Agent Based Modelling and Simulation in Systems Biology: Mobile Aspects

Emanuela Merelli
Università di Camerino

Reykjavik University - 18 September 2007

Plan

- Computational Systems Biology
- Strength and weakness of the formal approach
- Multi-level modeling in systems biology
- Agent-based modeling and simulation
- Case Study: from carbohydrate oxidation process

- UNICAM CoSy Research Group
- LITBIO project
- Erasmus Mundus Master Course in *Formal Methods for Systems Engineering* (FOMSE)
- Conclusion

The challenge of the 21st century

The challenge of the 21st century will be to understand **how** biological molecules in living systems **integrate** to complex systems, how these systems **function** and their **evolution**.

We are scaling up from computational molecular biology (i.e. genomes, transcriptomes, proteomes, metabolomes) to **computational systems biology**

Systems Biology

The science of **building models** of biological systems, from cells to organisms, by:
- systematically perturbing the system (biologically, genetically, or chemically); and
- monitoring gene, protein and other pathways.

Models are used in **drug research** for understanding their effects and side-effects and for building personalized medicine

Systems Biology is typically require tools to

- integrate the huge amount of biological data in order to
- understand the behaviour of biological systems, and to
- study the relationships and interactions between the various parts of a biological system, such as organelles, cells, physiological systems, organisms etc.

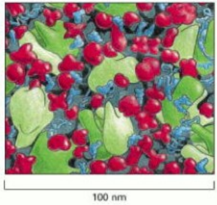
From local interactions to global behaviour

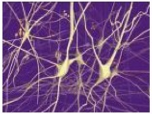
Most of the biological systems are **complex in nature**

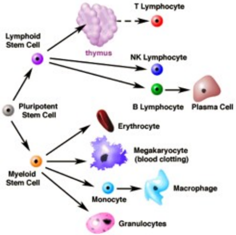
- have many of its components coupled in a non-linear fashion
- exhibit the emergence property from simpler interactions rules, i.e. the formation of complex patterns
- ...

Thus the proposal of a **multi-level approach** in modelling biological systems would allow zooming-in and zooming-out into through models

From molecules to cell compartments, organisms, ... ecosystems



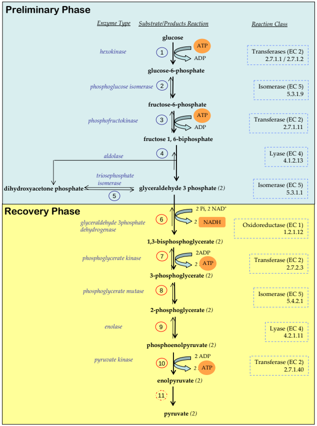




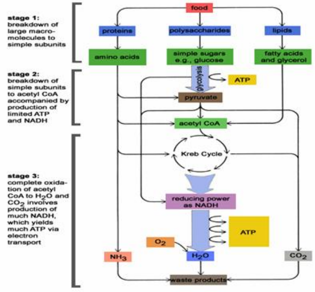
the global behaviour of the system can be determined by defining the lower-level interaction rules among components

Glycolysis metabolic pathway

$$glucose + 2NAD^{+} + 2ADP + 2Pi \rightarrow 2pyruvate + 2NAD + 2H^{+} + 2ATP + 2H_2O$$



The Carbohydrate Oxidation is the energy production process performed by two active components of the Cell: Cytoplasm and Mitochondrion


stage 1:
breakdown of large macro-molecules to simple subunits
stage 2:
breakdown of simple subunits to acetyl CoA accompanied by production of limited ATP and NADH
stage 3:
complete oxidation of acetyl CoA to H_2O and CO_2 involves production of much NADH, which yields much ATP via electron transport
food
proteins
polysaccharides
lipids
amino acids
simple sugars e.g., glucose
fatty acids and glycerol
glycolysis
ATP
pyruvate
acetyl CoA
Kreb Cycle
reducing power as NADH
ATP
O_2
NH_3
H_2O
CO_2
waste products

Metabolic pathways

- Cytoplasm (without oxygene)
  - **Glycolysis** *(convert 1 glucose into 2 pyruvate + ...)*
  - Lactic Fermentation *(reduce pyruvate to lactate)*
  - Alcoholic Fermentation *(reduce pyruvate to ethanol)*
- Mithocondrion (with oxygene)
  - Mithocondrial Inner Membrane
    - Transportation *(pyrutave frpm cytop. to MM)*
    - **Respiration Chain** *()*
  - Mithocondrial Matrix
    - **Partial oxidation of pyruvate**
    - Kreb's cycle

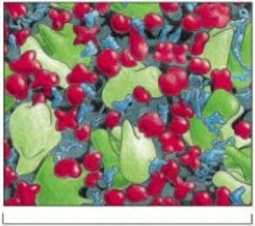
Cytoplasm: a lower-level bio-complexity

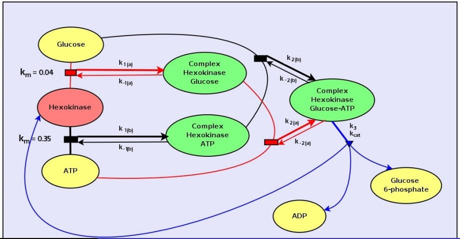
To model a portion (10^{-15} liters) of a cytoplasm, we must simulate **the movement** and the **interactions** of about **27 millions of autonomous entities**.

Each interaction has associated 2 kinetic constants;

in many cases those values are still undiscovered; each day one of them could be

discovered by one of the hundreds of research groups distributed all over the world





Multi-level modelling

A model (M) for a system (S) and experiment (E) is anything to which E can be applied in order to answer questions about S
M. Minsky, *Models, minds, machines*. 1965

This definition implies the co-existence of several models for any system.

A model can be distinguished w.r.t their organizational level:

- Is the system dscribed at *macro* or at *meso* ot at *micro* level?
- Are concentration changes the focus of interest, or the behaviour of individuals?
- ...

It has been observed that the language used by biologists for verbal communications is **vague** and avoid clear prediction.

Also, the use of colourful diagrams, so much beloved from biologists, is often usually useless for computer reasoning and quantitative analysis.

Formal techniques are instead likely to give the chance to inherit the wide range of existing methods and tools provided by computer science (such as property design and verification, and automated reasoning).

ODE limitations and Multi-Scaling power

The Ordinary Differential Equation (ODE) is the classical approach arising from the biochemical point of view.

A network of interactions (chemical reactions) between molecules (metabolites, proteins, genes) is established and ODE are used to numerically describe the continuous variations in the concentration of substances.

ODE beset with the **combinatorial explosion**

Process abstraction

Using the process abstraction opens up new possibilities for understanding molecular systems.

Computers and biomolecular systems both start from a small set of elementary components from which, layer by layer, more complex entities are constructed with evermore sophisticated functions.

The framework should also allow the system to be **zoomed-in** and **zoomed-out** at different levels of abstraction.

Stochastic π-calculus

In recent years, there has been increasing interest in computational models of biological systems based on various calculi of communicating processes such as the **stochastic π-calculus**.

Regev, A., Silverman, W., Shapiro, E.: Representation and simulation of biochemical processes using the pi-calculus process algebra. In: Pacific Symposium of Biocomputing 6 (PSB 2001)

Regev, A.: Representation and simulation of molecular pathways in the stochastic pi-calculus. In: 2nd workshop on Computation of Biochemical Pathways and Genetic Networks, Heidelberg, Germany (2001)

In 2003, Aviv Regev and Edu Shapiro proposed “the π-**calculus** as an abstraction for biomolecular modelling”

Albeit π-calculus allows to reflect the inherent concurrency, communication and stochasticity of cellular systems, **each new case reveals drawbacks** in modelling biological systems

The minimality of the calculus makes it heavy to manage models with **complex coordination** and dependencies. It forces the modeler to encode high-level interaction patterns into low-level means offered by the calculus.

The encodings makes models **harder to write and understand**, and obscure the original high-level biological model with a plethora of irrelevant computational details.

A criticism to π-calculus

Perhaps it depends on ... the asymmetry of the send/receive communication model

Whenever two molecules A and B bind to form a complex AB, they each create a private channel. The protocol for exchanging these new private channels is itself complicated by the directionality of communication

[D. Duchier, C. Klutter *Biomolecular agents as multi-behavioural concurrent objects* ENTCS, 150, 2006]

A criticism to π-calculus

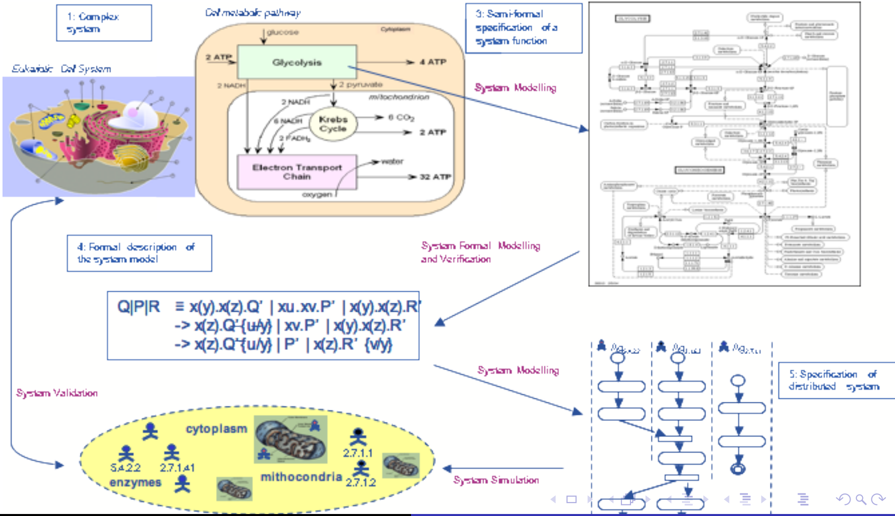
Perhaps it depends on ... the asymmetry of the send/receive communication model

not very suitable to model the **simultaneous exchange!**

Whenever two molecules *A* and *B* bind to form a complex *AB*, they each create a private channel. The protocol for exchanging these new private channels is itself complicated by the directionality of communication

[D. Duchier, C. Klutter *Biomolecular agents as multi-behavioural concurrent objects* ENTCS, 150, 2006]

Coordination, in CS, refers to languages and models for coordinating, configuring, and, generally, glueing together components. e.g., LINDA



1: Complex system

3: Semi-formal specification of a system function

4: Formal description of the system model

5: Specification of distributed system

$$Q|P|R \equiv x(y).x(z).Q' \mid xu.xv.P' \mid x(y).x(z).R' \\ \to x(z).Q'\{u/y\} \mid xv.P' \mid x(y).x(z).R' \\ \to x(z).Q'\{u/y\} \mid P' \mid x(z).R' \; \{v/y\}$$

Shall **agent-based paradigm** help to overcome the limitation of the understandability of the π-calculus by offering a **multi-level modelling environment** and ...

... exploit the Grid computational resources the simulation of biological systems?

Milner's agent point of view

In 1989 Milner wrote ...

- "each of the several part of the system has its own identity, which persist through time; we shall call these parts **agents**"
- ... Communication and concurrency notions, both essential in understanding complex dynamic systems. On the one hand such a system has diversity, being composed of **several parts** each acting concurrently with, and independently of, other parts; on the other hand a complex system has **unity** (or else we shall not call it a system) achieved through communication among its parts.
- ... underling both these notions is the assumption that each of the several parts of such a system has its own identity, which persists through time; we shall call these parts **agents**
- ... This lead to use the term **agent** very broadly, to mean any system whose behaviour consists of discrete actions; an which for one purpose we take to be atomic may, for other purpose, be decomposed into sub-agents acting concurrently and interacting.

An **agent** is an "encapsulated" computer system **situated** in some environment and capable of **flexible**, **autonomous** action in that environment in order to meet its design objectives

[M. Wooldridge, *Agent-based software engineering.* In IEE Proceedings of Software Engineering, 1997]

An autonomous agent is a computer system capable of flexible, **autonomous** (problem-solving) action, situated in dynamic, open, unpredictable and typically multi-agent domains.

The agent has the control over its internal state and over its own behaviour

A situated agent is a computer system capable of flexible, autonomous (problem-solving) action, **situated** in dynamic, open, unpredictable and typically multi-agent domains.

The agent perceives the environment through sensors and acts through effectors

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

reactive: **the agent responds in timely fashion to environmental change**

proactive: **the agent acts in anticipation of future goals**

Reactive agents can be defined by a 6-tuple

$$< E, P, A, see, do, action >$$

where

- E is the set of states for the environment
- P is a partition of E (representing a relevant abstraction of the environment from the agent's point of view)
- A is a set of actions
- $see : E \longrightarrow P$
- $action : P \longrightarrow A$
- $do : A \times E \longrightarrow E$

These agents observe the environment (*see*), find the appropriate action (*action*), and act (*do*)

Proactive agents can be defined by a 9-tuple

$$< D, E, P, A, d_0, see, KB, do, action >$$

where

- D is a set of predicate calculus databases
- E, P, A, see and do are the same as for reactive agents
- d_0 is the initial KB
- $action : D \times P \rightarrow A$
- $KB : D \times P \rightarrow D$

These agents observe the environment (*see*), choose their next action according to some kind of reasoning on their current knowledge (*action*), act (*do*), and update their knowledge base

Mobile agents are agents that move across distributed environments (execution platforms)

Mobile agents for Global computing and **mobile code** for Grid computing are new paradigms for distributed computing that complement such technologies as distributed objects, remote compute servers, and distributed programming systems such as PVM and MPI

Multi-agent system

The description of a system consists of a collection of **interactive agents** and a set of coordination rules.

An agent has a name and a set of behaviours. He can adopts **different behaviours** at differents times depending on the environmental conditions.

At **meso level** of modelling biological systems, the elementary interactions consist of the binding or unbinding of two agents, the modification of the state of a interface (site), and the deletion or creation of an agent.

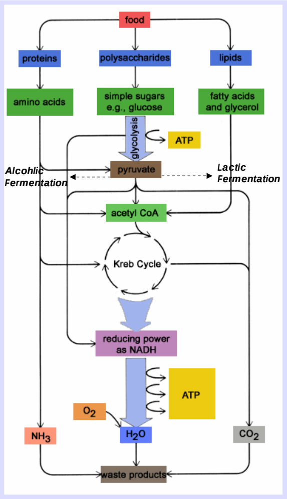
Legend

⇨ Metabolic pathway

→ Compounds reaction

🟦 🟨 🟩 Molecules

Step 1:
Transformatrion of glucose into pyruvate



Compartment 1:
Cytoplasm

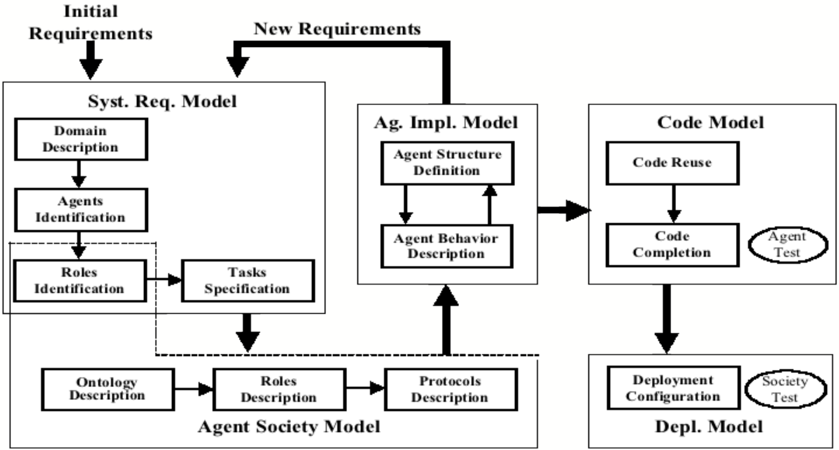
Step 2:
Transformation of acetyl CoA into NADH

Compartment 2.1:
Mitochondrial Matrix

Compartment 2:
Mitochondrion

Step 3:
Transportation of NADH into ATP

Compartment 2.2:
Inner Mitochondrial Membrane

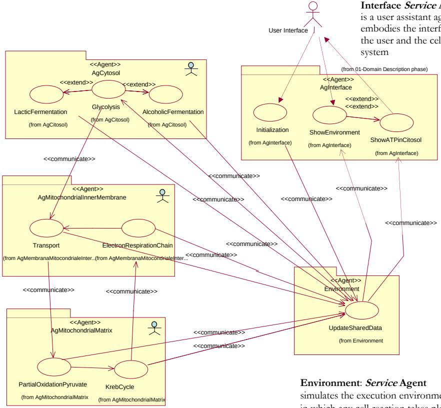

Initial Requirements
New Requirements
Syst. Req. Model
Domain Description
Agents Identification
Roles Identification
Tasks Specification
Ag. Impl. Model
Agent Structure Definition
Agent Behavior Description
Code Model
Code Reuse
Code Completion
Agent Test
Ontology Description
Roles Description
Protocols Description
Agent Society Model
Deployment Configuration
Society Test
Depl. Model

Carbohydrate oxidation: the PASSI agent identification UML diagram

Agent Citosol simulates the role of Cytoplasm in the Glycolysis, Lactic and Alcoholic Fermentations functions

Agent Inner Mithocondrial Membrane simulates the role of IMM in the Transport and Electron Respiration Chain functions

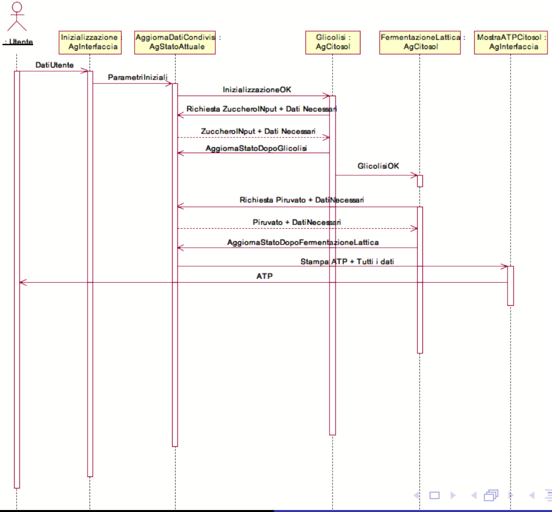
Agent Mithocondrial Matrix: simulates the role of MM in the Partial Oxidation of Pyruvate and Kreb's Cycle functions…



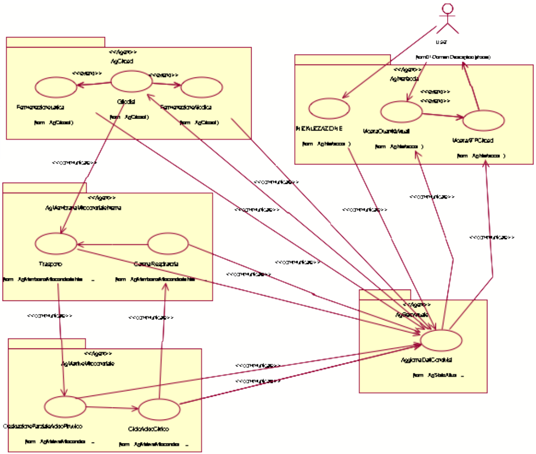
Interface *Service* Agent is a user assistant agent. It embodies the interface between the user and the cell simulation system

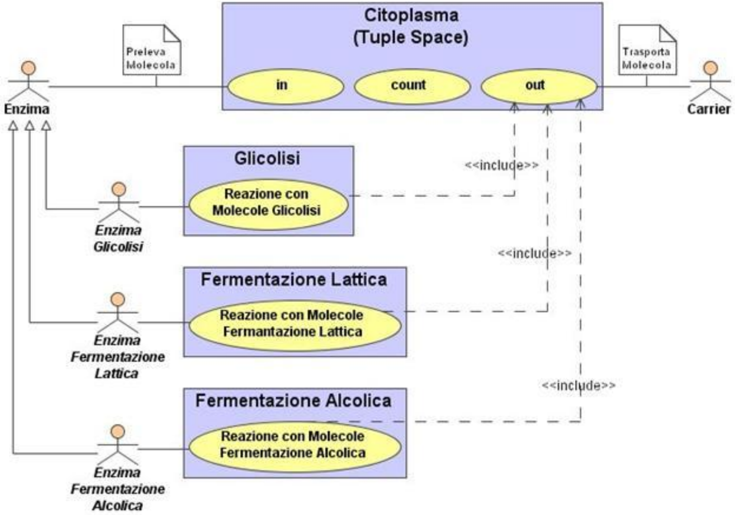
Environment: ***Service* Agent** simulates the execution environment in which any cell reaction takes place

Roles in Lactic Fermentation function

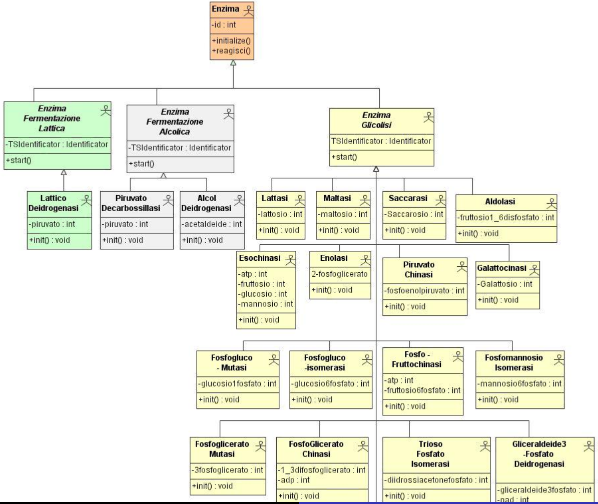

: Utente
Inizializzazione : AgInterfaccia
AggiornaDatiCondivisi : AgStatoAttuale
Glicolisi : AgCitosol
FermentazioneLattica : AgCitosol
MostraATPCitosol : AgInterfaccia
DatiUtente
ParametriIniziali
InizializzazioneOK
Richiesta ZuccheroINput + Dati Necessari
ZuccheroINput + Dati Necessari
AggiornaStatoDopoGlicolisi
GlicolisiOK
Richiesta Piruvato + DatiNecessari
Piruvato + DatiNecessari
AggiornaStatoDopoFermentazioneLattica
Stampa ATP + Tutti i dati
ATP

Carbohydrate oxidation: the PASSI agent identification UML diagram




Citoplasma
(Tuple Space)
in
count
out
Preleva Molecola
Trasporta Molecola
Enzima
Carrier
Glicolisi
Reazione con Molecole Glicolisi
Enzima Glicolisi
<<include>>
<<include>>
<<include>>
Fermentazione Lattica
Reazione con Molecole Fermantazione Lattica
Enzima Fermentazione Lattica
Fermentazione Alcolica
Reazione con Molecole Fermentazione Alcolica
Enzima Fermentazione Alcolica

UML Class Enzymes



Enzima
-id : int
+initialize()
+reagisci()

Enzima Fermentazione Lattica
-TSIdentificator : Identificator
+start()

Enzima Fermentazione Alcolica
-TSIdentificator : Identificator
+start()

Enzima Glicolisi
TSIdentificator : Identificator
+start()

Lattico Deidrogenasi
-piruvato : int
+init() : void

Piruvato Decarbossillasi
-piruvato : int
+init() : void

Alcol Deidrogenasi
-acetaldeide : int
+init() : void

Lattasi
-lattosio : int
+init() : void

Maltasi
-maltosio : int
+init() : void

Saccarasi
-Saccarosio : int
+init() : void

Aldolasi
-fruttosio1_6disfosfato : int
+init() : void

Esochinasi
-atp : int
-fruttosio : int
-glucosio : int
-mannosio : int
+init() : void

Enolasi
2-fosfoglicerato
+init() : void

Piruvato Chinasi
-fosfoenolpiruvato : int
+init() : void

Galattocinasi
-Galattosio : int
+init() : void

Fosfogluco - Mutasi
-glucosio1fosfato : int
+init() : void

Fosfogluco -isomerasi
-glucosio6fosfato : int
+init() : void

Fosfo - Fruttochinasi
-atp : int
-fruttosio6fosfato : int
+init() : void

Fosfomannosio Isomerasi
-mannosio6fosfato : int
+init() : void

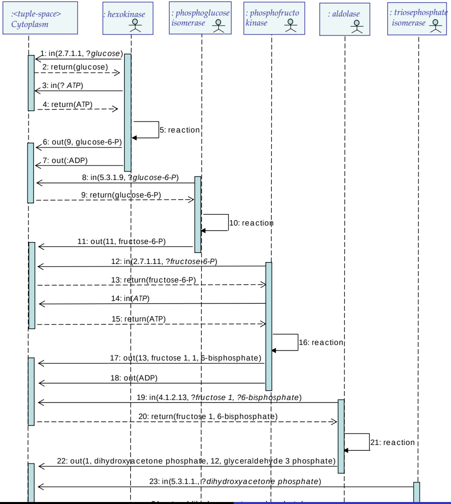
Fosfoglicerato Mutasi
-3fosfoglicerato : int
+init() : void

FosfoGlicerato Chinasi
-1_3difosfoglicerato : int
-adp : int
+init() : void

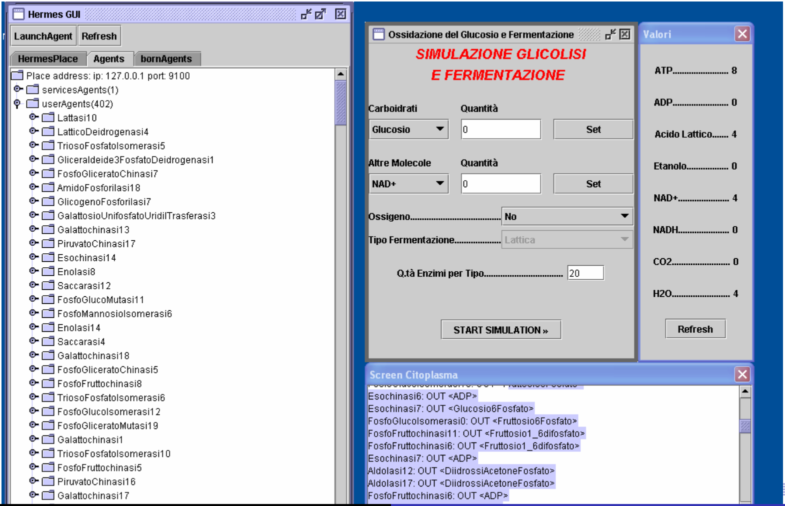
Trioso Fosfato Isomerasi
-diidrossiacetonefosfato : int
+init() : void

Gliceraldeide3 -Fosfato Deidrogenasi
-gliceraldeide3fosfato : int
-nad : int

UML Sequence diagram of the preliminary phase of glycolysis



An agent-based "calculator" for glycolysis



Place address: ip: 127.0.0.1 port: 9100
- servicesAgents(1)
- userAgents(402)
  - Lattasi10
  - LatticoDeidrogenasi4
  - TriosoFosfatoIsomerasi5
  - Gliceraldeide3FosfatoDeidrogenasi1
  - FosfoGliceratoChinasi7
  - AmidoFosforilasi18
  - GlicogenoFosforilasi7
  - GalattosioUnifosfatoUridilTrasferasi3
  - Galattochinasi13
  - PiruvatoChinasi17
  - Esochinasi14
  - Enolasi8
  - Saccarasi12
  - FosfoGlucoMutasi11
  - FosfoMannosioIsomerasi6
  - Enolasi14
  - Saccarasi4
  - Galattochinasi18
  - FosfoGliceratoChinasi5
  - FosfoFruttochinasi8
  - TriosoFosfatoIsomerasi6
  - FosfoGlucoIsomerasi12
  - FosfoGliceratoMutasi19
  - Galattochinasi1
  - TriosoFosfatoIsomerasi10
  - FosfoFruttochinasi5
  - PiruvatoChinasi16
  - Galattochinasi17

Carboidrati	Quantità
Glucosio	0

Altre Molecole	Quantità
NAD+	0

Ossigeno.. No

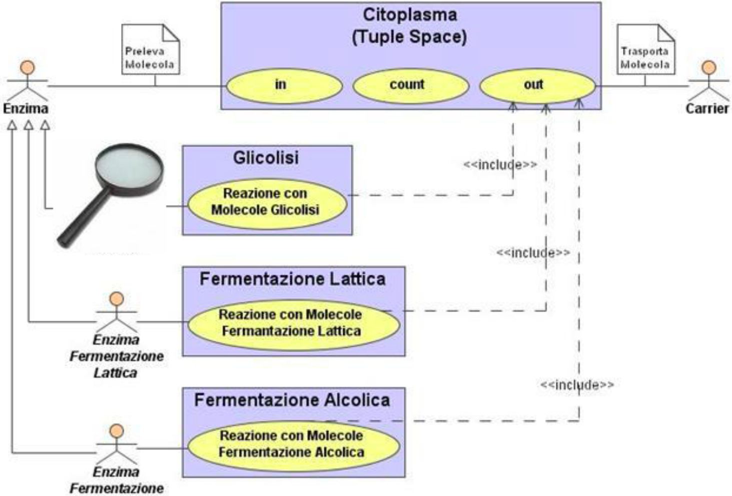
Tipo Fermentazione.................... Lattica

Q.tà Enzimi per Tipo.................................. 20

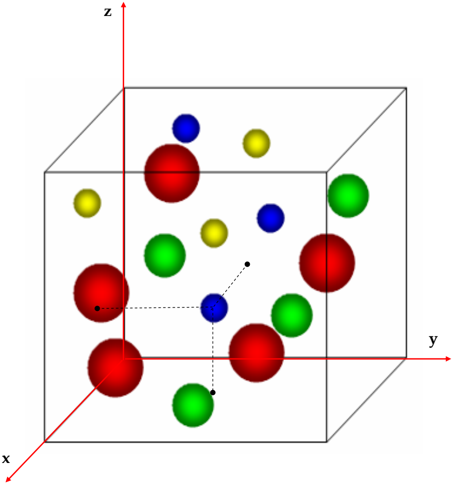
ATP........................	8
ADP........................	0
Acido Lattico.......	4
Etanolo..................	0
NAD+......................	4
NADH......................	0
CO2.........................	0
H2O.........................	4

```
Esochinasi6: OUT <ADP>
Esochinasi7: OUT <Glucosio6Fosfato>
FosfoGlucoIsomerasi0: OUT <Fruttosio6Fosfato>
FosfoFruttochinasi11: OUT <Fruttosio1_6difosfato>
FosfoFruttochinasi6: OUT <Fruttosio1_6difosfato>
Esochinasi7: OUT <ADP>
Aldolasi12: OUT <DiidrossiAcetoneFosfato>
Aldolasi17: OUT <DiidrossiAcetoneFosfato>
FosfoFruttochinasi6: OUT <ADP>
```

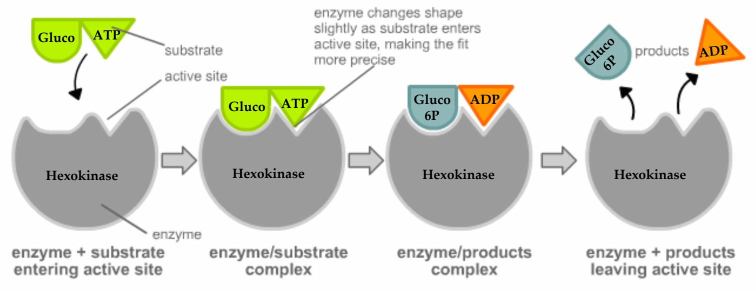
But ... Zooming-in



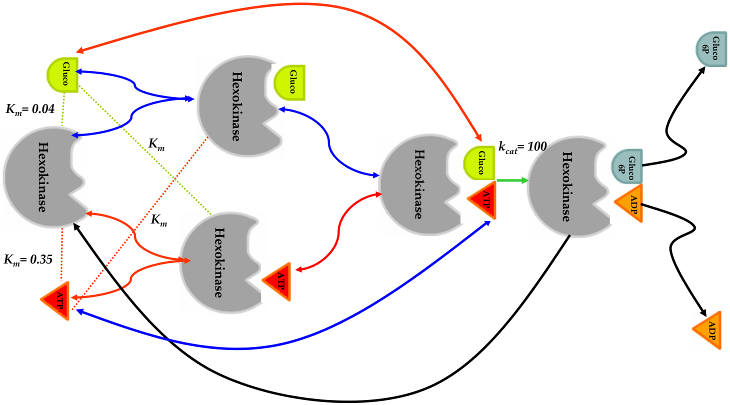
But ... molecules move

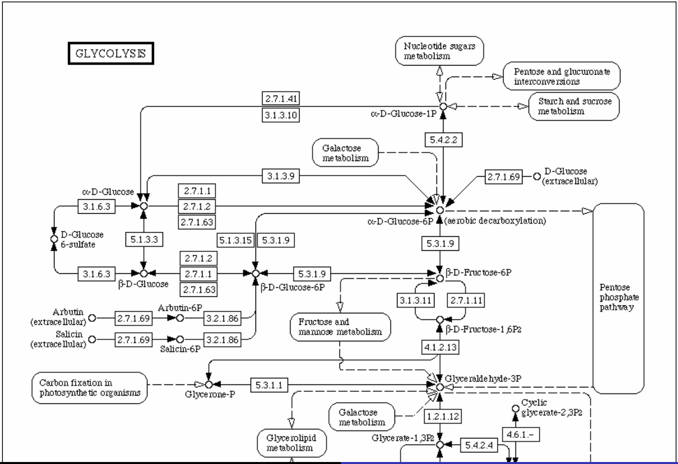


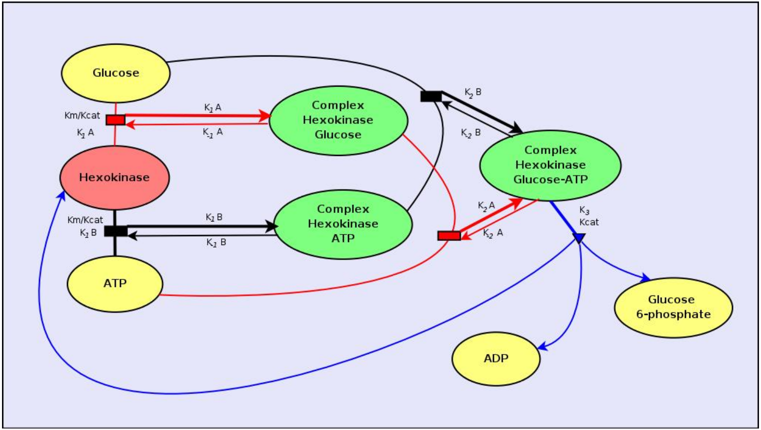
... and then interact by affinity


Gluco
ATP
substrate
active site
Hexokinase
enzyme
enzyme + substrate entering active site
Gluco
ATP
enzyme changes shape slightly as substrate enters active site, making the fit more precise
Hexokinase
enzyme/substrate complex
Gluco 6P
ADP
Hexokinase
enzyme/products complex
Gluco 6P
products
ADP
Hexokinase
enzyme + products leaving active site

... and then interact by affinity




GLYCOLYSIS
Nucleotide sugars metabolism
Pentose and glucuronate interconversions
Starch and sucrose metabolism
2.7.1.41
3.1.3.10
α-D-Glucose-1P
5.4.2.2
Galactose metabolism
3.1.3.9
2.7.1.69
D-Glucose (extracellular)
α-D-Glucose
2.7.1.1
3.1.6.3
2.7.1.2
2.7.1.63
α-D-Glucose-6P
(aerobic decarboxylation)
D-Glucose 6-sulfate
5.1.3.3
5.1.3.15
5.3.1.9
5.3.1.9
2.7.1.2
2.7.1.1
2.7.1.63
3.1.6.3
β-D-Glucose
5.3.1.9
β-D-Glucose-6P
β-D-Fructose-6P
Pentose phosphate pathway
3.1.3.11
2.7.1.11
Arbutin (extracellular)
2.7.1.69
Arbutin-6P
3.2.1.86
Salicin (extracellular)
2.7.1.69
Salicin-6P
3.2.1.86
Fructose and mannose metabolism
β-D-Fructose-1,6P2
4.1.2.13
Glyceraldehyde-3P
Carbon fixation in photosynthetic organisms
Glycerone-P
5.3.1.1
Galactose metabolism
1.2.1.12
Cyclic glycerate-2,3P2
4.6.1.-
Glycerolipid metabolism
Glycerate-1,3P2
5.4.2.4


Glucose
Km/Kcat
K_1 A
K_1 A
K_{-1} A
Complex Hexokinase Glucose
K_2 B
K_{-2} B
Complex Hexokinase Glucose-ATP
Hexokinase
Km/Kcat
K_1 B
K_1 B
K_1 B
Complex Hexokinase ATP
K_2 A
K_2 A
K_3
Kcat
ATP
Glucose 6-phosphate
ADP

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <sbml xmlns="http://www.sbml.org/sbml/level2"
xmlns:jd="http://www.sys-bio.org/sbml"
xmlns:sl2="http://projects.eml.org/bcb/sbml/level2" metaid="metaid_0000001"
level="2" version="1"> - <model metaid="metaid_0000002" id="Glycolysis_Nielsen"
name="Nielsen1998_Glycolysis"> - <notes> - <body
xmlns="http://www.w3.org/1999/xhtml">
  <p>Reference: Nielsen et al; Biophys. Chem. (1998) 72:49-62</p>
- <p>
  The reaction looks like this:
  <br />
  reaction_1: GLC + ATP -> F6P + ADP;
  <br />
  reaction_2: F6P + ATP -> FBP + ADP;
  <br />
  reaction_3: FBP => 2 * GAP;
  <br />
  reaction_4: GAP + NAD -> DPG + NADH;
  <br />
  reaction_5: DPG + ADP => PEP + ATP;
  <br />
  reaction_6: PEP + ADP -> PYR + ATP;
  <br />
  reaction_7: PYR -> ACA;
  <br />
  reaction_8: ACA + NADH => EtOH + NAD;
  <br />
  reaction_9: AMP + ATP => 2 * ADP;
  <br />
  reaction_10: F6P -> P
```

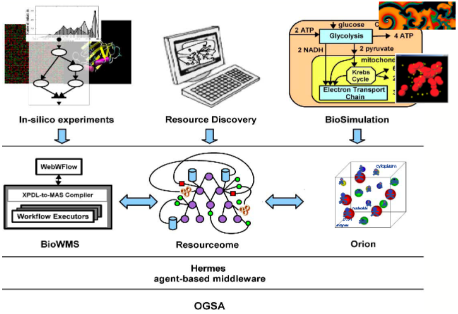
```
 - <reaction metaid="metaid_0000059" id="reaction_1"
reversible="false"> - <annotation> - <rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"> - <rdf:Description
rdf:about="#metaid_0000059"> - <dc:relation> - <rdf:Bag>
  <rdf:li rdf:resource="http://www.ebi.ac.uk/IntEnz/#EC 2.7.1.2" />
  <rdf:li rdf:resource="http://www.ebi.ac.uk/IntEnz/#EC 5.3.1.9" />
  <rdf:li rdf:resource="http://www.genome.jp/kegg/reaction/#R00771" />
  <rdf:li rdf:resource="http://www.genome.jp/kegg/reaction/#R00299" />
  <rdf:li rdf:resource="http://www.reactome.org/#70112" />
  </rdf:Bag>
  </dc:relation>
  </rdf:Description>
  </rdf:RDF>
  </annotation>
- {\bf <listOfReactants>
  <speciesReference species="GLC" />
  <speciesReference species="ATP" />
  </listOfReactants>
- <listOfProducts>
  <speciesReference species="F6P" />
  <speciesReference species="ADP" />
  </listOfProducts> }
- <kineticLaw> - <math xmlns="http://www.w3.org/1998/Math/MathML"> - <apply>
  <times />
 ...
  </kineticLaw>
```

Glycolysis

$$glucose + 2ATP + 4ADP + 2Pi + 2NADox \rightarrow$$
$$2pyruvate + 2ADP + 4ATP + 2NADrid + 2H_2O$$

Step	Reactions Substrates $\leadsto$ Products	Enzyme Catalyser	$\Delta G^{o'}$ in Kcal mol^{-1}
1	glucose, ATP $\rightleftharpoons$ glucose-6-phosphate, ADP, H^+	*hexokinase*	-4.0
2	glucose-6-phosphate $\rightleftharpoons$ fructose-6-phosphate	*phos . glu . isomerase*	+0.4
3	fructose-6-phosphate, ATP $\longrightarrow$ fructose1,6-bisphosphate, ADP, H^+	*phosphofructokinase*	-3.4
4	fructose1,6-bisphosphate $\rightleftharpoons$ dihydroxyacetone phosphate, glyceraldehyde 3-phosphate	*aldolase*	+3.7
5	dihydroxyacetone phosphate $\rightleftharpoons$ glyceraldehyde 3-phosphate	*trio . phos . isomerase*	+1.8
6	glyceraldehyde 3-phosphate, Pi, NAD^+ $\rightleftharpoons$ 1,3-bisphosphoglycerate, NADH, H^+	*glyceraldehyde* *3phos . dehydro . se*	+1.5
7	1,3-bisphosphoglycerate, ADP $\rightleftharpoons$ 3-phosphoglycerate, ATP	*phosphoglyceratekinase*	-4.5
8	3-phosphoglycerate $\rightleftharpoons$ 2-phosphoglycerate	*phosphoglyceromutase*	+1.1
9	2-phosphoglycerate $\rightleftharpoons$ phosphoenolpyruvate, H_2O	*enolase*	+0.4
10	phosphoenolpyruvate, ADP, H^+ $\rightleftarrows$ enolpyruvate, ATP	*pyruvatekinase*	-7.5
11	enolpyruvate $\longrightarrow$ pyruvate	— — —	

LITBIO Project

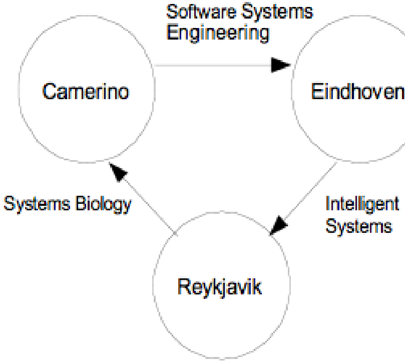


Erasmus Mundus Master Course in *Formal Methods for Systems Engineering* (FOMSE)

The master curricula are specified by three tracks. Every student chooses a track, which defines the mobility plan (between two sites, at the end of the first year) of the student. The choice is driven by the specialization/thesis/research interest specified in the name of the track. Every track has a minimum number of ECTS (30) covering the following Basic Topics:

- Process Algebra
- Distributed and Concurrent Computation
- Theory of Computation

Erasmus Mundus Partnership



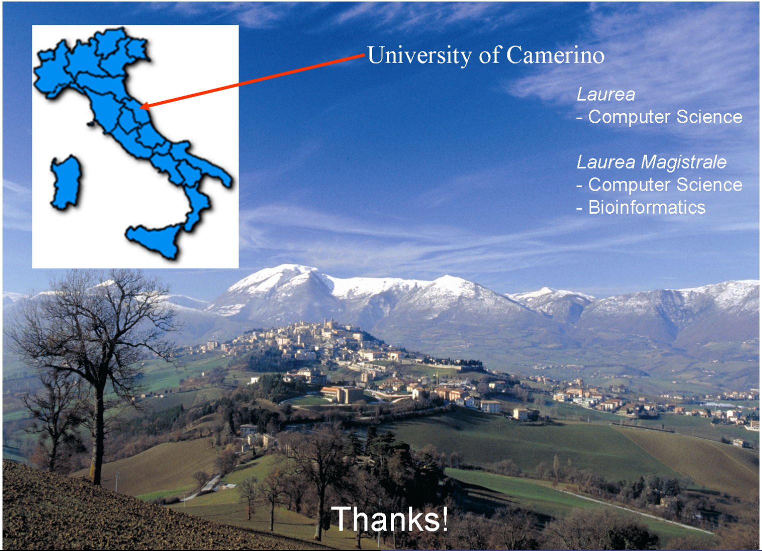
Master Tracks

- **Systems Biology**: Year 1 Reykjavik (Basic Topics + Courses in Bioinformatics and Systems Biology + free choiches). Year 2 Camerino (Specialization + Research Credits + Thesis)
- **Software Systems Engineering**: Year 1 Camerino (Basic Topics + Service Oriented Architectures + free choices). Year 2 Eindhoven (Specialization + Research Credits + Thesis)
- **Intelligent Systems** Year 1 Eindhoven (Basic Topics + Algorithms for Intelligent Systems + free choices). Year 2 Reykjavik (Specialization + Research Credits + Thesis)

Coordinator
Flavio Corradini

Researchers
Diletta Cacciagrano
Nicola Cannata
Rosario Culmone
Maria Rita Di Berardini
Emanuela Merelli
Luca Tesei

Phd Students
Ezio Bartocci
Francesco De Angelis
Francesca Piersigilli
Barbara Re
Oliviero Riganelli
Leonardo Vito
Roberta Alfieri

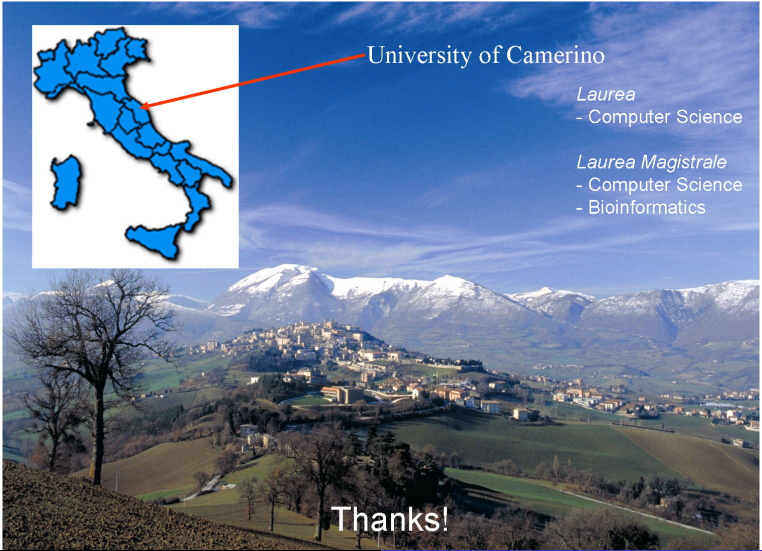


University of Camerino

Laurea
- Computer Science

Laurea Magistrale
- Computer Science
- Bioinformatics

Thanks!



University of Camerino

Laurea
- Computer Science

Laurea Magistrale
- Computer Science
- Bioinformatics

Thanks!