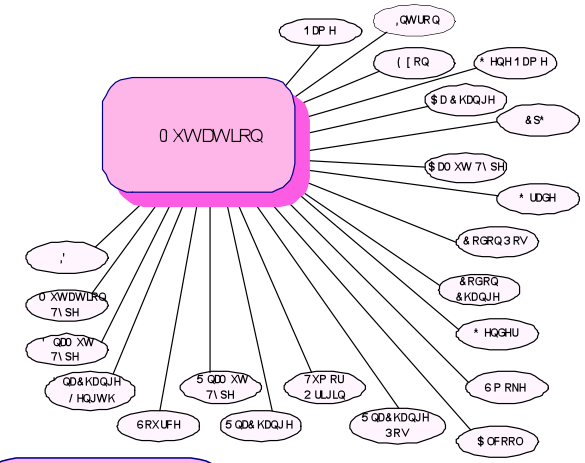
# Future extensions to the tool

- Design and implementation of a knowledge database
- Automatic generation, from a workflow to a Multiagent system that behaves as the execution engine of a WfMS
- Development of an ontological service to support the user during the u e or

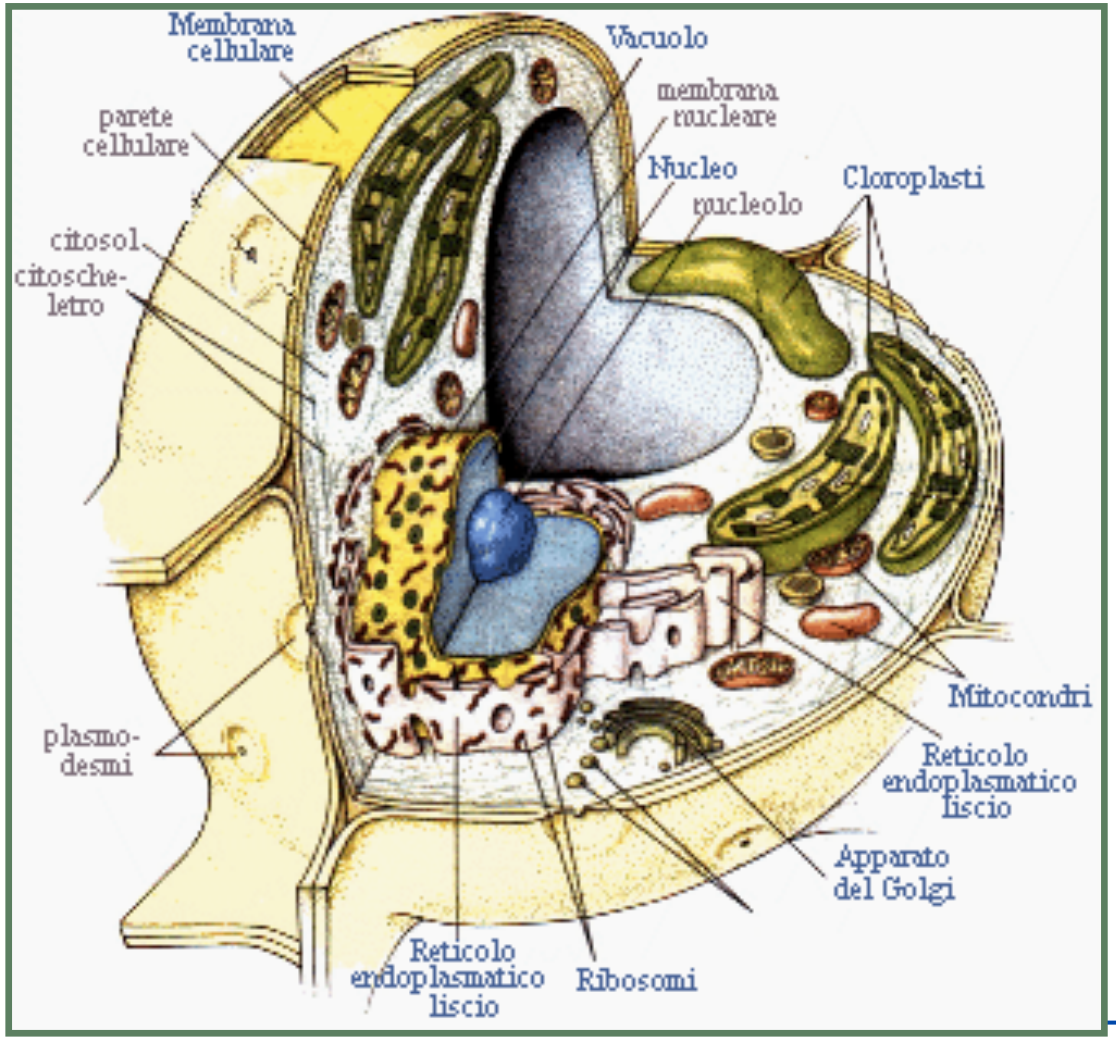


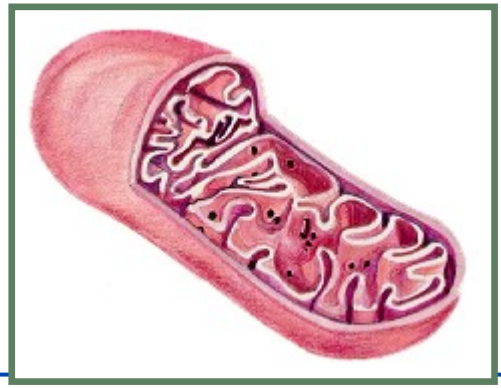



# La Struttura Cellulare




## Componenti Coinvolte

- **Citoplasma**
  - **Mitocondrio**
    - **Matrice Mitocondriale**
    - **Membrana Mitocondriale Interna**

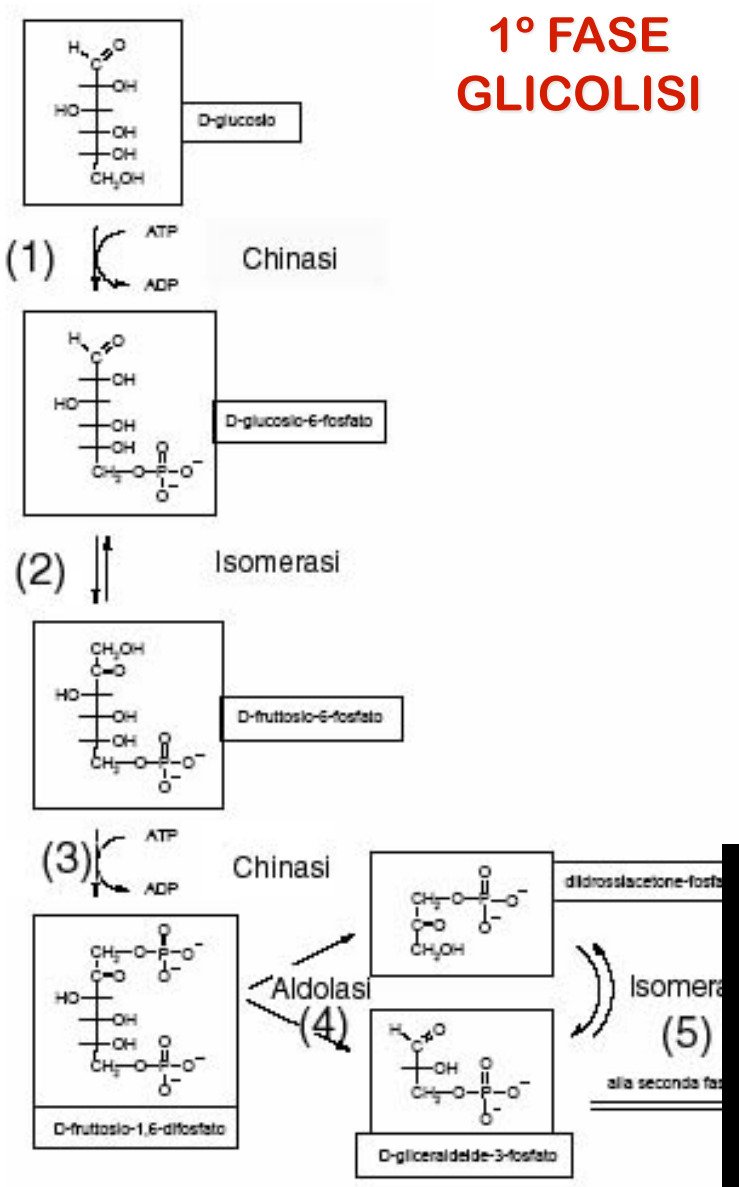


# Processi Cellulari

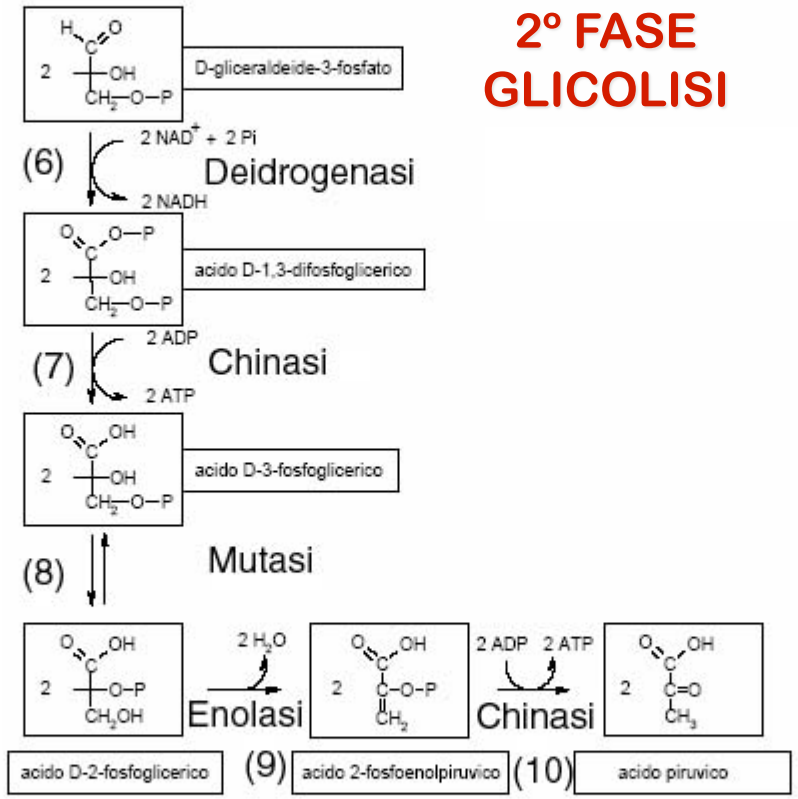
- **Nel Citoplasma:**
  - Glicolisi
  - Fermentazione Lattica→(senza Ossigeno)
  - Fermentazione Alcolica→ (senza Ossigeno)
- **Nel Mitocondrio:**
  - Respirazione Cellulare→(se c'è Ossigeno)
  - **Matrice Mitocondriale:**
    - Ciclo di Krebs
  - **Membrana Mitocondriale Interna:**
    - Catena Respiratoria
    - Fosforilazione Ossidativa

# Citoplasma – Glicolisi

## 1° FASE GLICOLISI




## 2° FASE GLICOLISI

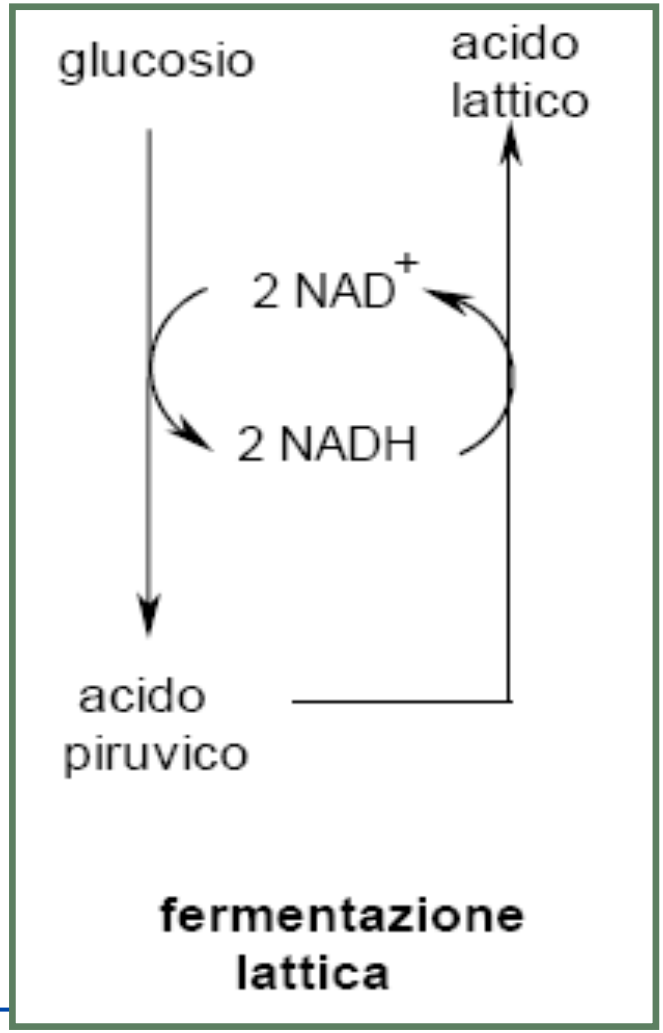



**Glucosio + 2ATP + 4ADP + 2Pi + 2NADox**

**2 acido piruvico + 2ADP + 4ATP + 2NADHrid + $2H_2O$**

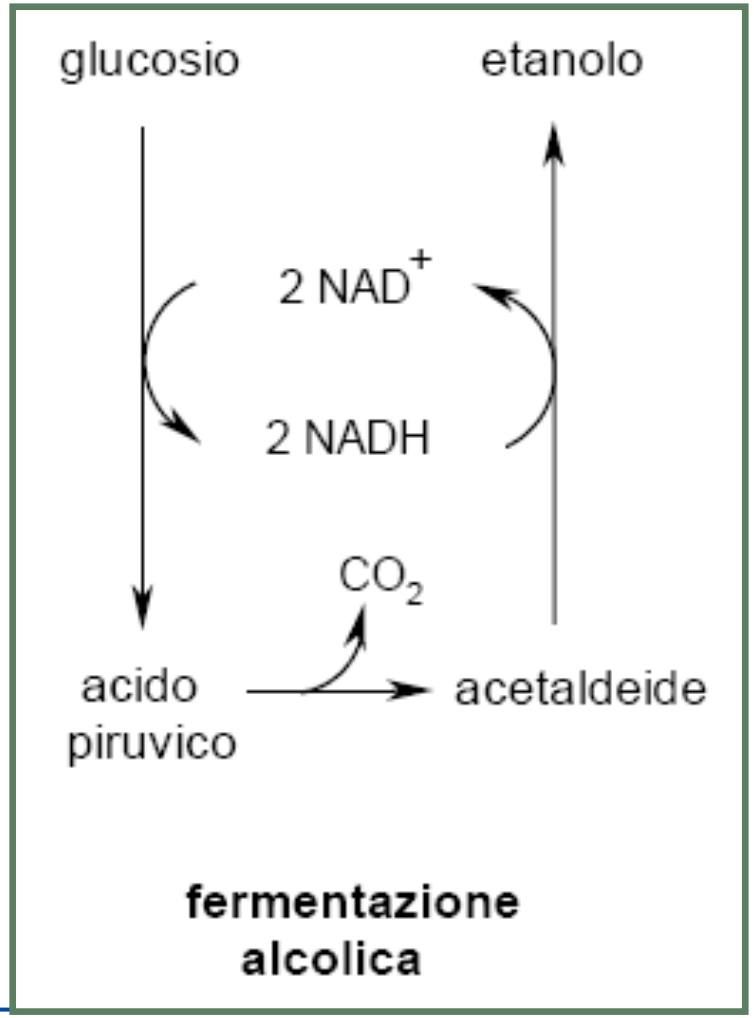
# Citoplasma – Fermentazione Lattica

- In condizioni anaerobiche, cioè in assenza di ossigeno, bisogna che qualche molecola funga da ossidante.
- L'**acido piruvico** può essere **ridotto ad acido lattico** per consentire l'ossidazione del NADH a NAD+.



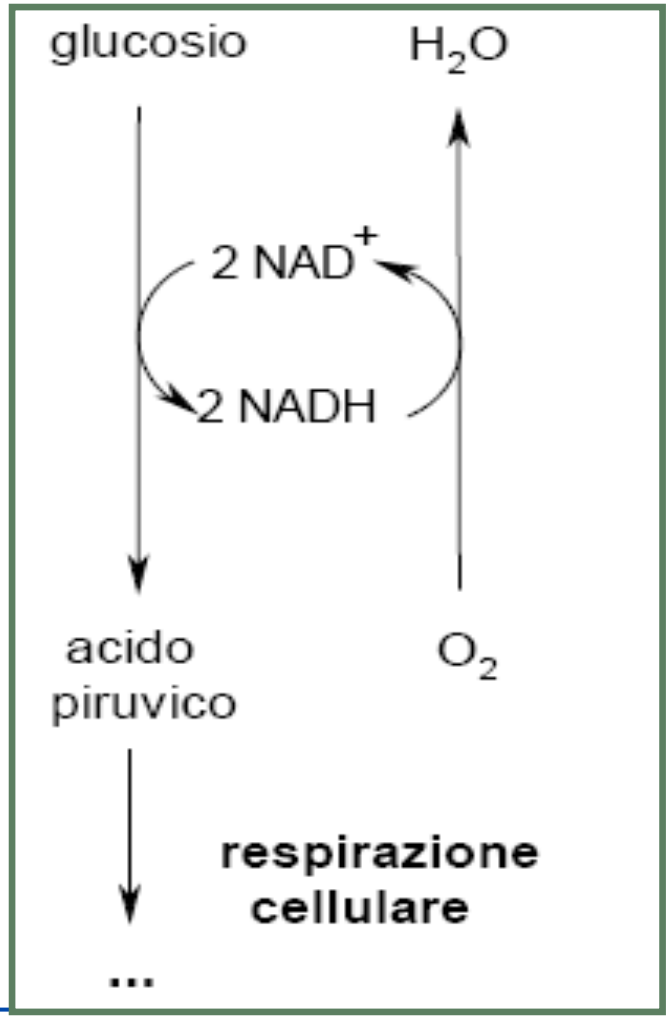
# Citoplasma – Fermentazione Alcolica

- In condizioni anaerobiche l'acido piruvico, il prodotto finale della glicolisi, può essere ridotto con una diversa via metabolica.
- **L'acido piruvico** viene decarbossilato ad **acetaldeide**, per poi essere ridotto ad **etanolo**. In questo modo viene ossidato il NADH a NAD+ e si può continuare la glicolisi.

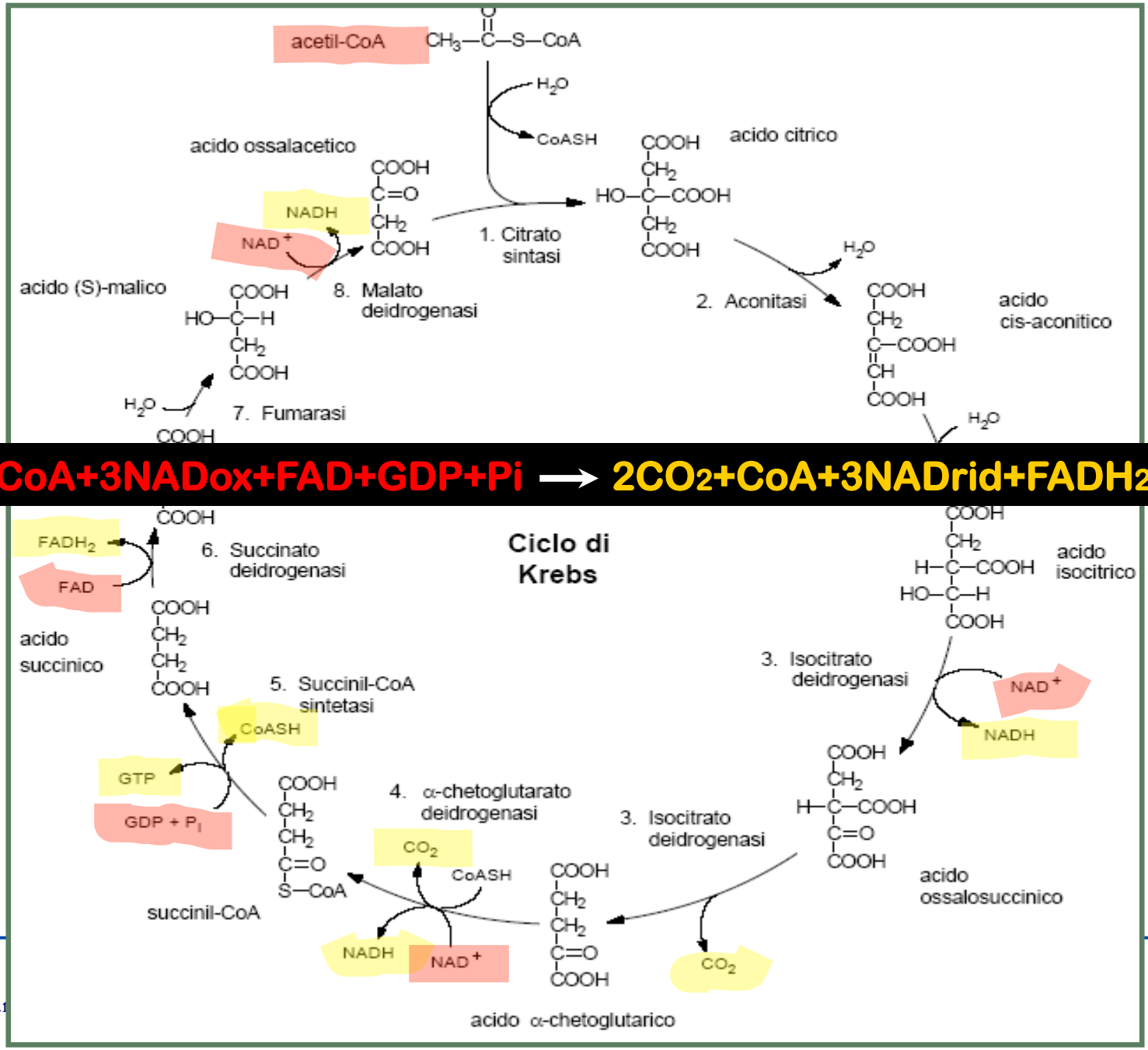


# Mitocondrio – Respirazione Cellulare

- In condizioni aerobiche la glicolisi costituisce solo il primo passo della respirazione cellulare cioè dell'Ossidazione del Glucosio a $CO_2$ ad opera di $O_2$ .
- Il NADH prodotto viene riossidato a NAD + e questo si realizza ad opera dell'ossigeno molecolare $O_2$ che si riduce ad $H_2O$ nella catena respiratoria.

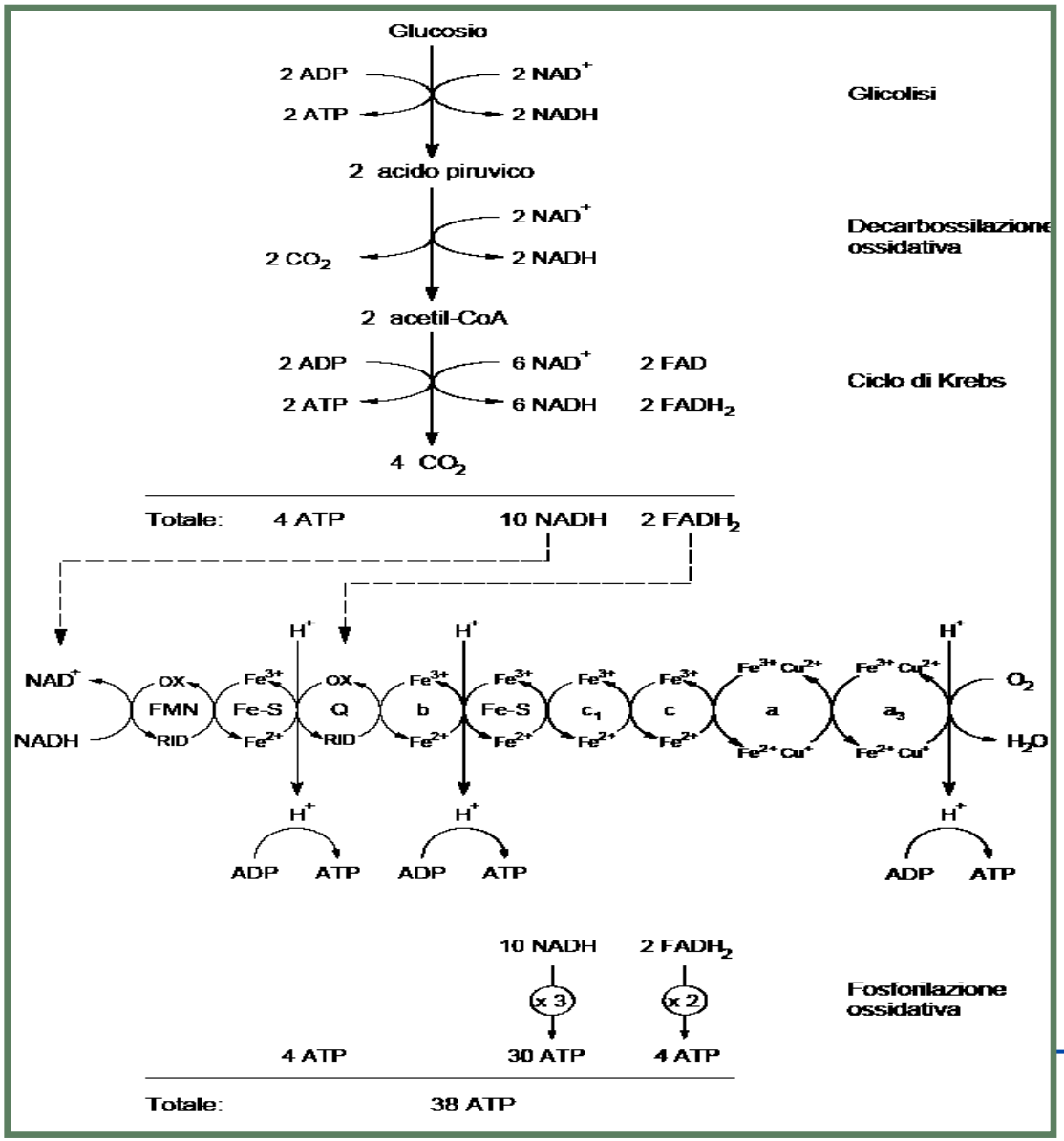


# Schema Ciclo di Krebs




**Acetil-CoA+3NADox+FAD+GDP+Pi ⟶ 2CO₂+CoA+3NADrid+FADH₂+GTP**

# Sintesi Processo




**Glicolisi:**

**2 ATP; 2 NADH**

**Decarbossilazione Ossidativa:**

**2 $CO_2$; 2 NADH**

**Ciclo di Krebs:**

**2 ATP; 6 NADH; 2 $FADH_2$**

~~**Tot: 4 ATP 10 NADH 2 $FADH_2$**~~

**Catena Respiratoria:**

- **1 NADH → 3 ATP**
- **1 $FADH_2$ → 2 ATP**

**(10 NADH*3) (2 FADH*2)**

**4 ATP 30 ATP 4 ATP**

~~**Tot: 38 ATP**~~

# Approccio ad Agenti

## Quando un sistema si presta maggiormente ad essere modellato con un approccio Agent-Oriented? (1/2)

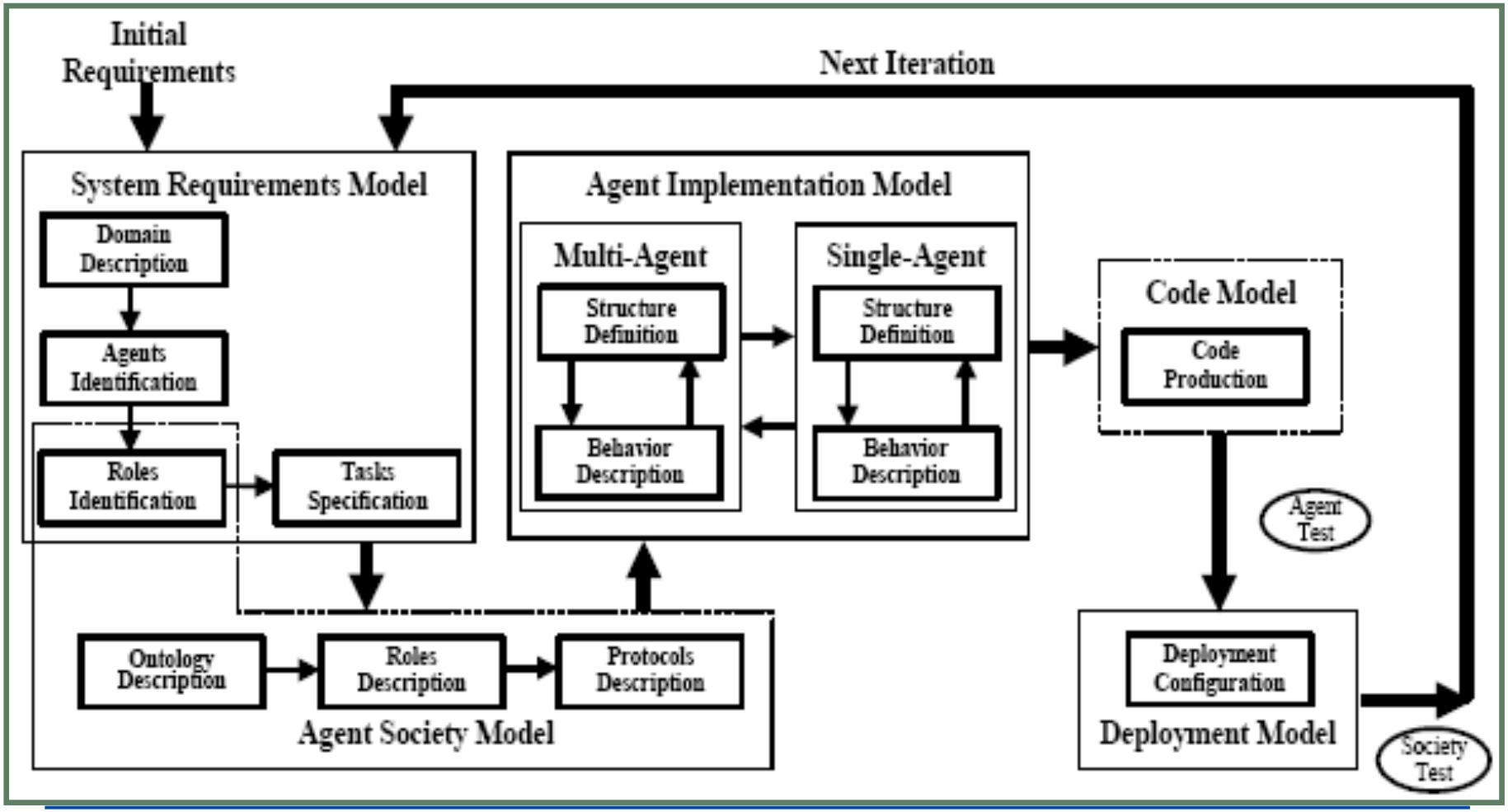
- In situazioni in cui sono richieste tipi di comunicazione complessa/ diversa.
- Quando il sistema deve rapportarsi bene in situazioni in cui non è possibile specificare il suo Behavior in ogni caso.
- In situazioni che coinvolgono la negoziazione, la cooperazione e la competizione tra diverse entità.
- In situazioni in cui il sistema deve agire autonomamente.
- Quando il sistema potrebbe espandersi o essere modificato inaspettatamente.

# Approccio ad Agenti

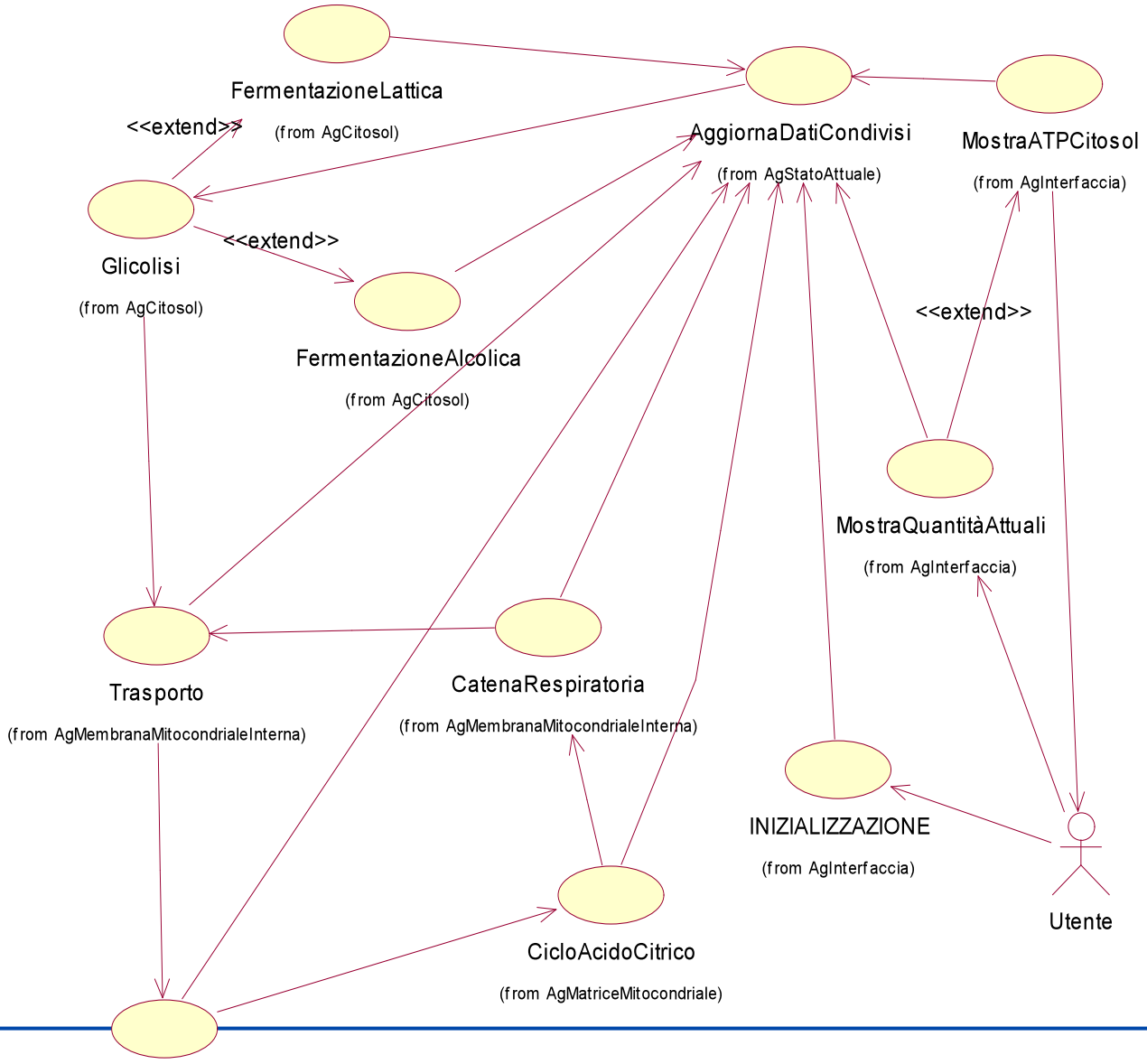
- Realtà complessa come le Scienze Biologiche, in particolare lo studio dei *processi inter e intra cellulari*.
- In un processo come la "Produzione dell'ATP nella cellula a partire dai Carboidrati", ci siamo basati sul paradigma ad agenti perché:
  - ✓ Sono capaci di Azioni AUTONOME e FLESSIBILI;
  - ✓ Sono situate in un dominio APERTO, DINAMICO e NON PIENAMENTE CONOSCIUTO;
  - ✓ Sono situate in un dominio in cui devono obbligatoriamente INTERAGIRE con altri agenti → MAS;
  - ✓ La complessità dei processi cellulari richiede un paradigma di coordinazione e collaborazione come quello ad Agenti.

# Metodologia Utilizzata

## Schematizzazione della metodologia PASSI



# Fase 1: Domain Description



# Fase 1: Domain Description

## I 5 AGENTI DEL SISTEMA

1. "**AgCitosol**": corrisponde al citoplasma della cellula al cui interno avvengono i processi di Fermentazione Lattica, Glicolisi e Fermentazione Alcolica.

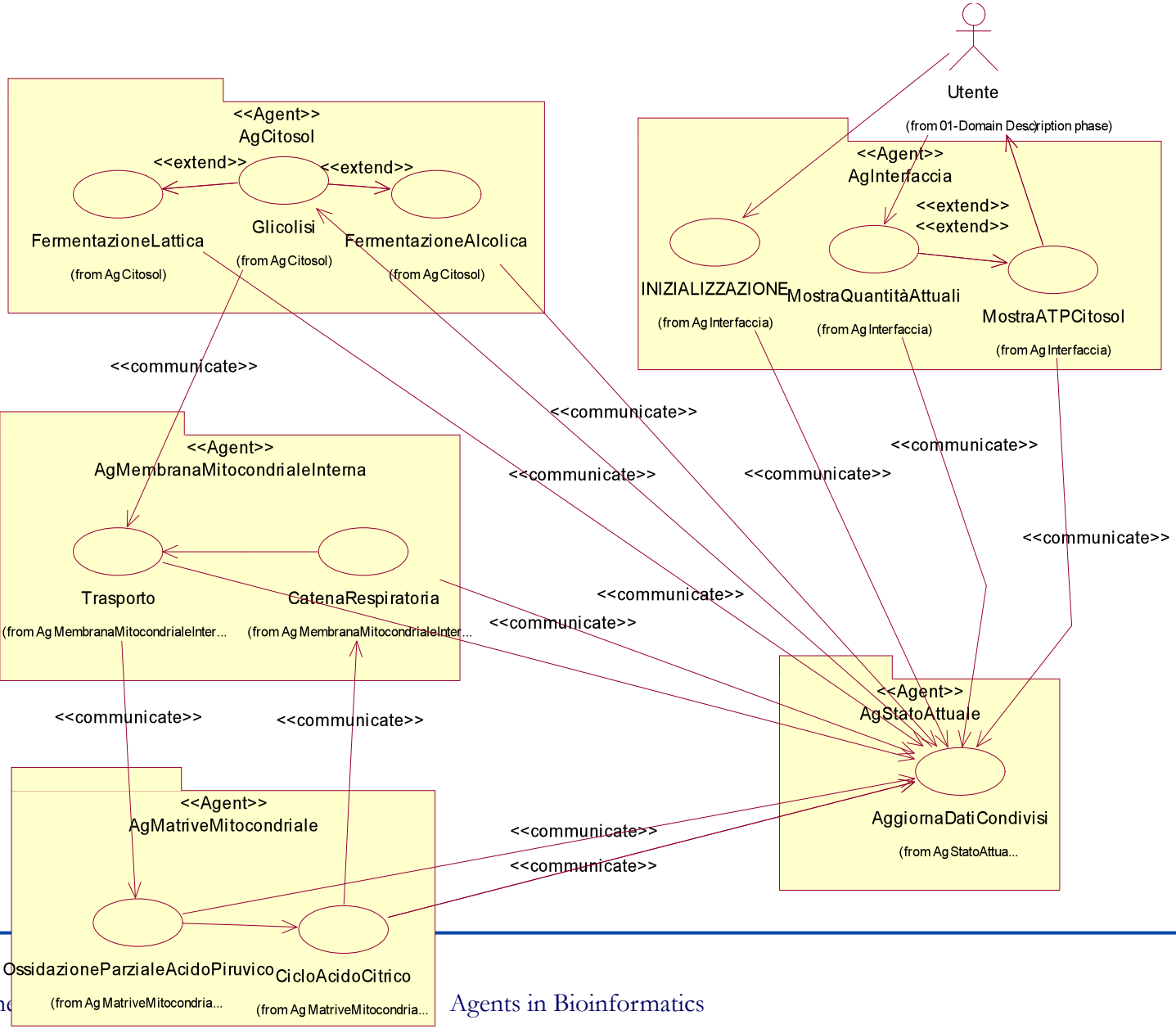
2. "**AgMembranaMitocondrialeInterna**": corrisponde ad una parte del Mitocondrio in cui avvengono i processi del Trasporto del Piruvato in input, Trasporto di elettroni e Catena respiratoria.

3. "**AgMatriceMitocondriale**": corrisponde ad un'altra parte del Mitocondrio in cui avvengono i processi di Ossidazione parziale dell'acido piruvico (decarbossilazione ossidativa) e Ciclo dell'acido citrico (Ciclo di Krebs).

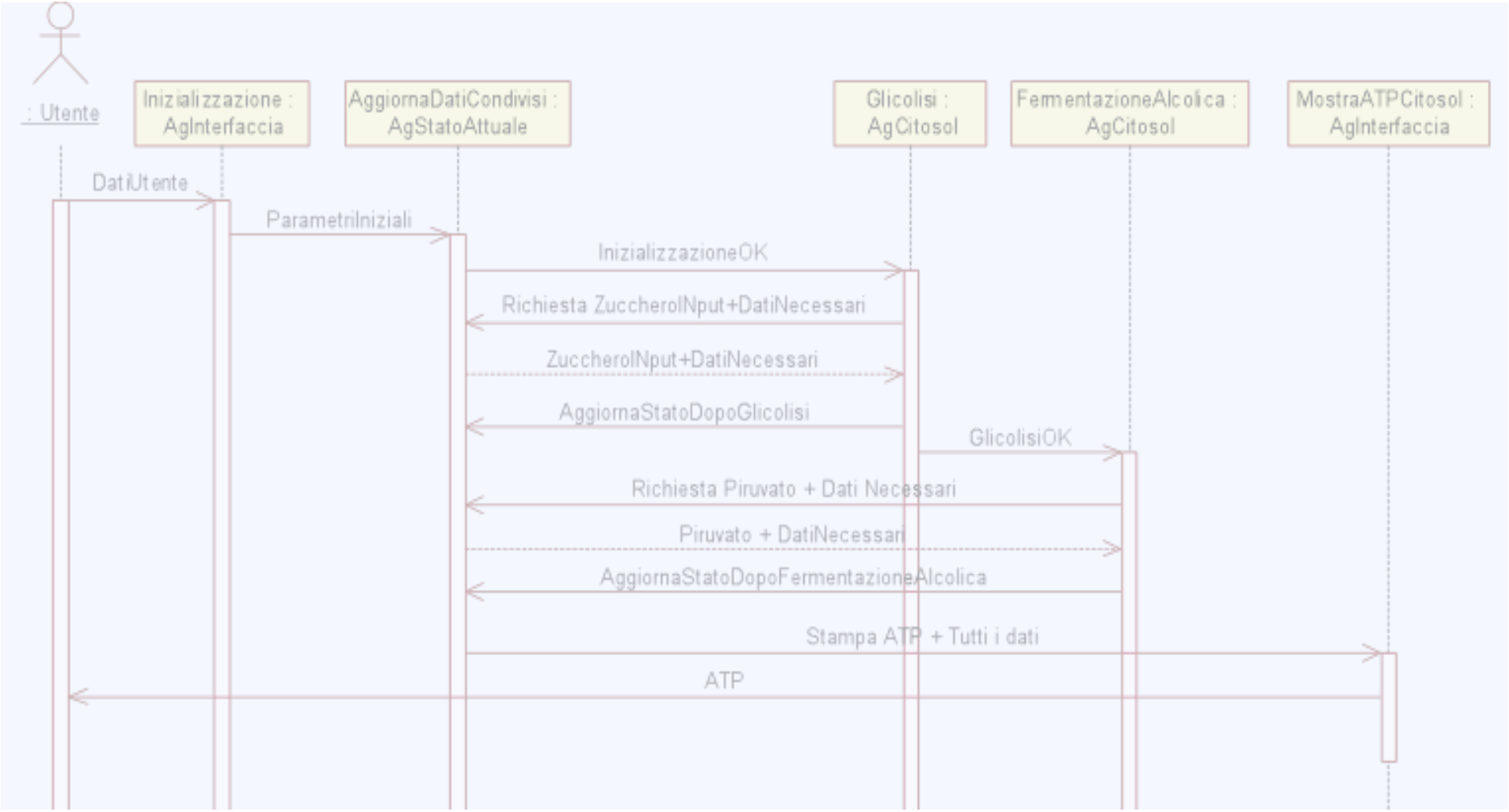
4. "**AgInterfaccia**": corrisponde all'Interfaccia del sistema a cui accede l'attore Utente.

5. "**AgStatoAttuale**": corrisponde all'ambiente, è il "*tuple space*" del sistema. Tutti i processi cellulari che producono e utilizzano sostanze comunicano con questo agente.

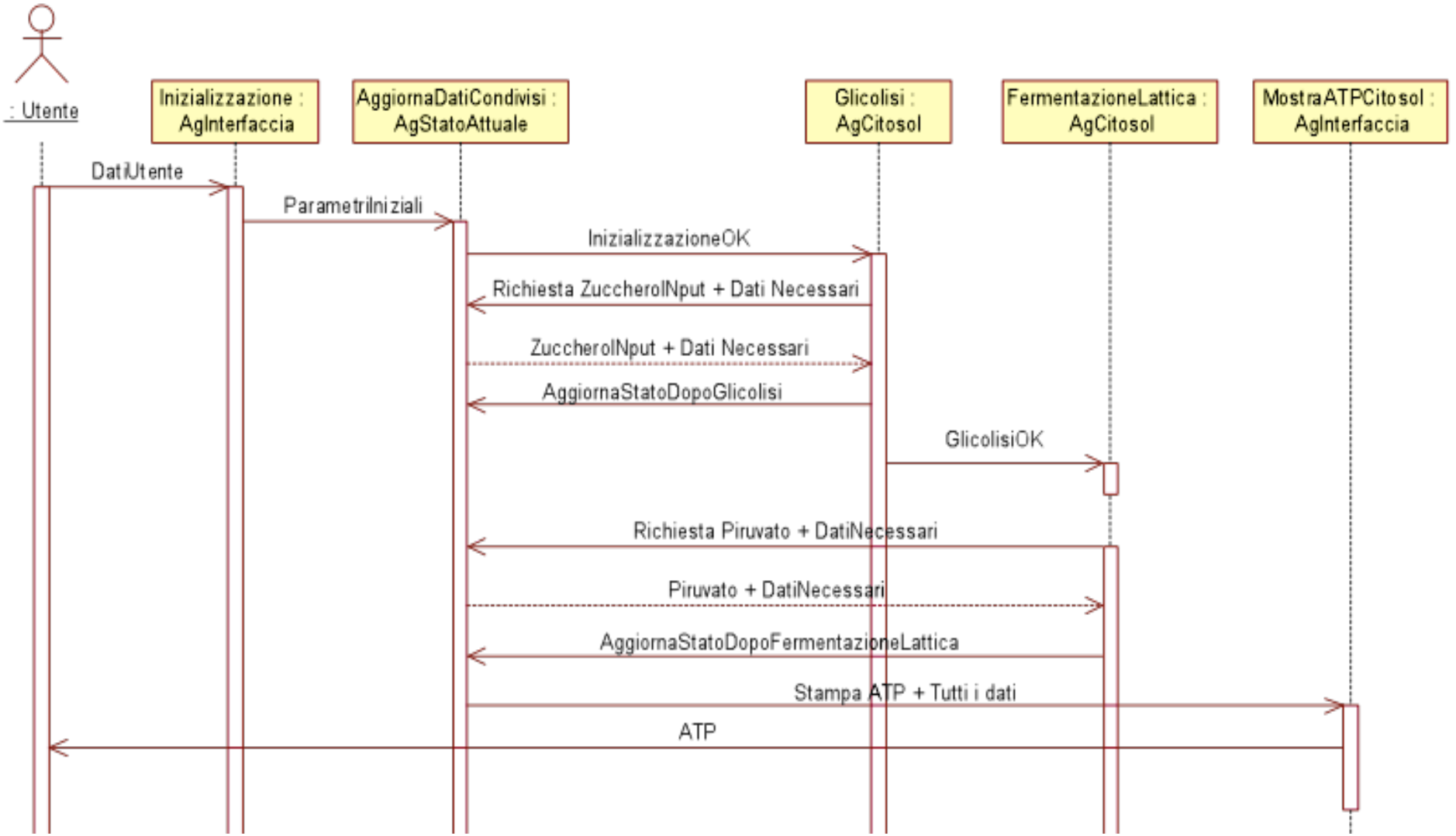
# Fase 2: Agent Identification



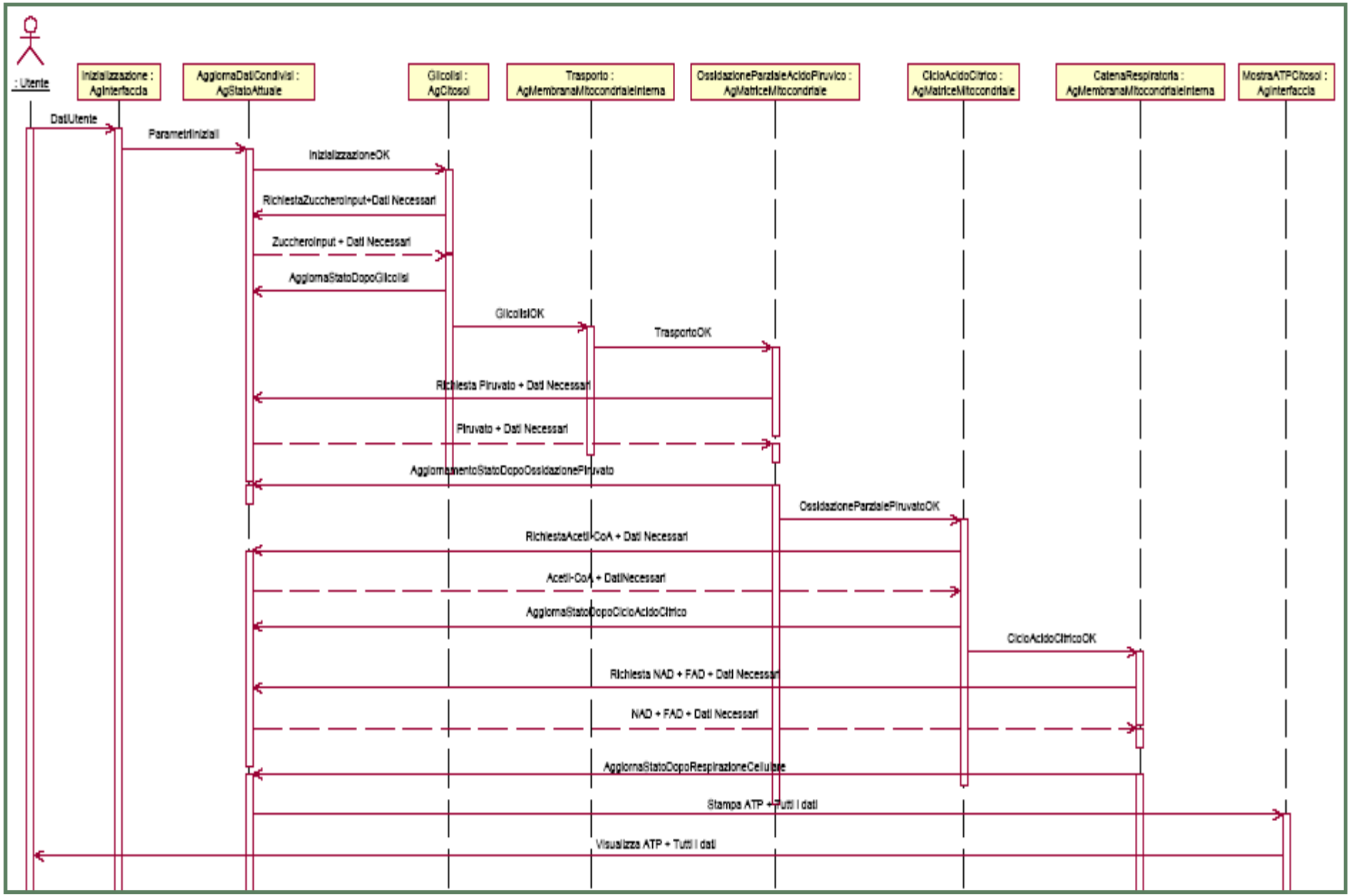
# Fase 3: Role Identification – Fermentazione Alcolica



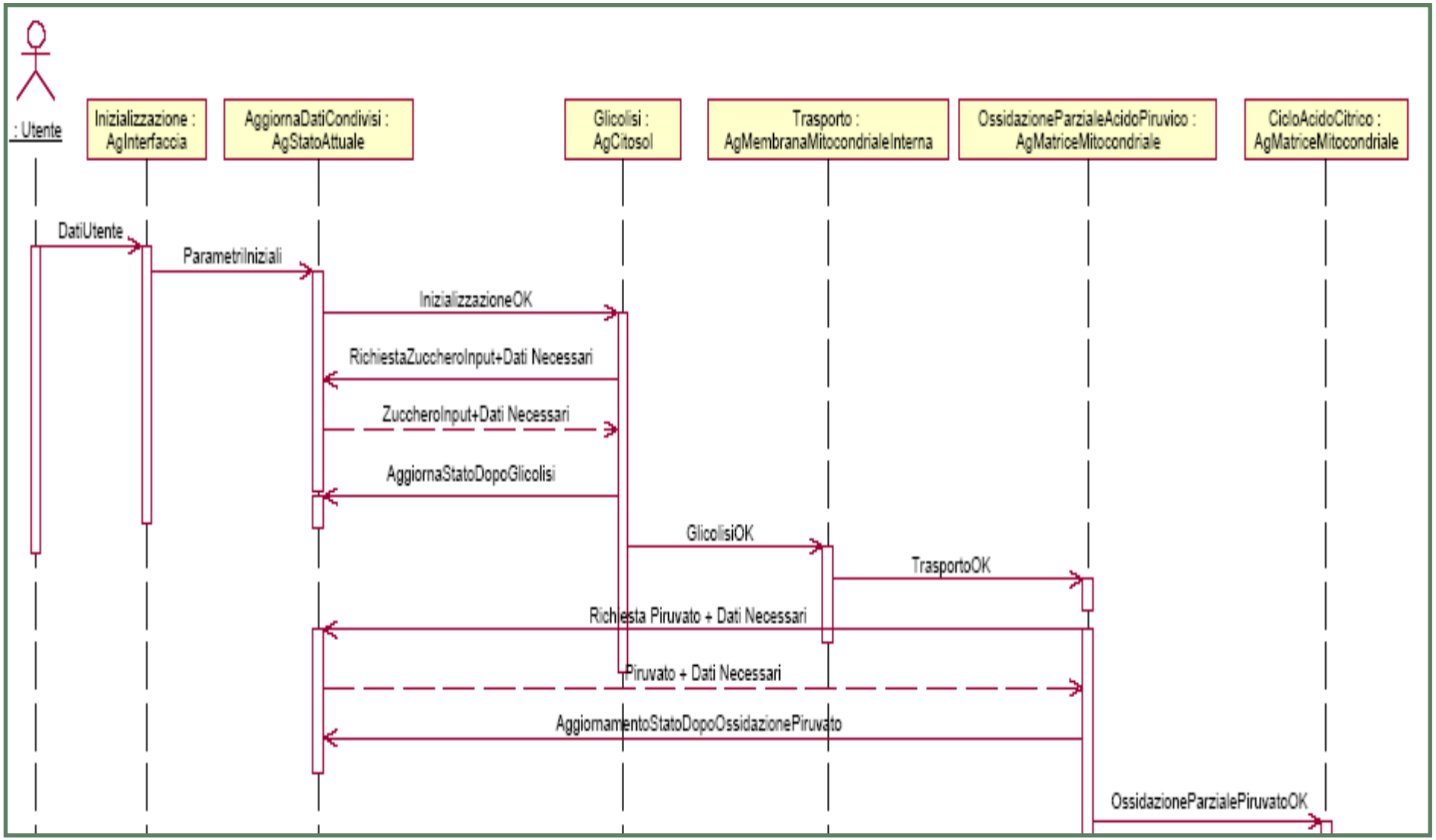
# Fase 3: Role Identification – Fermentazione Lattica



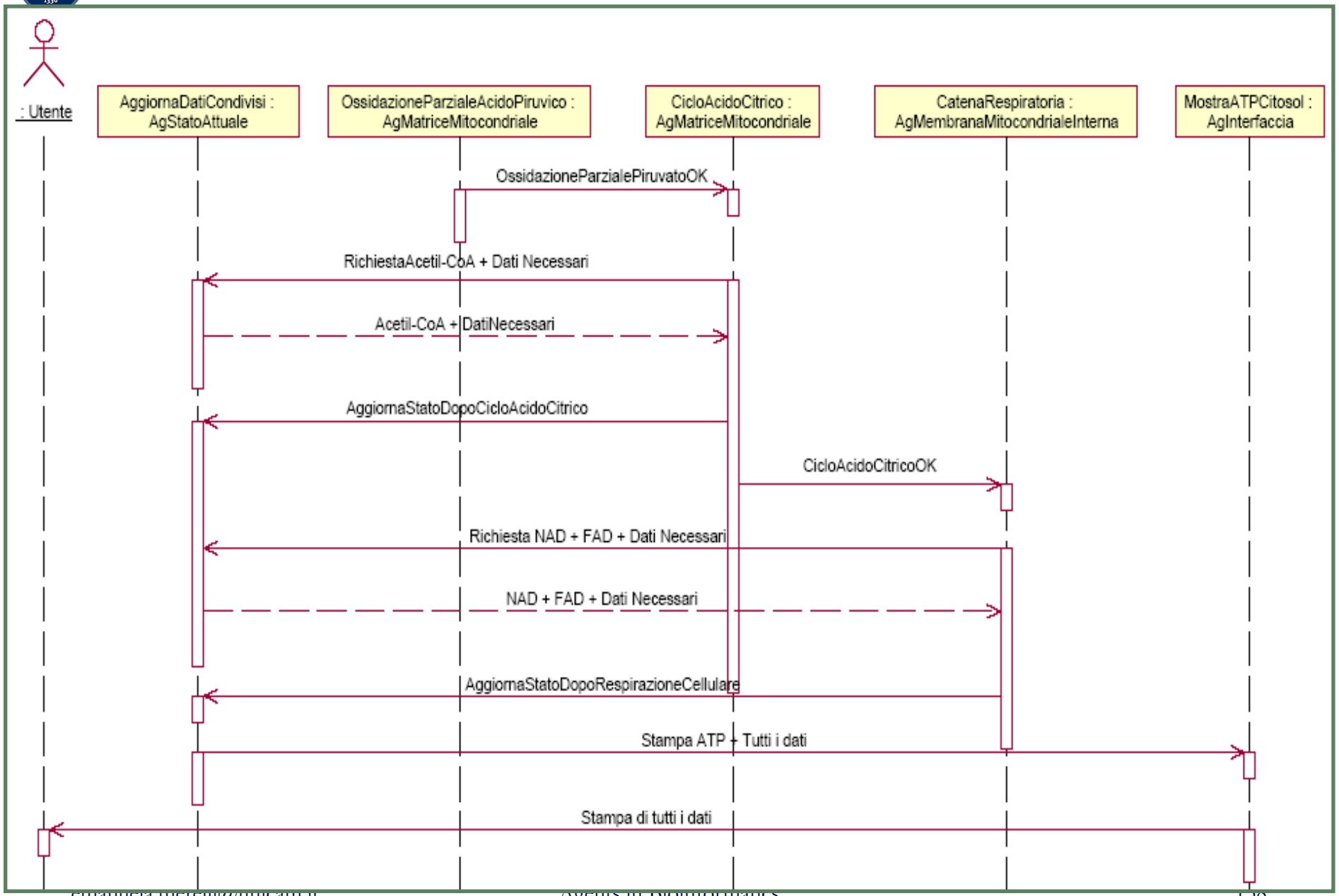
# Fase 3: Role Identification – Via Aerobica Completa

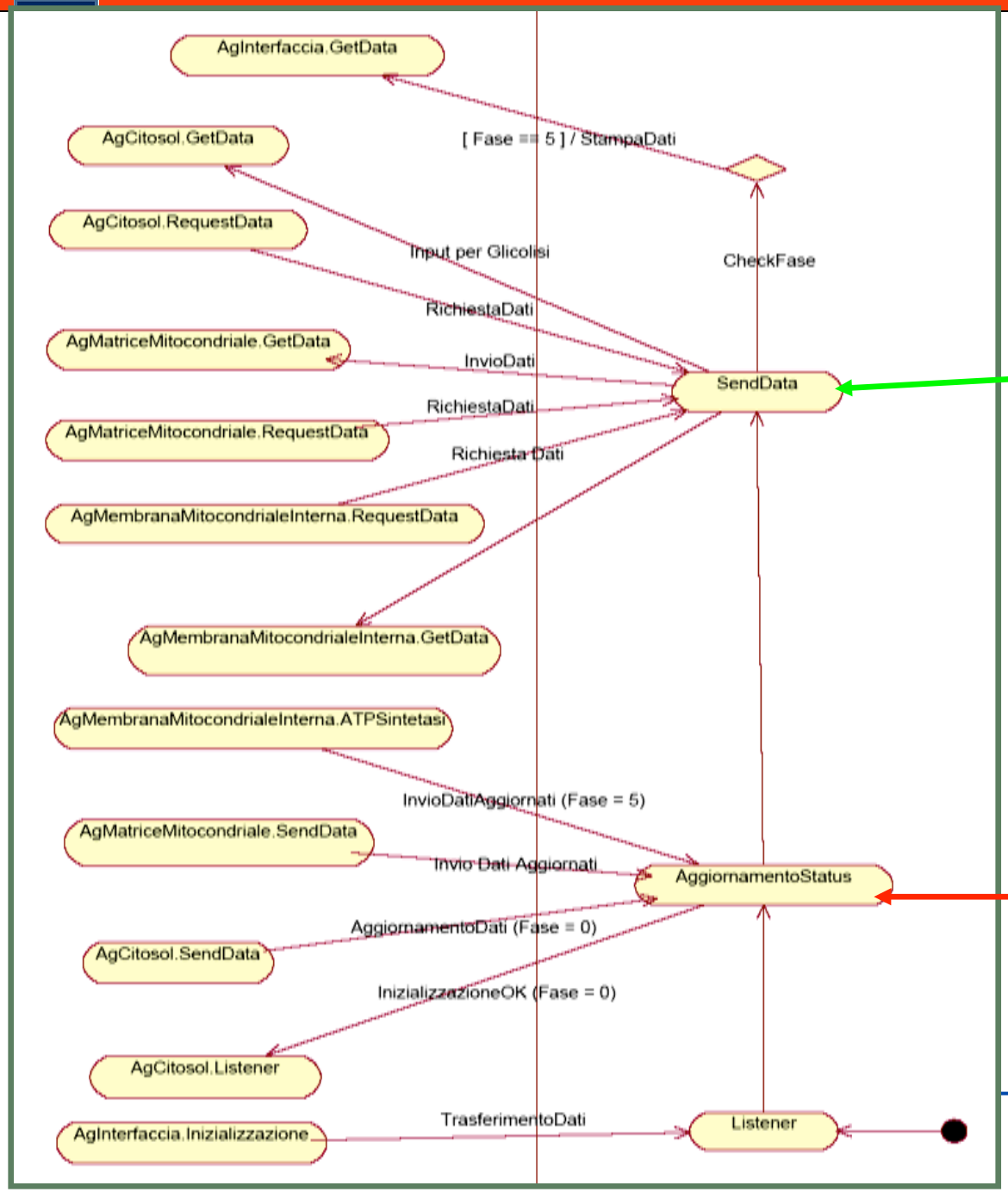


# Fase 3: Role Identification – Via Aerobica - 1



# Fase 3: Role Identification – Via Aerobica - 2





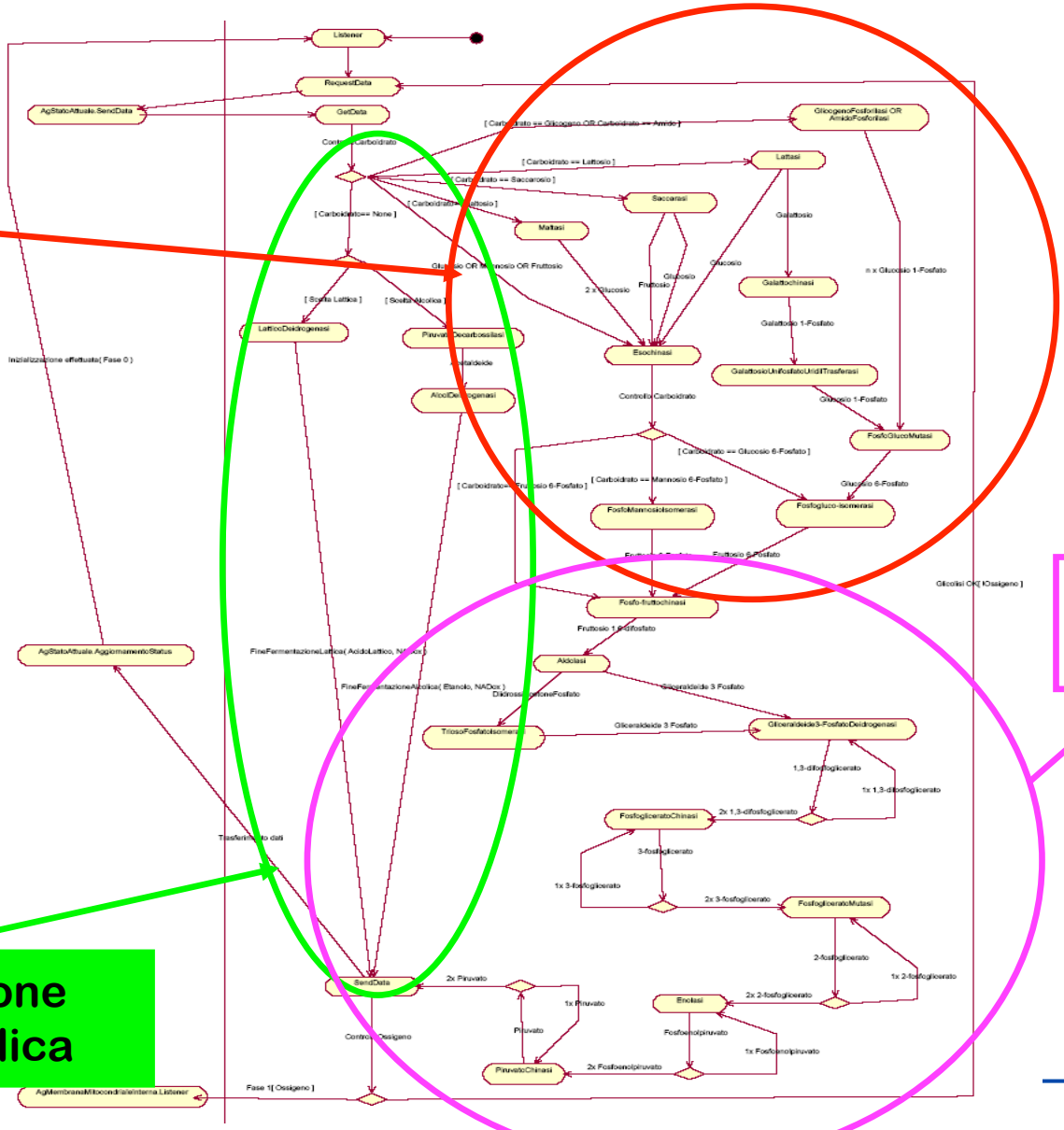

# Fase 4: Task Specification – AgStatoAttuale

**SendData** risponde alle richieste degli UserAgent dei dati necessari ai loro processi; In base alla richiesta ricevuta invierà i relativi dati aggiornati

**AggiornamentoStatus** attende i dati alla fine di ogni processo e aggiorna lo stato attuale dell'ambiente

# Fase 4: Task Specification – Agente Cytosol

**I° Fase Glicolisi + Controllo Carboidrato**



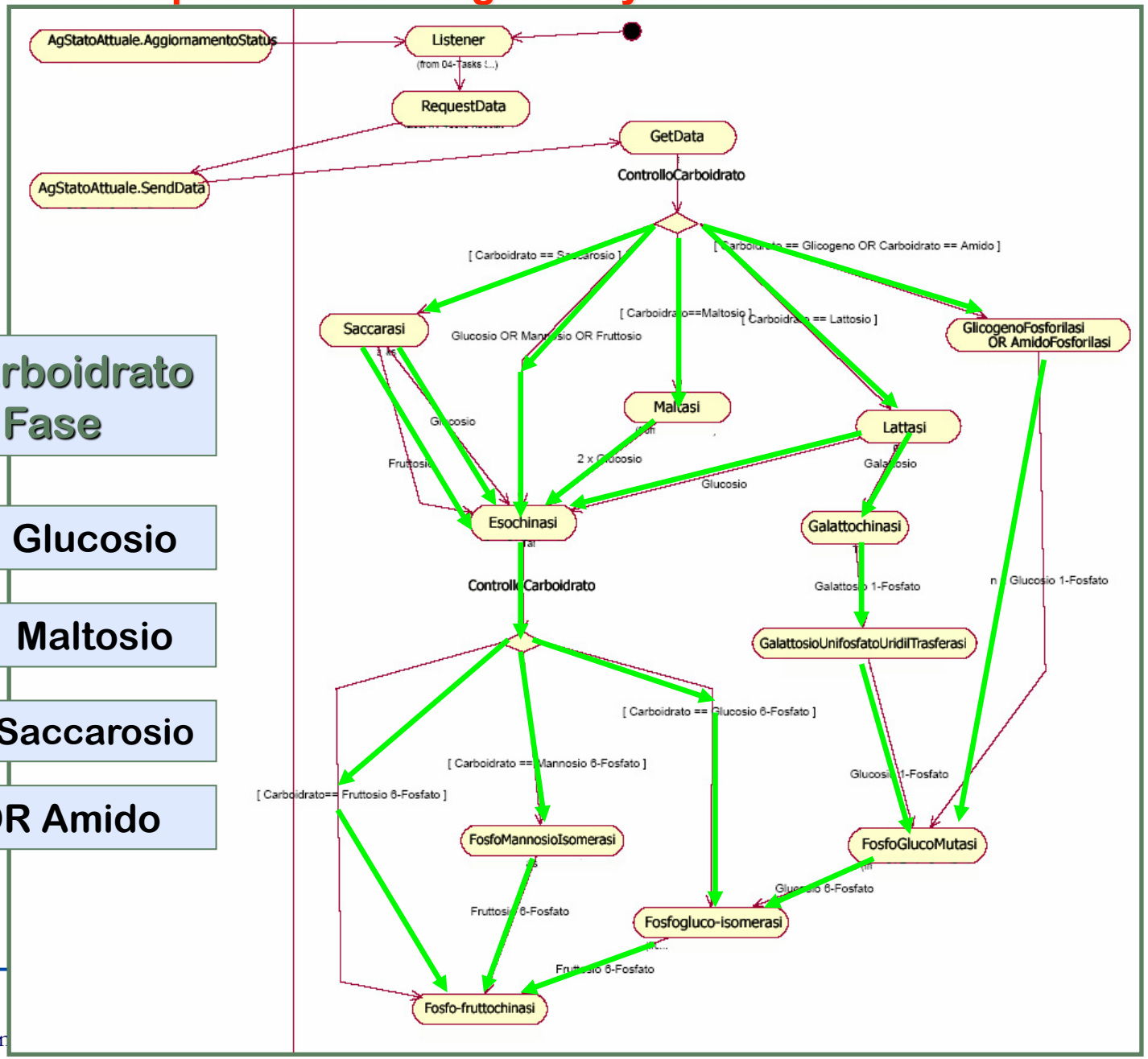

**II° Fase Glicolisi**

**Fermentazione Lattica/Alcolica**

# Fase 4: Task Specification – Agente Cytosol – Glicolisi –

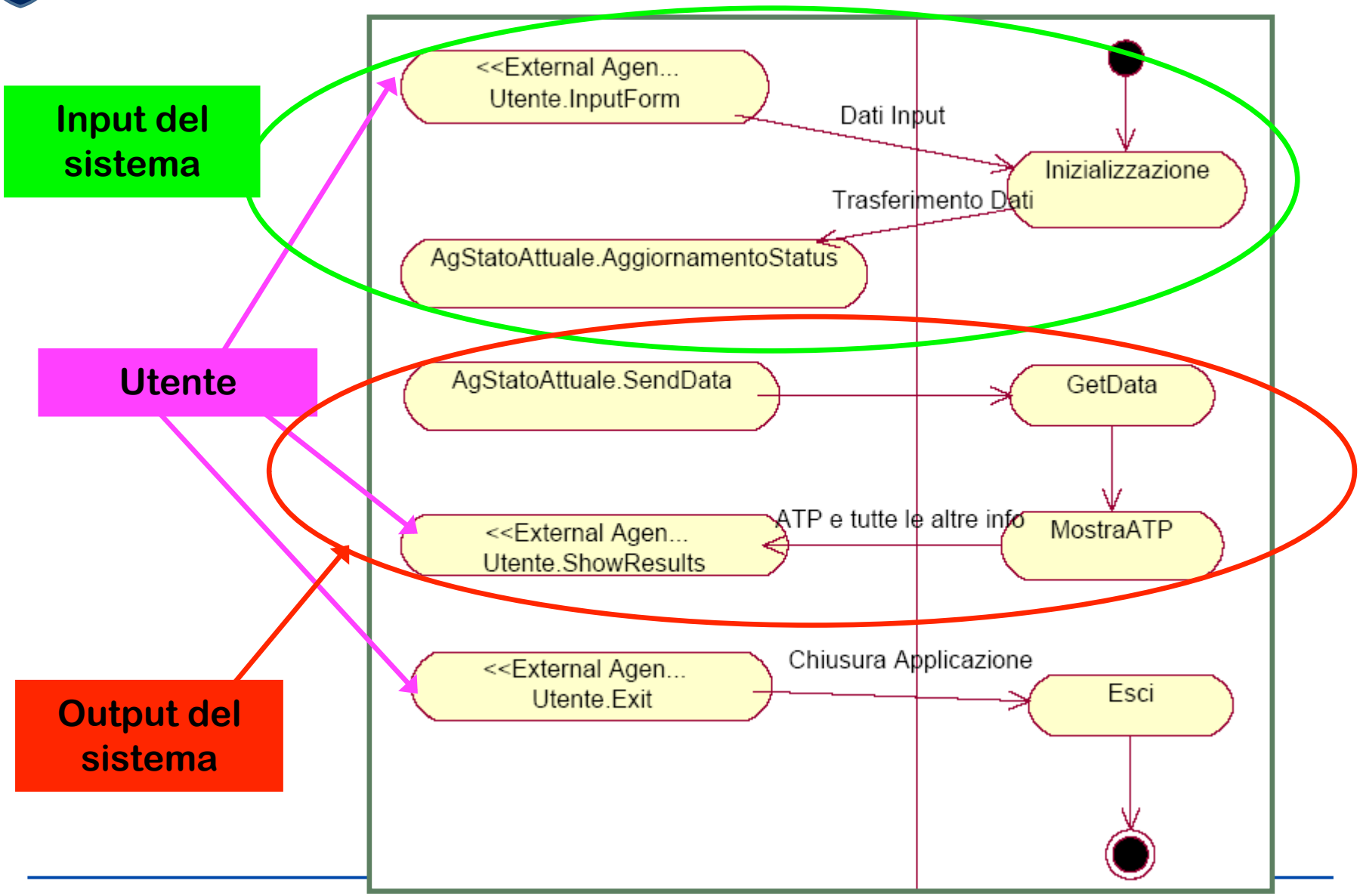
**Controllo Carboidrato + Prima Fase**

| | |
|---|---|
| Fruttosio | Glucosio |
| Mannosio | Maltosio |
| Lattosio | Saccarosio |
| Glicogeno OR Amido | |

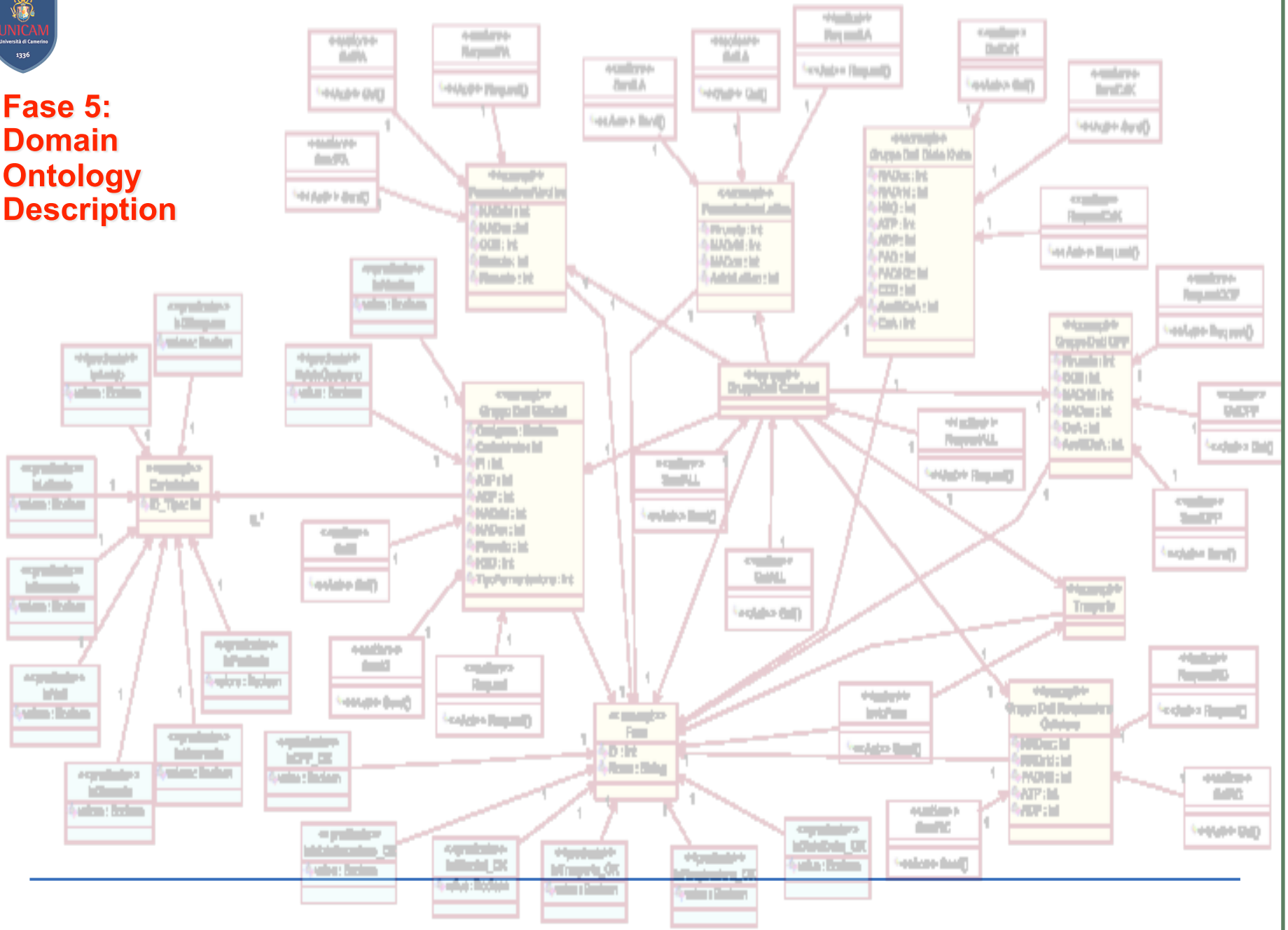



emanuela.merelli@unicam

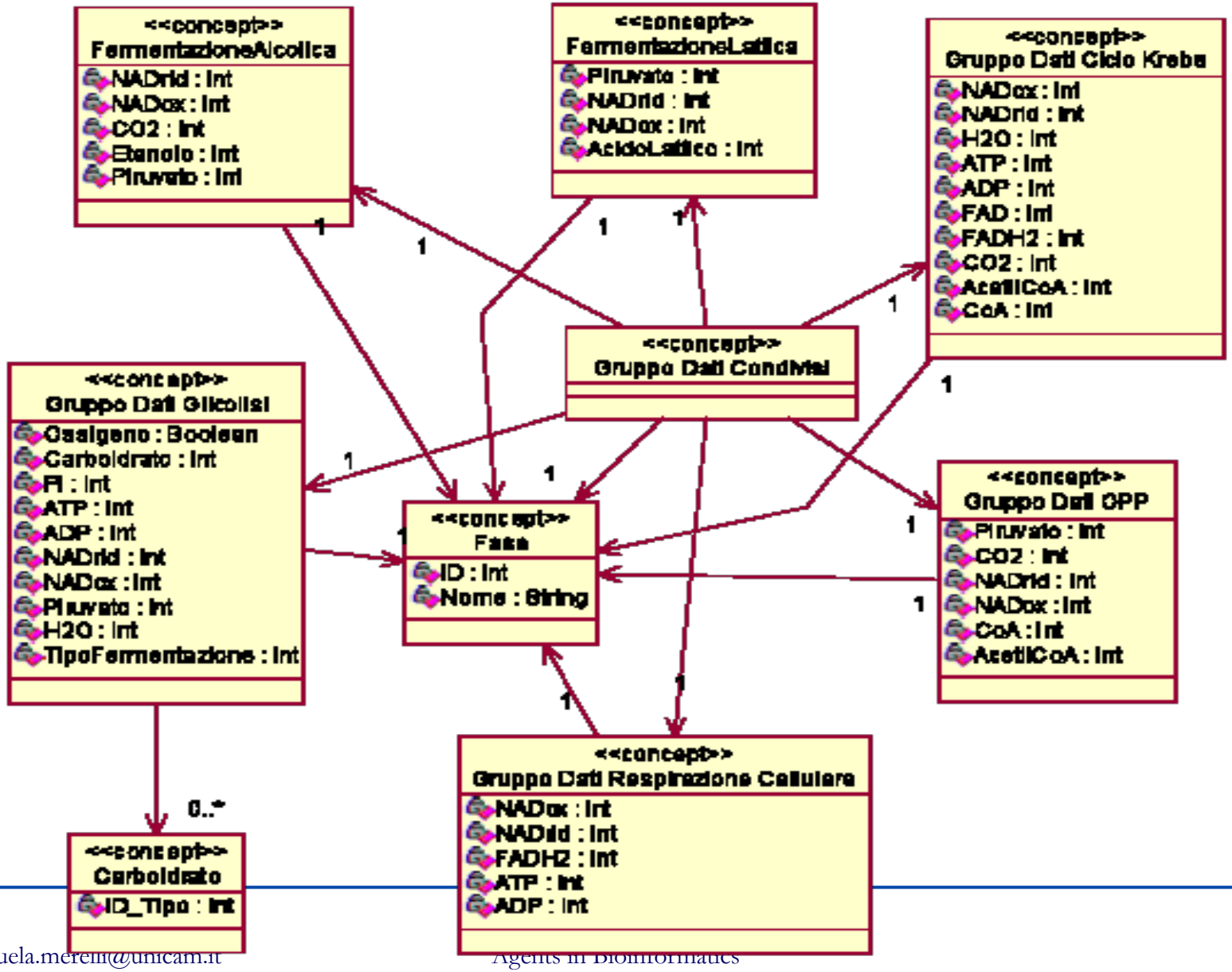
# Fase 4: Task Specification – AgInterfaccia



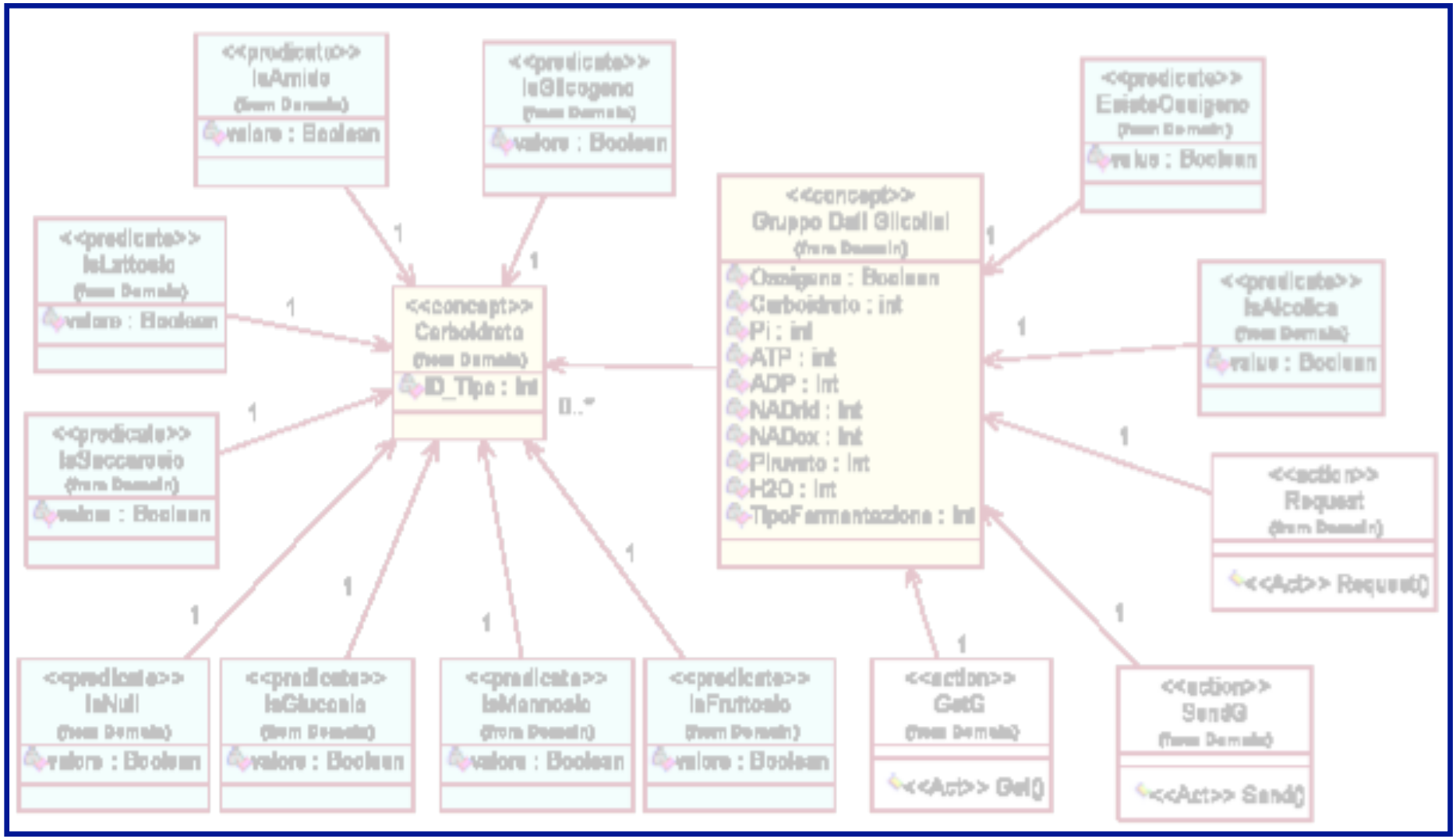
# Fase 5: Domain Ontology Description



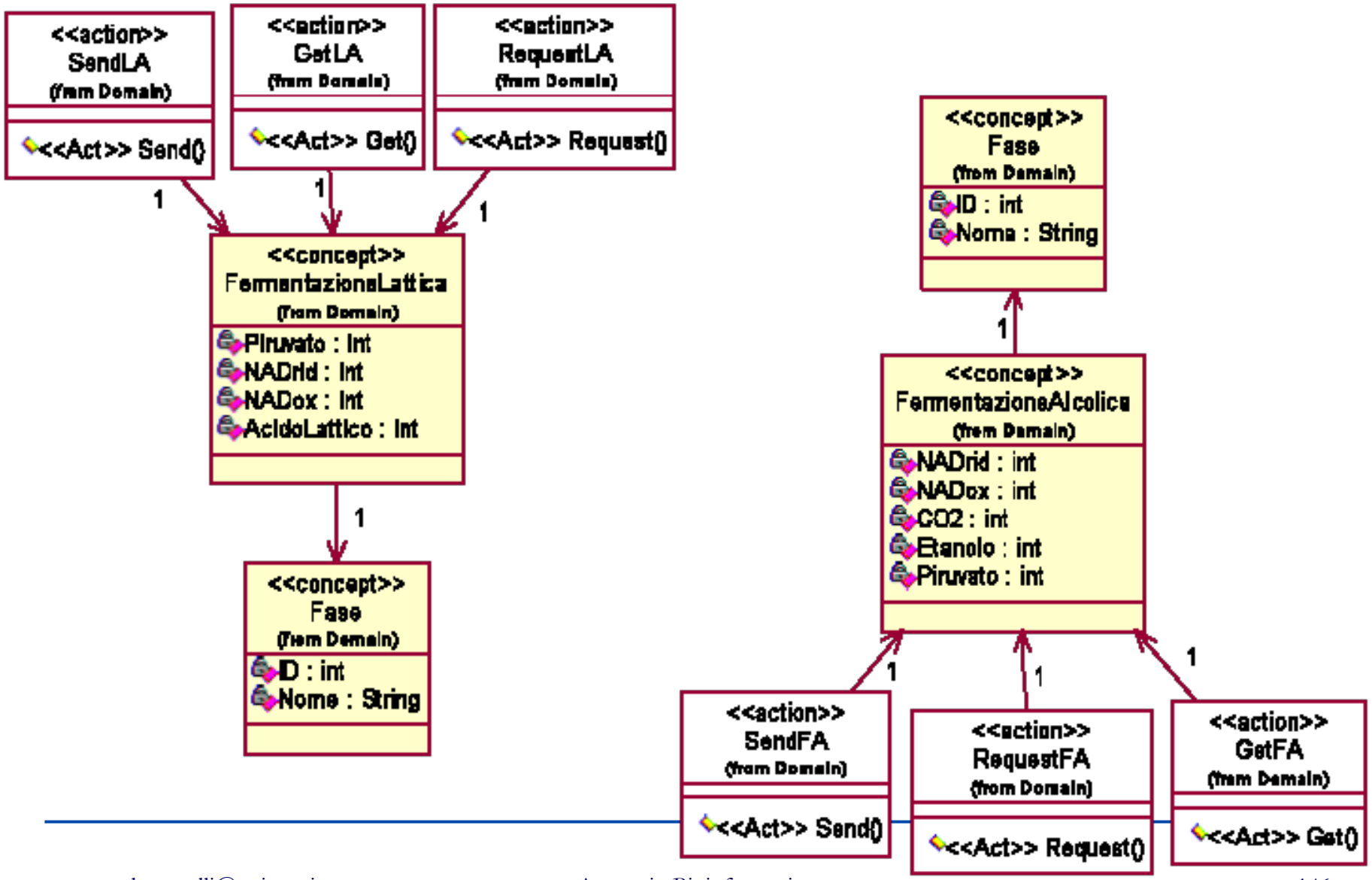
# Fase 5: Domain Ontology Description – Concetti principali



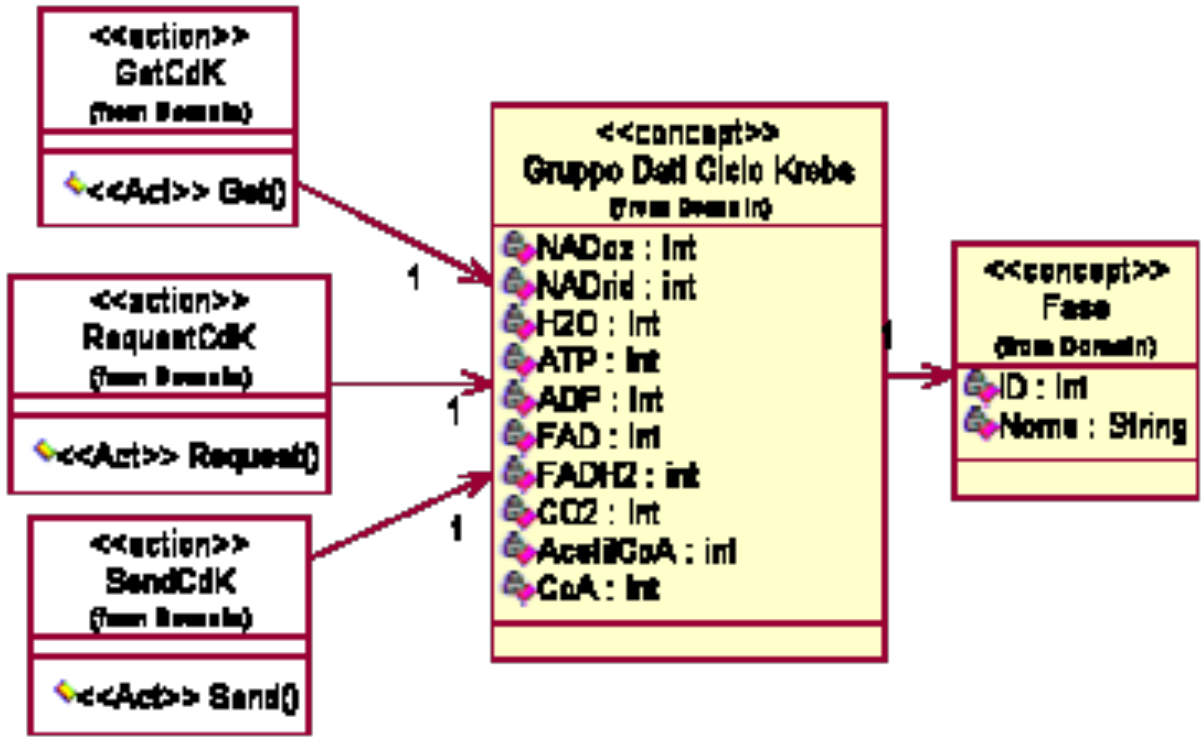
# Fase 5: Domain Ontology Description – Gruppo dati Glicolisi



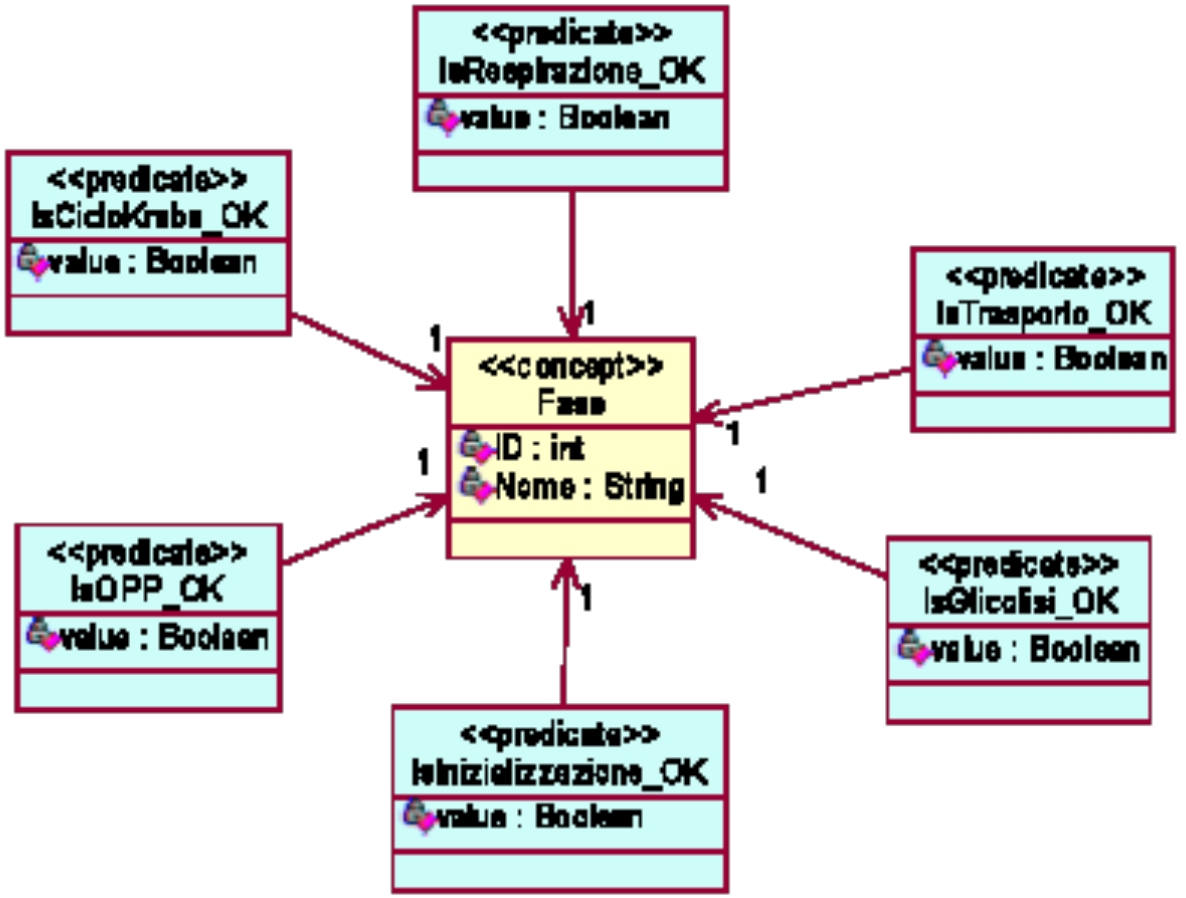
# Fase 5: Domain Ontology Description – Fermentazione

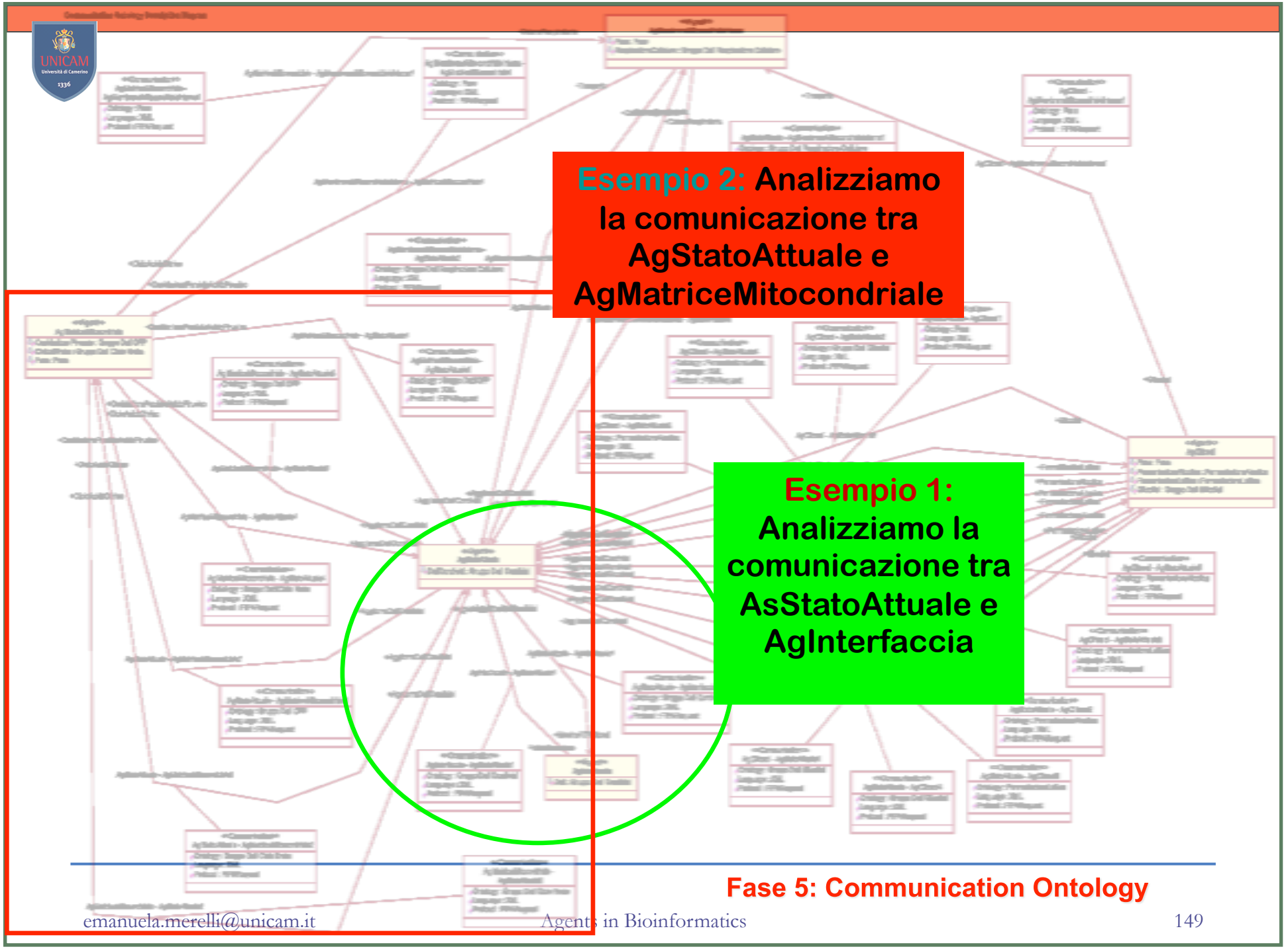


# Fase 5: Domain Ontology Description – Gruppo dati Ciclo di Krebs



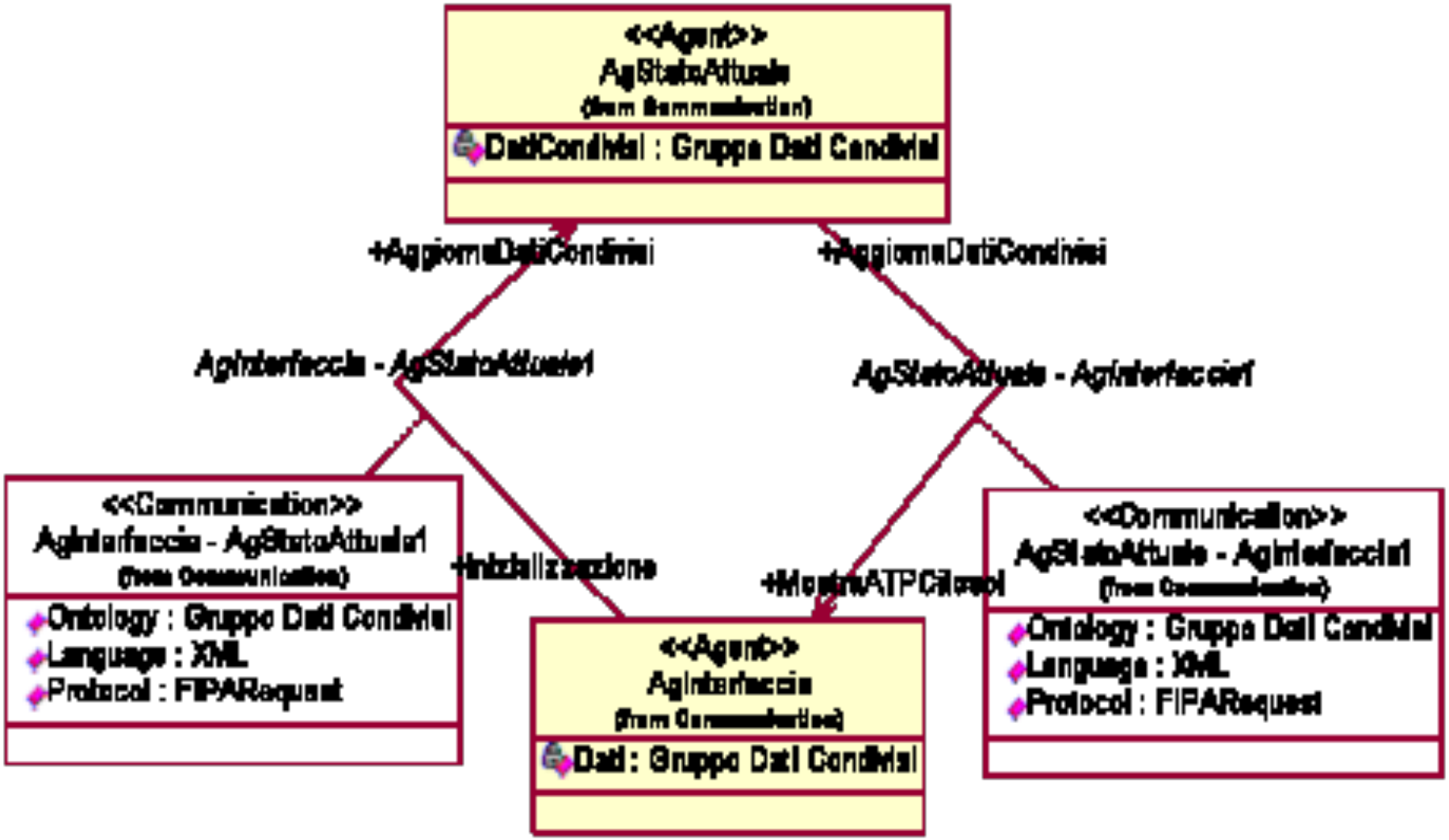
# Fase 5: Domain Ontology Description – Fase



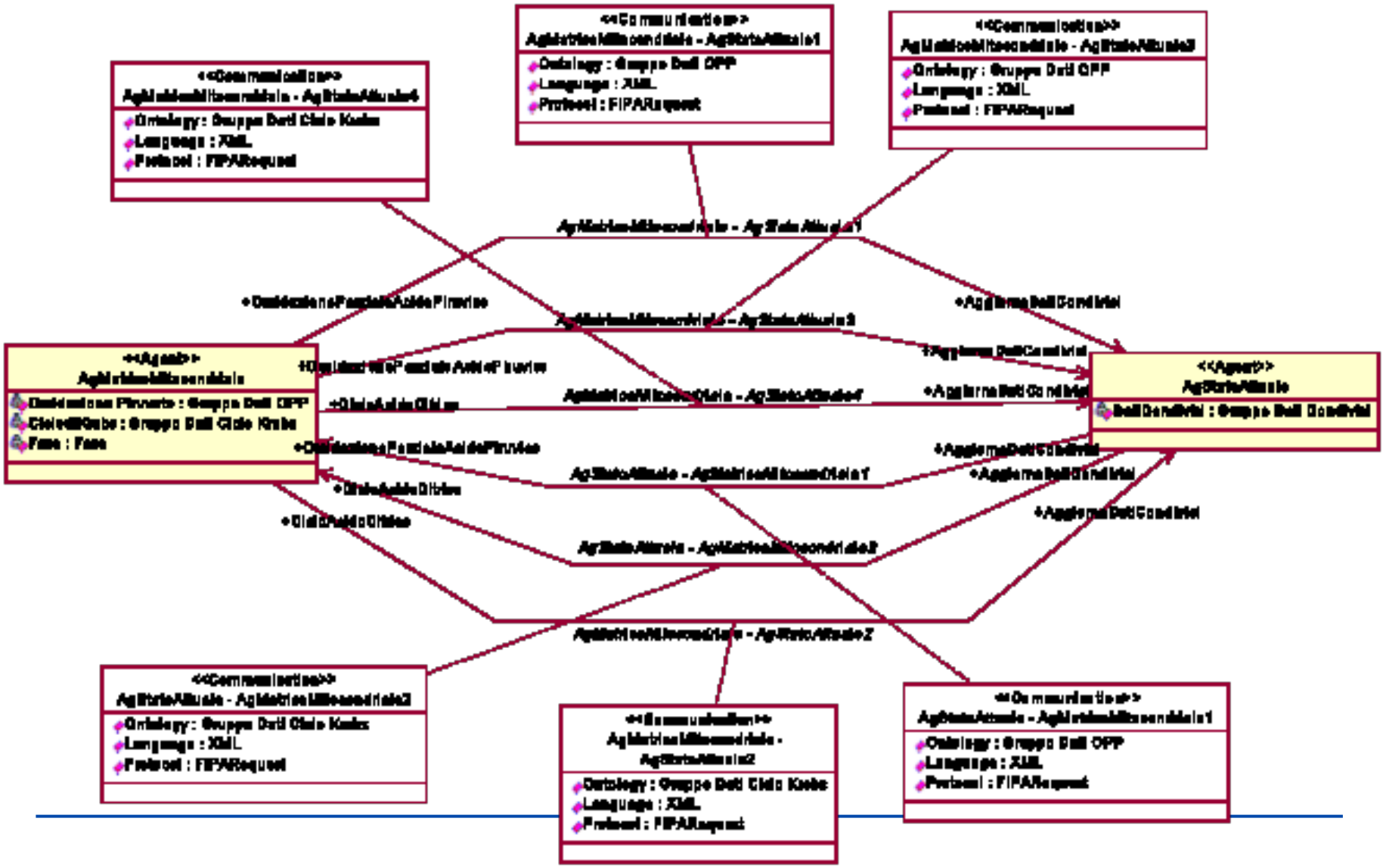



Fase 5: Communication Ontology

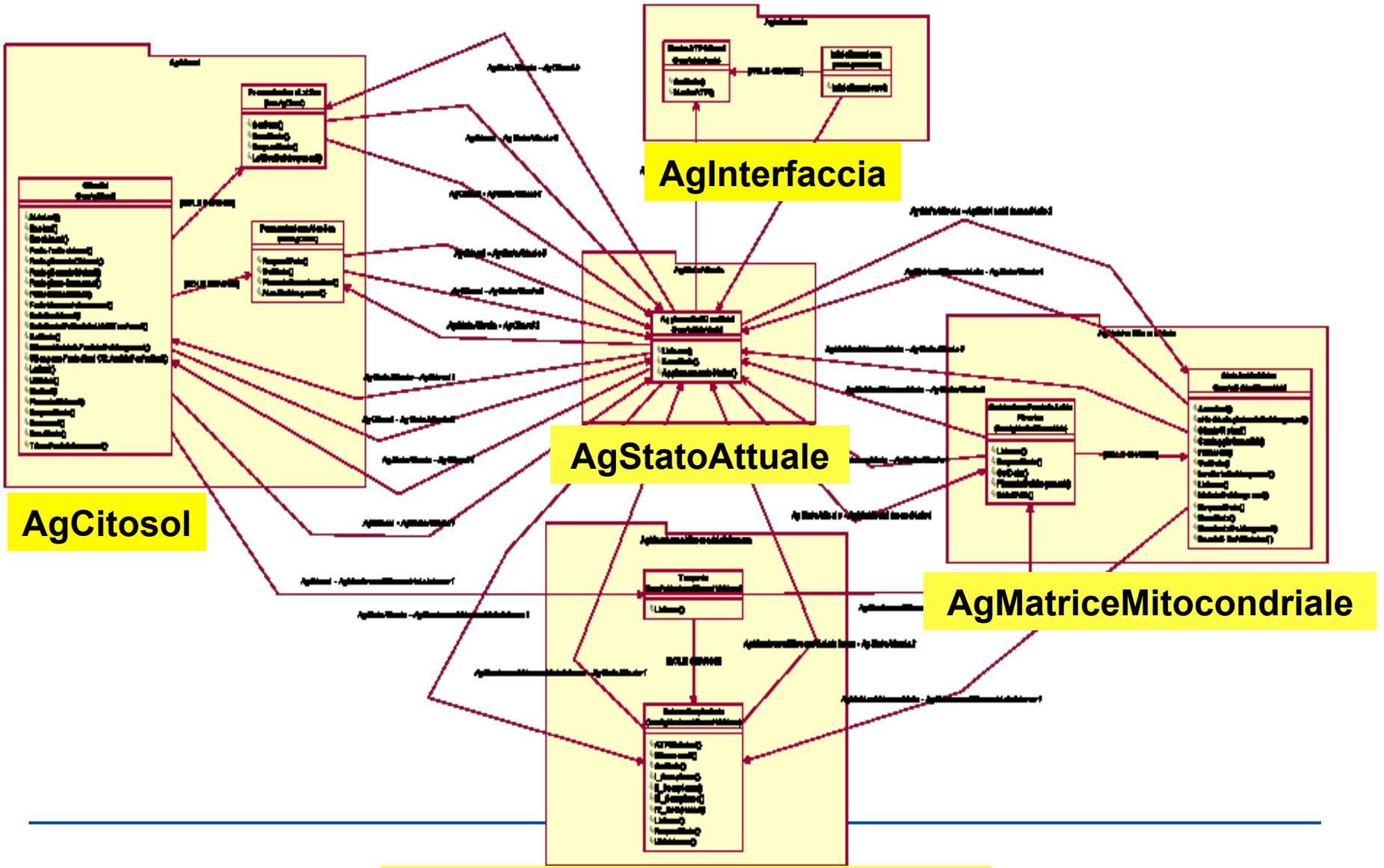
# Fase 5: Communication Ontology – Esempio 1



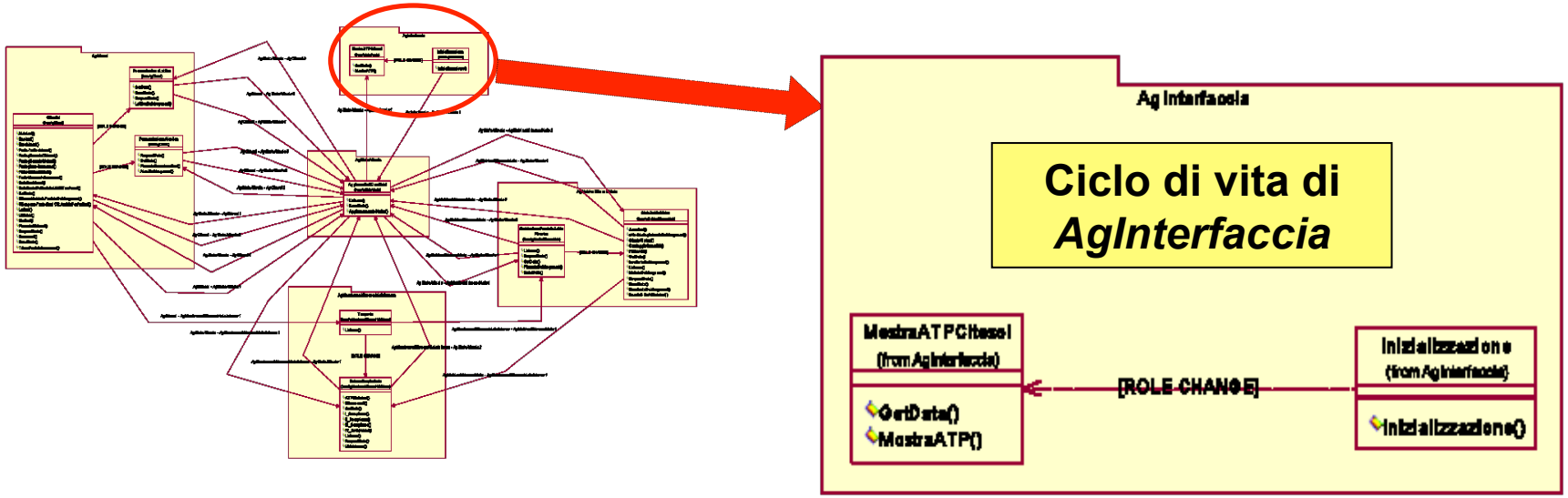
# Fase 5: Communication Ontology – Esempio 2



# Fase 6: Roles Description



# Fase 6: Roles Description




## *Ruoli:*

- **Inizializzatore**
- **Visualizzatore Risultati (MostraATPCitosol)**

## *Comunicazioni:*

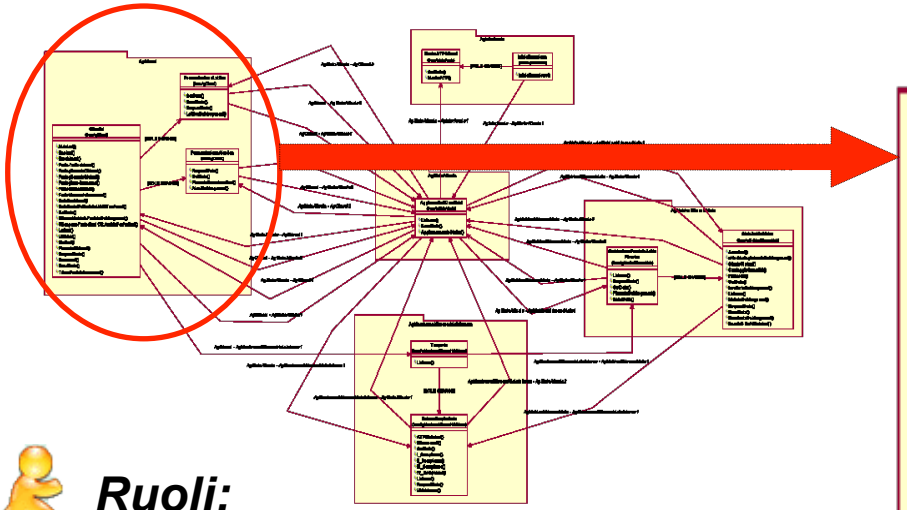
**Inizializzatore**

- **Inizializzazione**: Invio di GDC (inseriti dall'utente) ad AgStatoAttuale.

**Visualizzatore Risultati (MostraATPCitosol)**

- **GetData**: Ricezione GDC aggiornato da AgStatoAttuale.
- **MostraATP**: Filtraggio e Visualizzazione dei risultati finali.

# Fase 6: Roles Description





## *Ruoli:*

- Glicolisi
- Fermentazione Lattica
- Fermentazione Alcolica

## *Comunicazioni:*

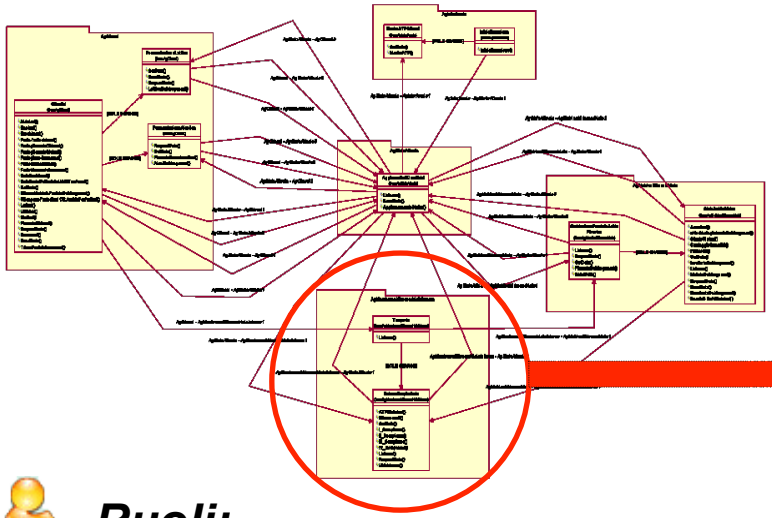
**Glicolisi**

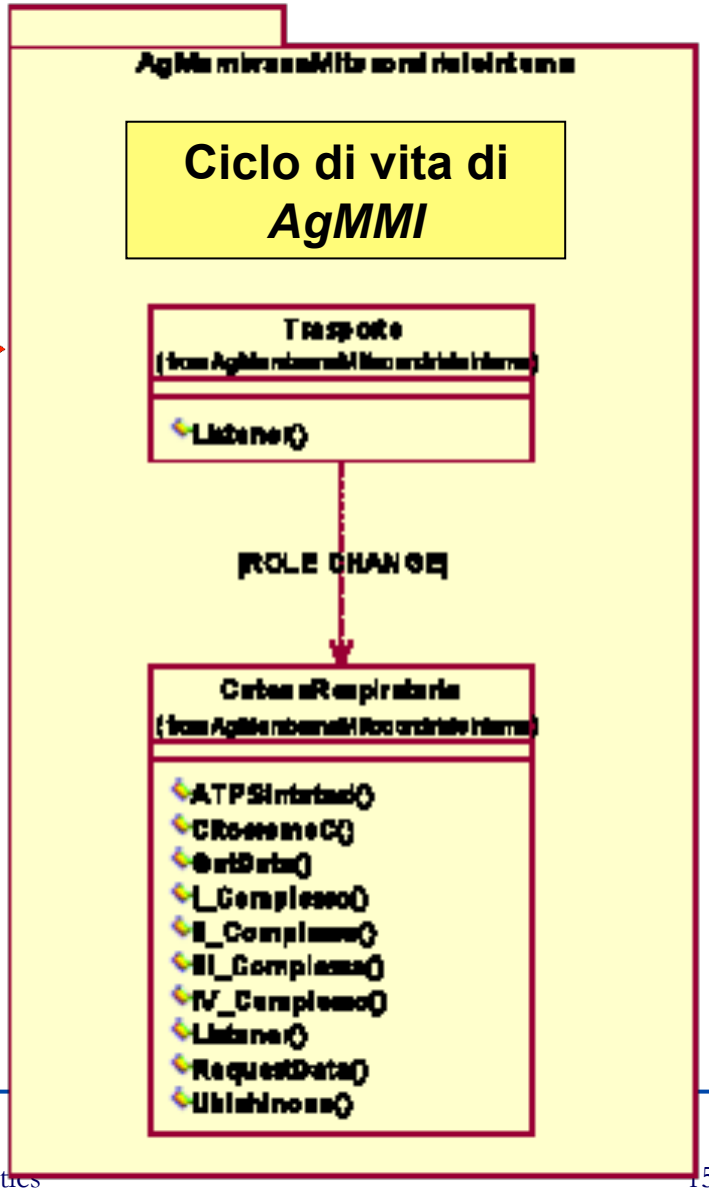
- **Listener**: Ricezione Fase da AgStatoAttuale
- **RequestData**: Richiesta GDG ad AgStatoAttuale
- **GetData**: Ricezione GDG aggiornato da AgStatoAttuale
- **SendData**: Invio GDG aggiornato dopo Glicolisi ad AgStatoAttuale
- **ControlloOssigeno**: Se presente→Invio Fase ad AgMMI

**Ferm. Lattica (Alcolica)**

- **RequestData**: Richiesta GDFL(GDFA) ad AgStatoAttuale
- **GetData**: Ricezione GDFL(GDFA) aggiornato da AgStatoAttuale
- **SendData**:Invio GDFL/A aggiornato dopo Fermentazione ad AgStatoAttuale

# Fase 6: Roles Description






## *Ruoli:*

- **Trasporto**
- **CatenaRespiratoria**

## *Comunicazioni:*

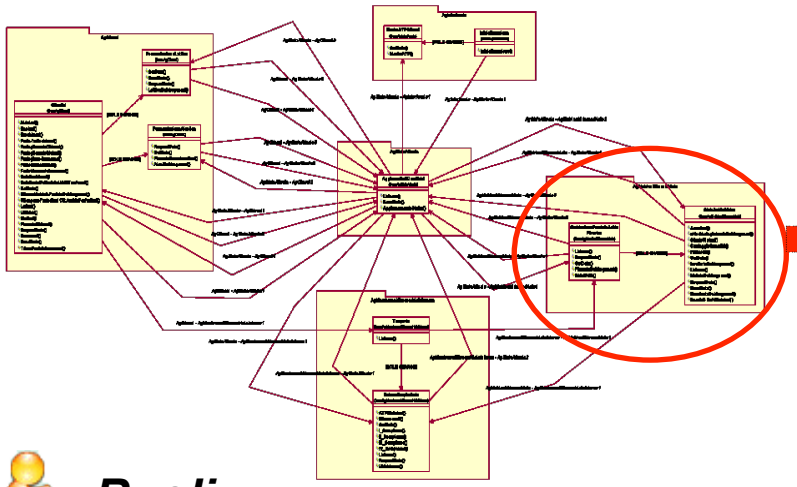
### Trasporto

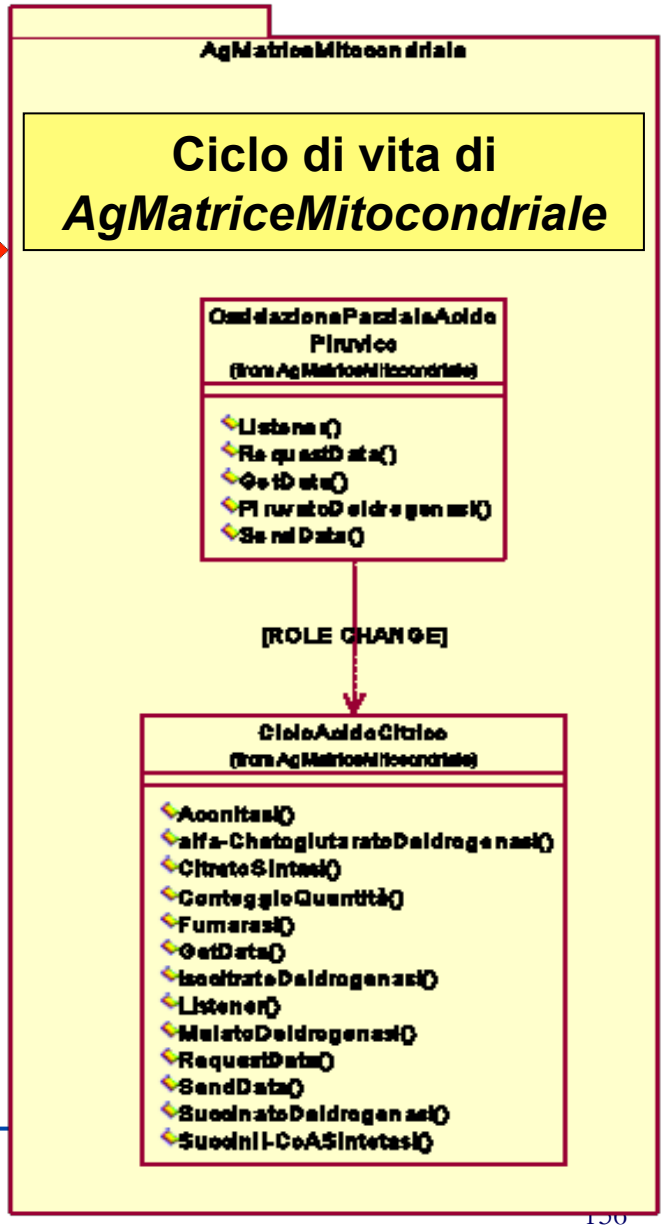
- **Listener**: Trasporto della fase da AgCitosol ad AgMatriceMitocondriale.

### Catena Respiratoria

- **Listener**: Ricezione Fase da AgMatriceMitocondriale.
- **RequestData**: Richiesta GDCR ad AgStatoAttuale.
- **GetData** :Ricezione GDCR aggiornato da AgStatoAttuale.
- **ATPSintetasi** :Invio GDCR aggiornato dopo la Respirazione Cellulare ad AgStatoAttuale.

# Fase 6: Roles Description






## *Ruoli:*

- Ossidazione Parziale Acido Piruvico
- Ciclo Acido Citrico

## *Comunicazioni:*

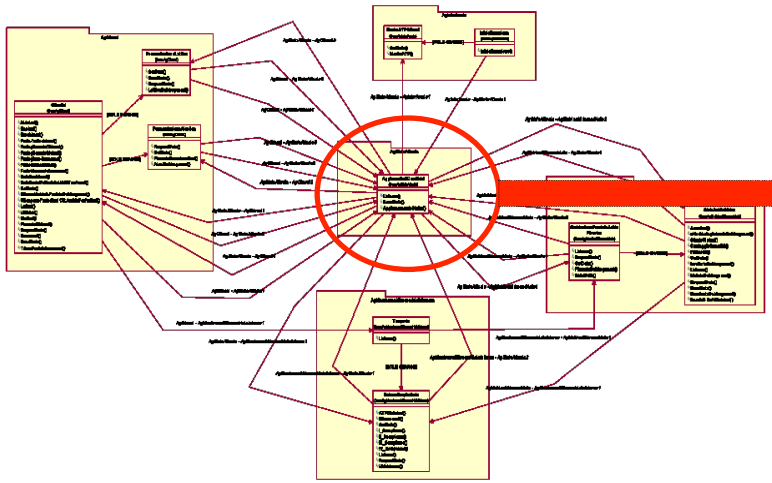
### Ossidazione Parziale Acido Piruvico

- **Listener**: Ricezione Fase da AgMembranaMitocondrialeInterna.
- **RequestData**: Richiesta GDOPP ad AgStatoAttuale.
- **GetData**: Ricezione GDOPP aggiornato da AgStatoAttuale.
- **SendData**: Invio GDOPP aggiornato dopo l'Ossidazione Parziale dell'Acido Piruvico ad AgStatoAttuale.

### Ciclo Acido Citrico (Ciclo di Krebs)

- **RequestData**: Richiesta GDCdK ad AgStatoAttuale dopo l' Ossidazione Parziale dell' Acido Piruvico.
- **GetData**: Ricezione GDCdK aggiornato da AgStatoAttuale.
- **SendData**: Invio GDCdK aggiornato dopo il Ciclo dell'Acido Citrico (Ciclo di Krebs) ad AgStatoAttuale.

# Fase 6: Roles Description




## *Ruoli:*

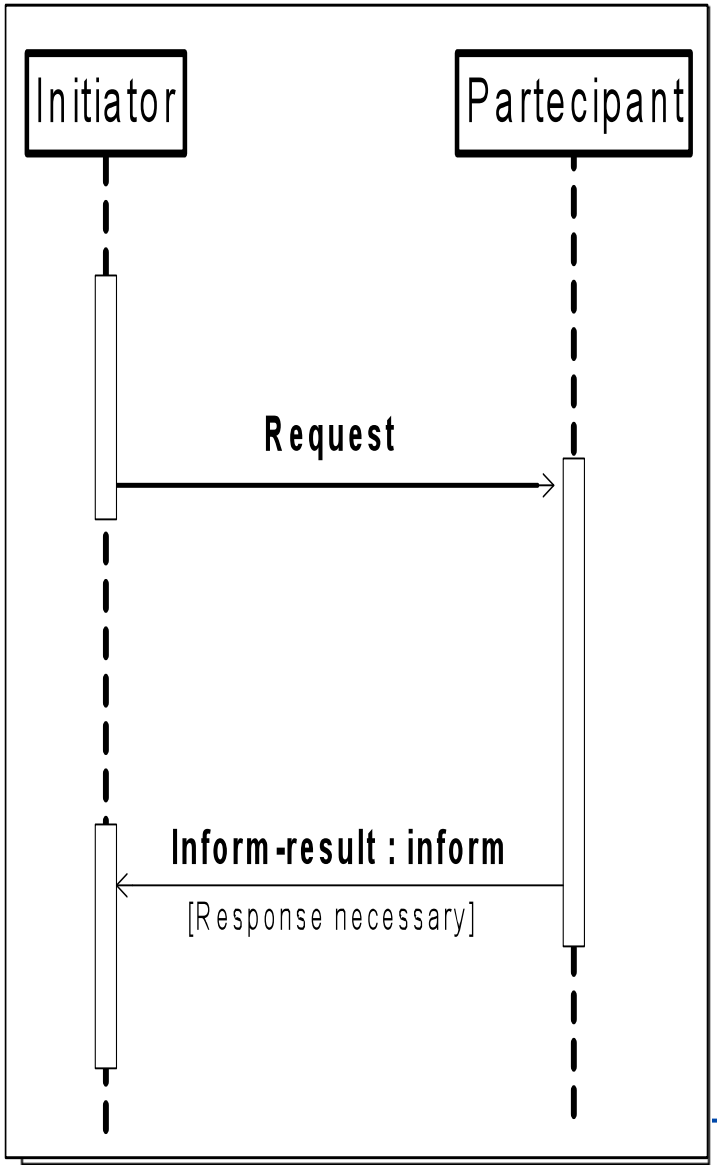
- **Aggiorna Dati Condivisi**

## *Comunicazioni:*

- **Aggiorna Dati Condivisi**

**- Listener**: Ricezione dei messaggi di tutti gli altri agenti.
**- SendData**: Invio dei dati relativi al tipo di comunicazione richiesta.
-**AggiornamentoStatus**: Operazione di aggiornamento di GDC a seguito della ricezione di un gruppo di dati specifico da parte di un altro agente.

# Fase 7: Protocol Description




## FIPA-Request Interaction Protocol

Il "Partecipant" processa la richiesta e prende la decisione di accettarla o meno. Se avviene il rifiuto, allora *refuse* = true e il Partecipant comunica un '`refuse`'. Altrimenti, *agreed* = true.
Se la condizione indica che è richiesta esplicitamente un' accettazione (*notification necessary* = true) allora il partecipant fornisce un `agree`.
Una volta che la richiesta è stata accettata, il Partecipant deve comunicare:

- `Failure`: fallimento nell' esecuzione della richiesta
- `Inform-done`: Se si vuole notificare il compimento della richiesta
- `Inform-result`: Si notifica il compimento ed il risultato della richiesta

Ogni interazione che utilizza questo protocollo è identificata globalmente tramite un unico (e non nullo) conversation-id, assegnato dall' Initiator. Gli agenti coinvolti nell'interazione devono contrassegnare i tutti i loro messaggi ACL con questo identificativo di conversazione.

## Our Hermes Interaction Protocol

L'interazione tra UserAgents e ServiceAgents avviene tramite messaggi

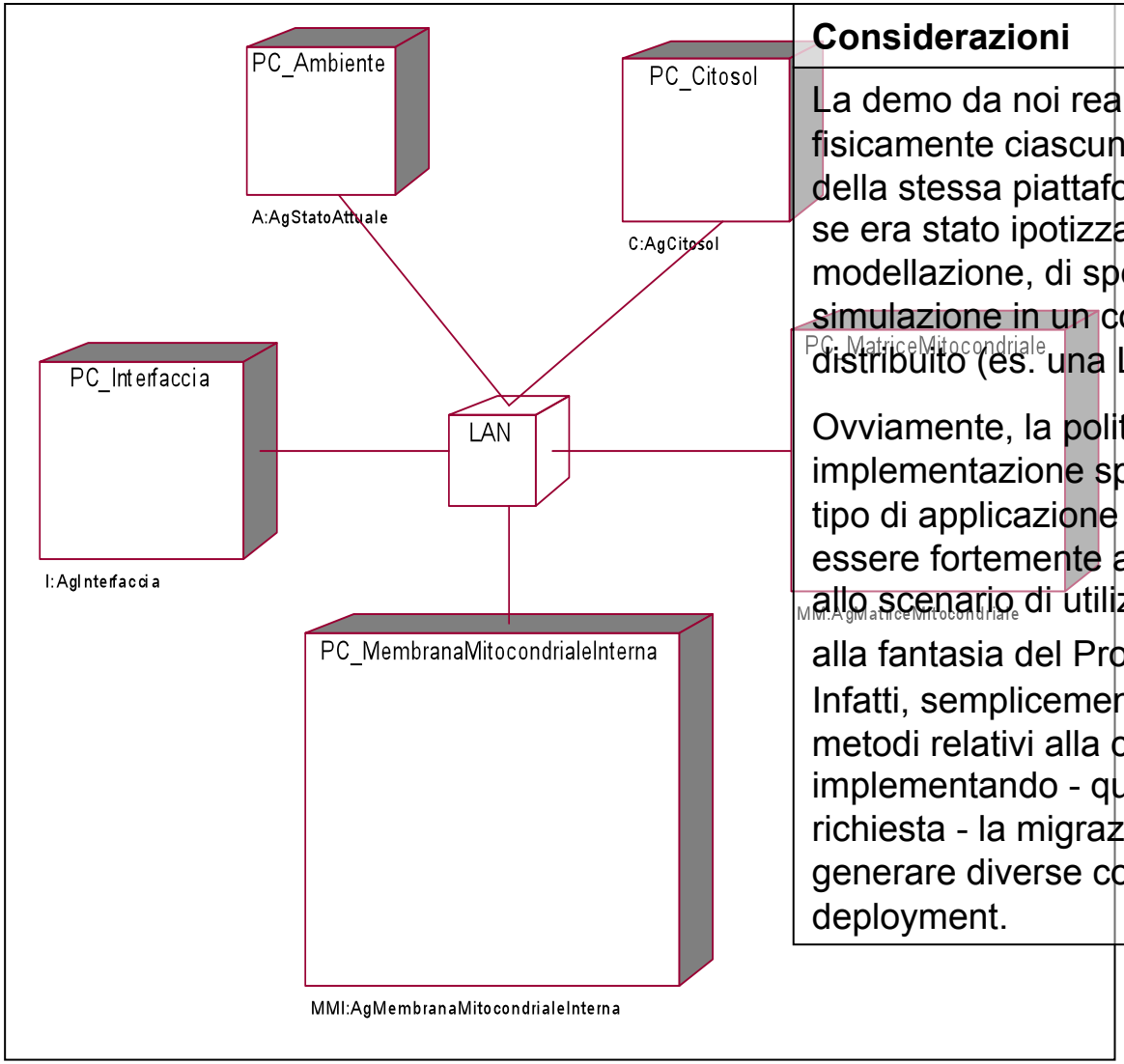
**{ SenderIdentificator , ReceiverIdentificator , MessageContent }**

L'Interaction protocol va dunque implementato ad hoc.
Nella nostra simulazione, sono stati adottati i seguenti principi:

- Il Partecipant non può rifiutare la sottomissione di una Request.
- Una volta accettata la richiesta, il Partecipant comunica, a seconda della fase corrente, un `Inform-result`.

# Fase 8: Multi-Agent Structure Definition

# Fase 9: Deployment Configuration




**Considerazioni**

La demo da noi realizzata includerà fisicamente ciascun agente all'interno della stessa piattaforma locale anche se era stato ipotizzato, in fase di modellazione, di sperimentare la simulazione in un contesto realmente distribuito (es. una LAN).

Ovviamente, la politica di implementazione specifica per questo tipo di applicazione distribuita può essere fortemente adattabile allo scenario di utilizzo o comunque alla fantasia del Programmatore ☺ . Infatti, semplicemente modificando i metodi relativi alla comunicazione ed implementando - qualora fosse richiesta - la migrazione, si possono generare diverse configurazioni di deployment.

# Implementazione su Hermes

## Requisiti e Limitazioni poste

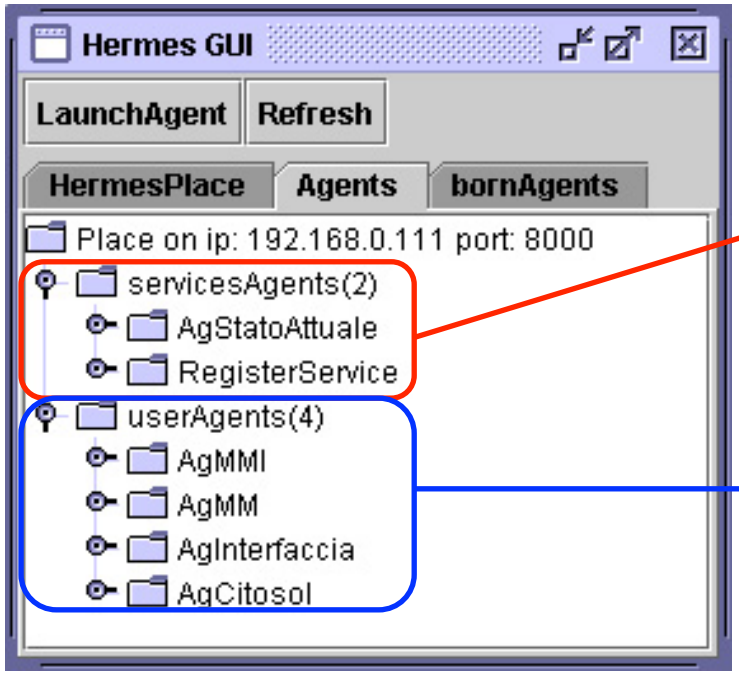
- **Gli agenti comunicano tramite messaggi dalla struttura nota (*C.O.D. Diagram*) che rappresentano concetti noti (*D.O.D. Diagram*).**
- **Le comunicazioni sono sincrone.**
- **L' interazione tra agenti richiede un protocollo `request-response` like.**
- **Gli agenti devono, all' occorrenza, "conoscersi".** (e.g. Per permettere lo scambio e la validazione delle informazioni, due o più agenti devono possedere gli Identificator dei partecipanti della comunicazione).

## Soluzioni possibili

- **Ciascun concetto è una Classe Java poiché viene trattato come un complesso tipo di dato interscambiabile.**
- **Utilizzo di `getMessageSynch()` per ciascun `Message`.**
- **Implementazione (da codice) di un protocollo ad hoc.**
- **Registrazione di tutti gli Agenti ad un `ServiceAgent` '*RegisterService*'.**

# Implementazione su Hermes

## Simulation Context: *CellPlace* su localhost




**serviceAgents (2)**:
- AgStatoAttuale
- RegisterService

**userAgents (4)**:
- AgInterfaccia
- AgCitosol
- AgMM
- AgMMI

**Motivazioni Principali**

- **AgStatoAttuale** rappresenta un Ambiente artificiale, contenitore di dati aggiornabili e condivisibili da ciascun agente. Esso può anche intraprendere operazioni attive (e.g. passaggio Fase etc.).
- **Register Service** è stato introdotto appositamente per implementare l'interscambio di Identificator. Questo è fondamentalmente fondalmente un servizio… quindi è un `serviceAgent` ☺.
- I rimanenti agenti sono `userAgents` dal momento che rappresentano componenti mobili del sistema.

# Implementazione su Hermes

## Registrazione di AgStatoAttuale e RegisterService

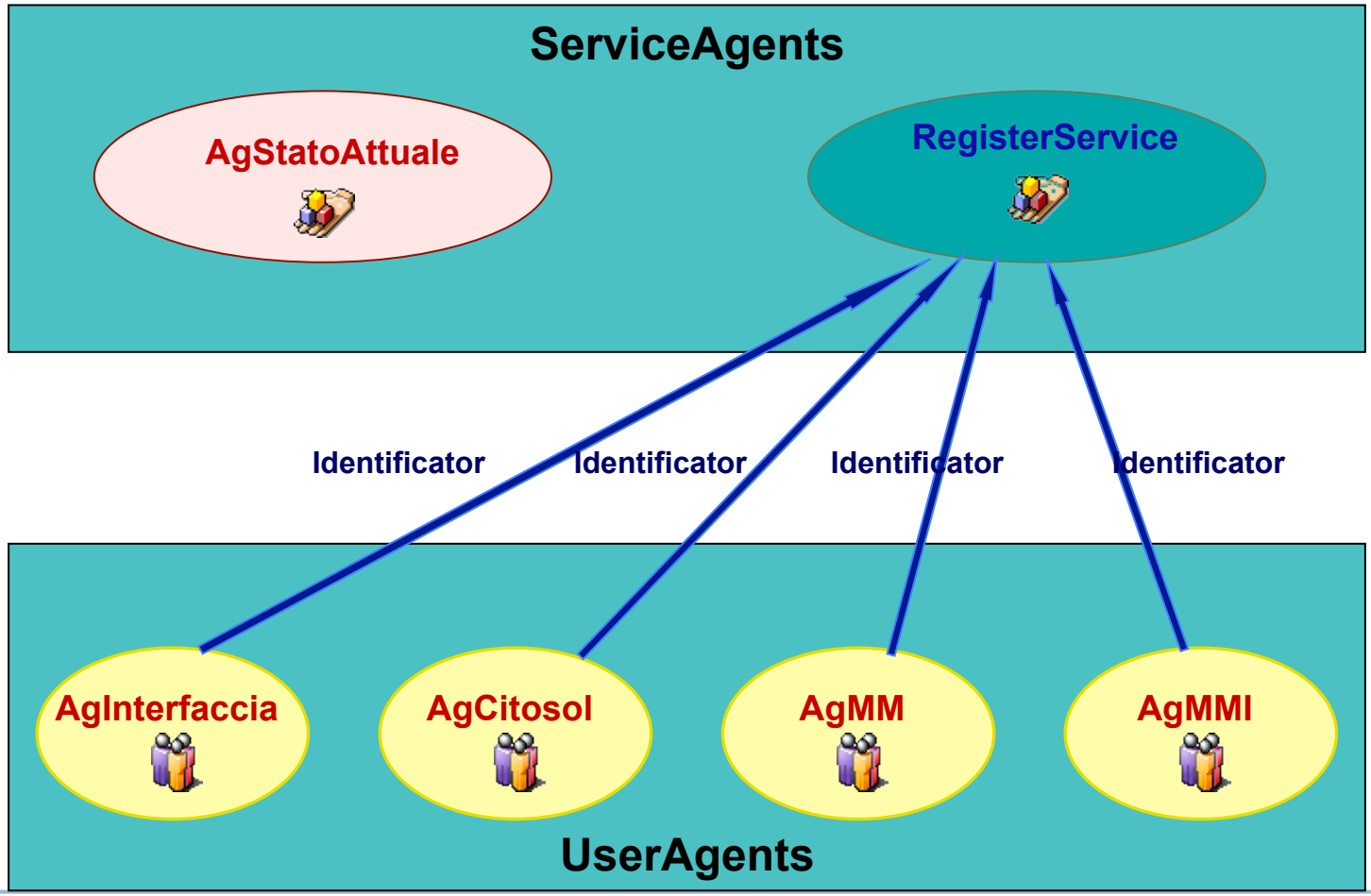


```
public void init()
{
  reception();
  System.out.println(this.getAgentName() + " Attivo.");
  String[] features = new String[2];
  features[0] = "AgStatoAttuale";
  features[1] = "localEnvironment";
  registerService(features);
  .....
}
```

```
public void init()
{
  reception();
  System.out.println(this.getAgentName() + " Attivo.");
  String[] features = new String[2];
  features[0] = "RegisterService";
  features[1] = "localRegisterService";
  registerService(features);
  .....
}
```

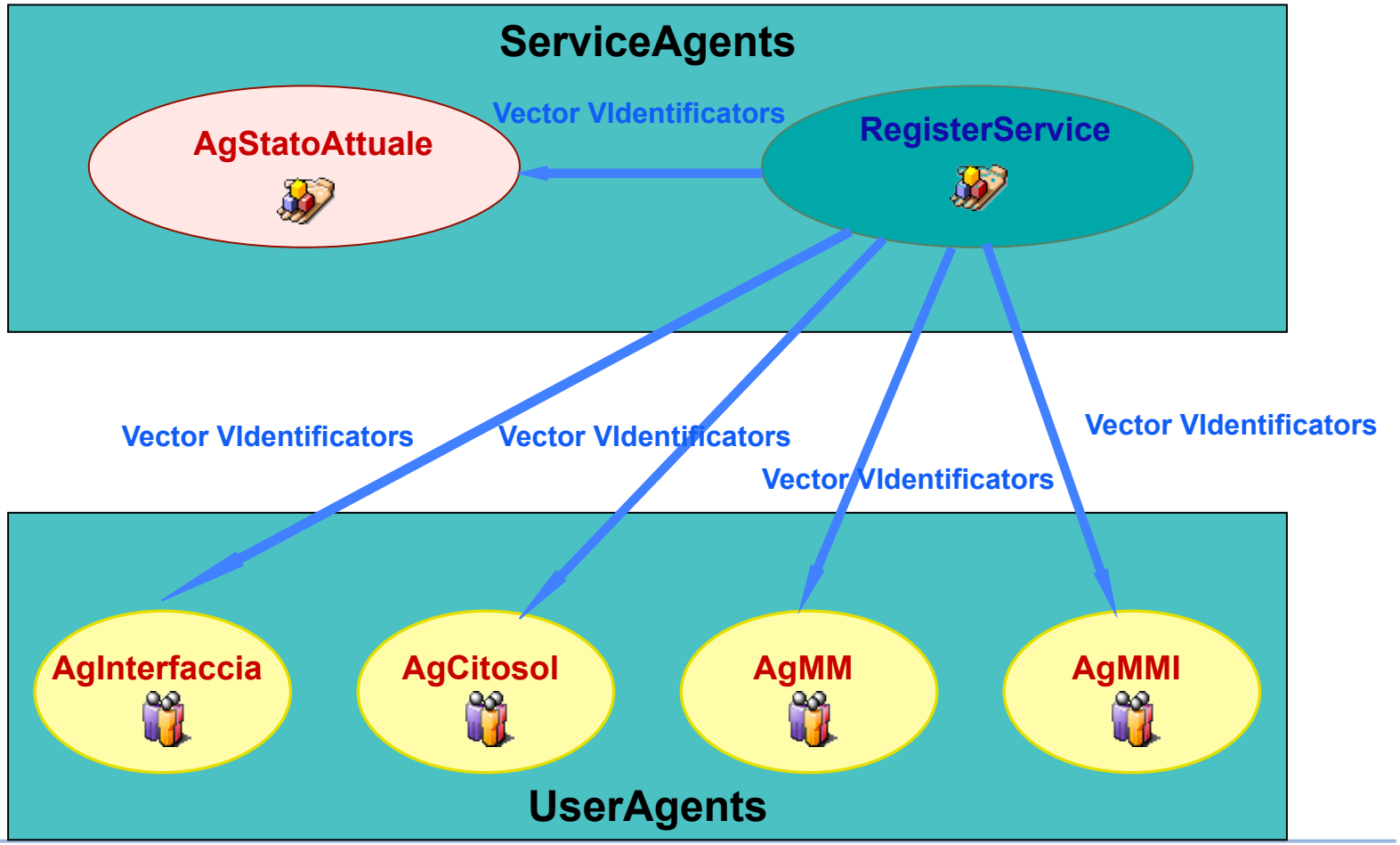
# Implementazione su Hermes

- Registrazione di AgInterfaccia, AgCitosol, AgMM ed AgMMI a RegisterService



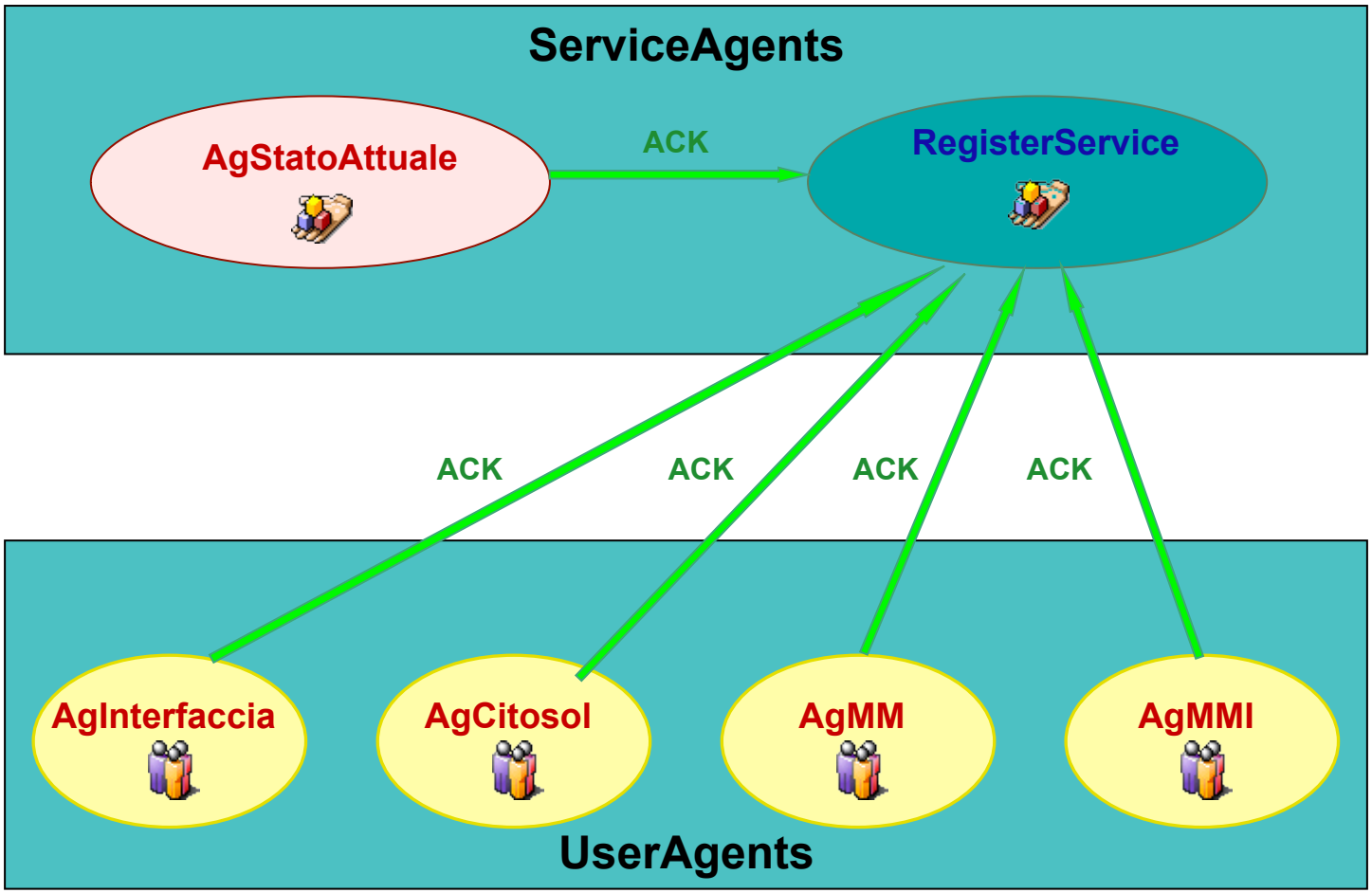
# Implementazione su Hermes

- Registrazione di AgInterfaccia, AgCitosol, AgMM ed AgMMI a RegisterService



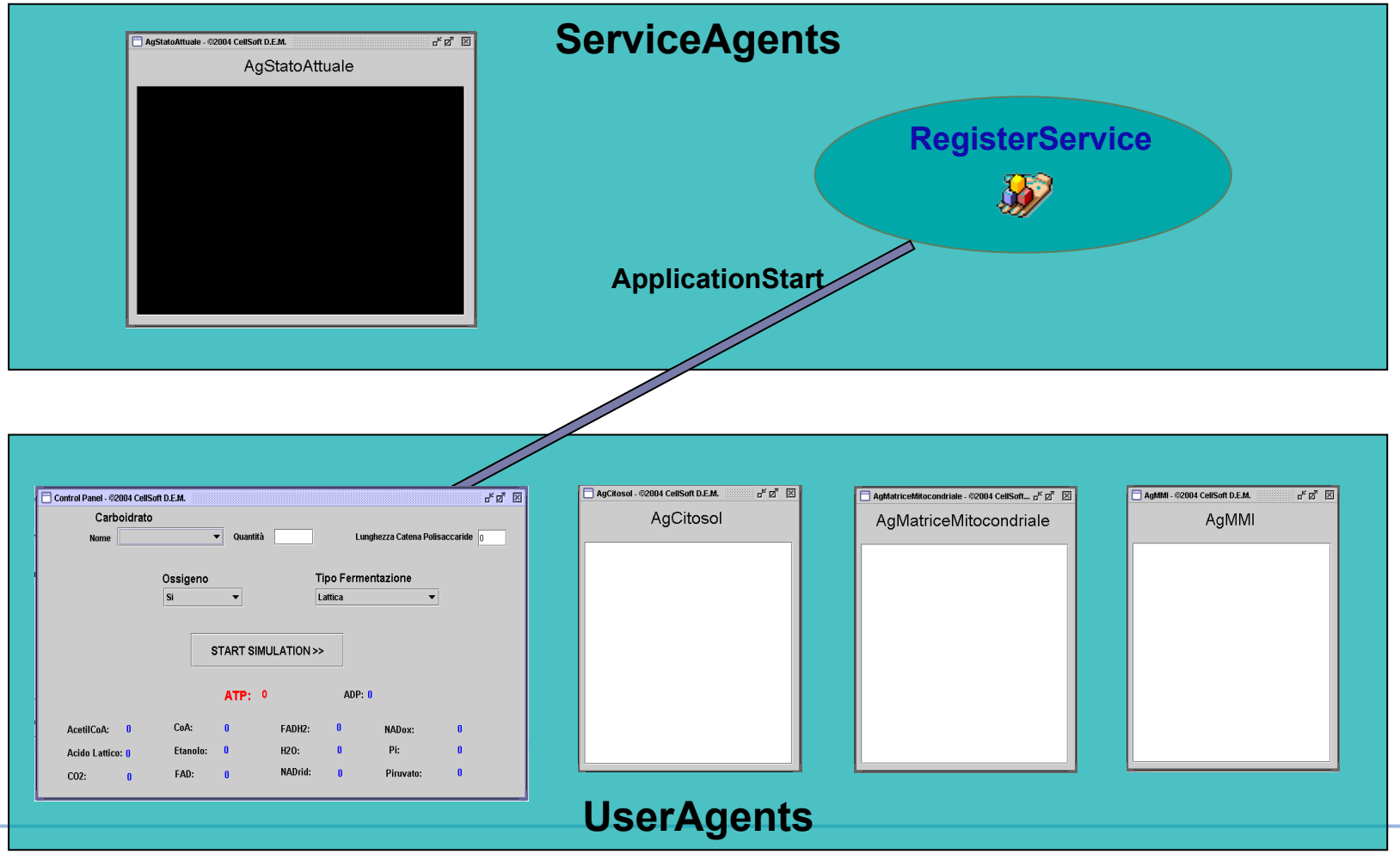
# Implementazione su Hermes

- Registrazione di AgInterfaccia, AgCitosol, AgMM ed AgMMI a RegisterService



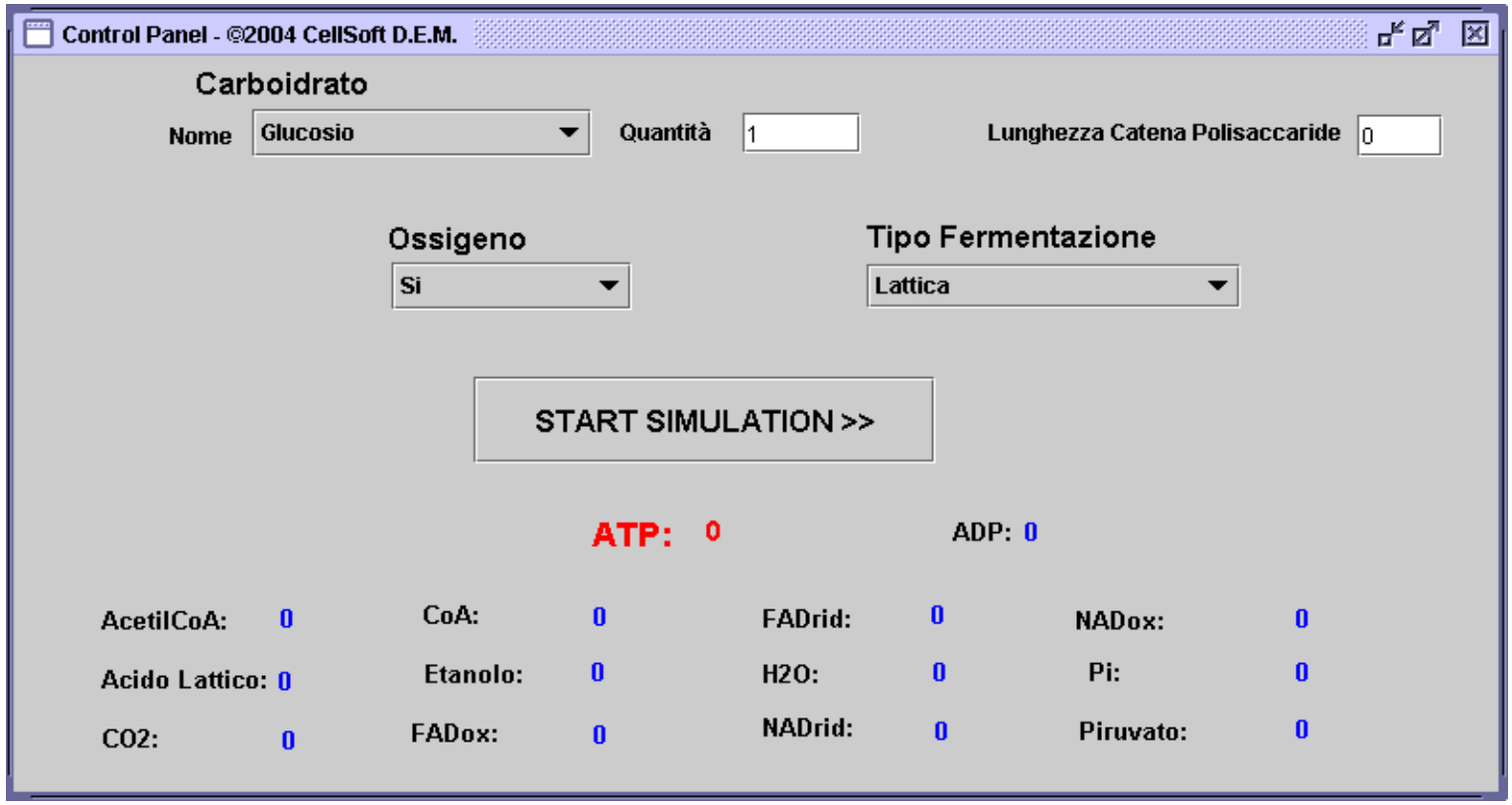
# Implementazione su Hermes

- Registrazione di AgInterfaccia, AgCitosol, AgMM ed AgMMI a RegisterService



# Implementazione su Hermes

- **Test: produzione di ATP a partire da una molecola di Glucosio (in presenza di ossigeno)**

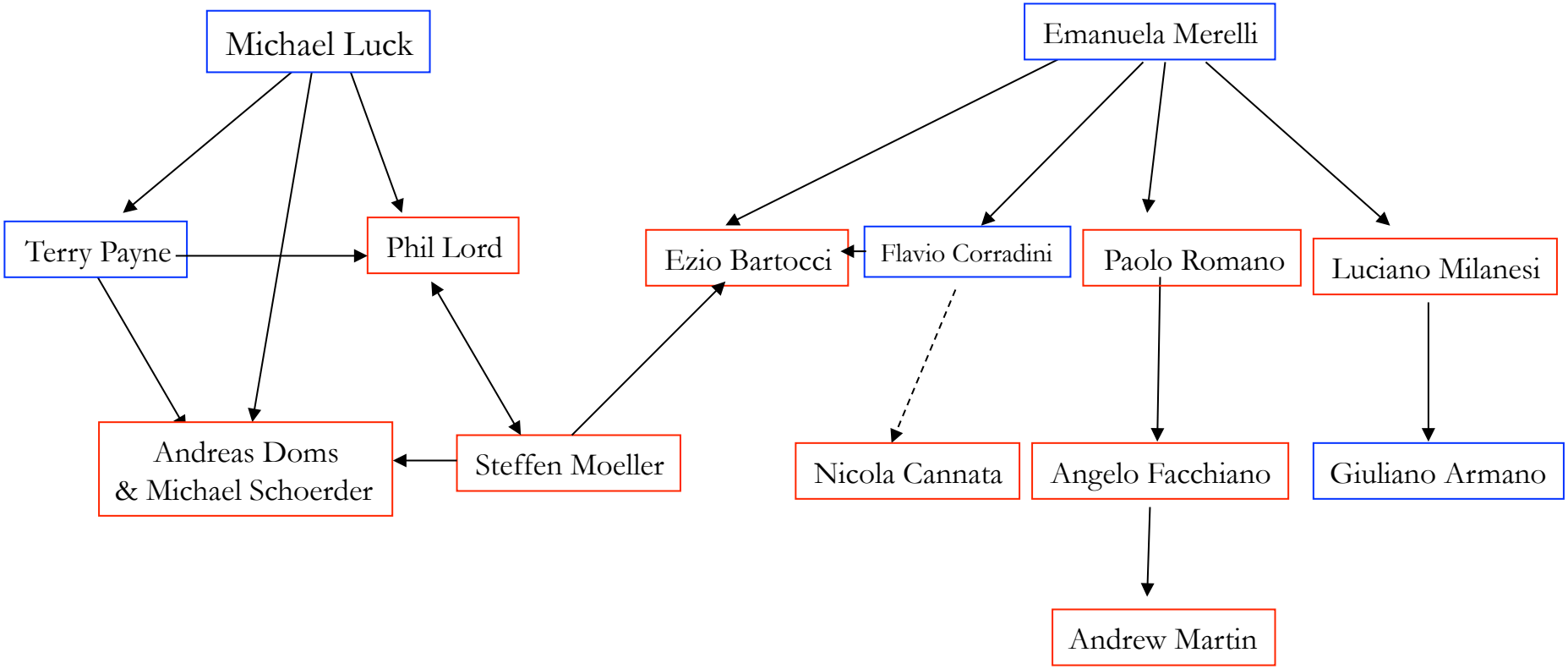



Ci aspettiamo che l'ATP prodotto, a seguito della Glicolisi, della OPP, del Ciclo di Krebs e la Respirazione Cellulare, sia **38**.

# AgentLink III and Bioinformatics

- TFG – why?
- Roadmap – what aims?
  - What technology is required?
  - Why is agent technology a good bet
  - What more is needed?
  - Is it just an application area for standard techniques?

# BioagentTFG Promoters -- July 2004



# Informatics and BioScience group at Unicam

## Computer Scientists

Formal Methods

- Flavio Corradini (prof.)
- Diletta Cacciagrano (reseacher)
- M.Rita Di Berardini (Phd std)

Agents

- Emanuela Merelli (reseacher)
- Rosario Culmone (researcher)
- Marco Vita (student in thesis)

- Diego Bonura (undergraduate)
- Leonardo Vito (undergraduate)
- Matteo Kalogris (undergraduate)

## Bioinformatics

- Nicola Cannata (Univ. Padova)
- Ezio Bartocci (graduate)

## Bioscientists

- Roberto Ciccocioppo (pharmacologist)
- Antonietta La Terza (biologist)
- Mauro Angeletti (biochemist)