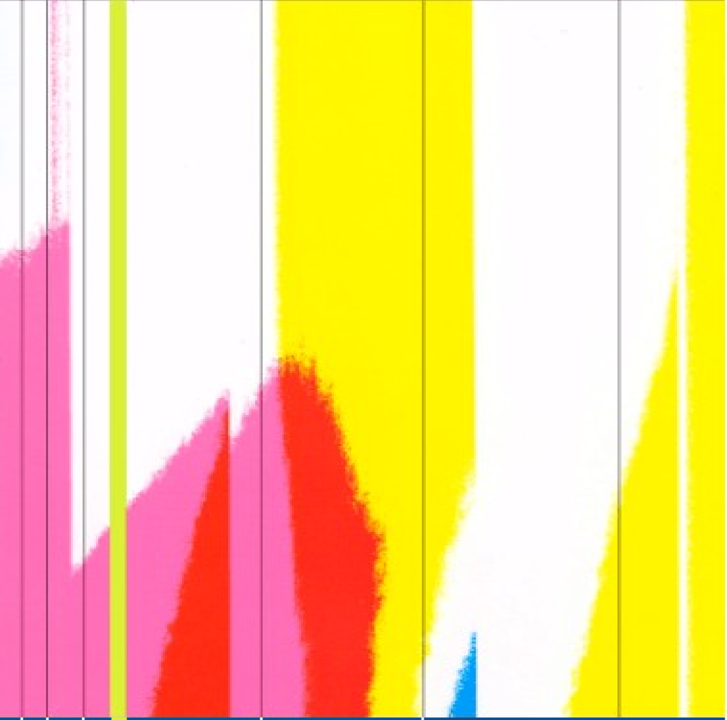


BioRica :

A platform to describe and simulate multi-models systems in biology.

http://biorica.gforge.inria.fr/

April 2011 – Alice Garcia

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
BORDEAUX - SUD-OUEST

Plan

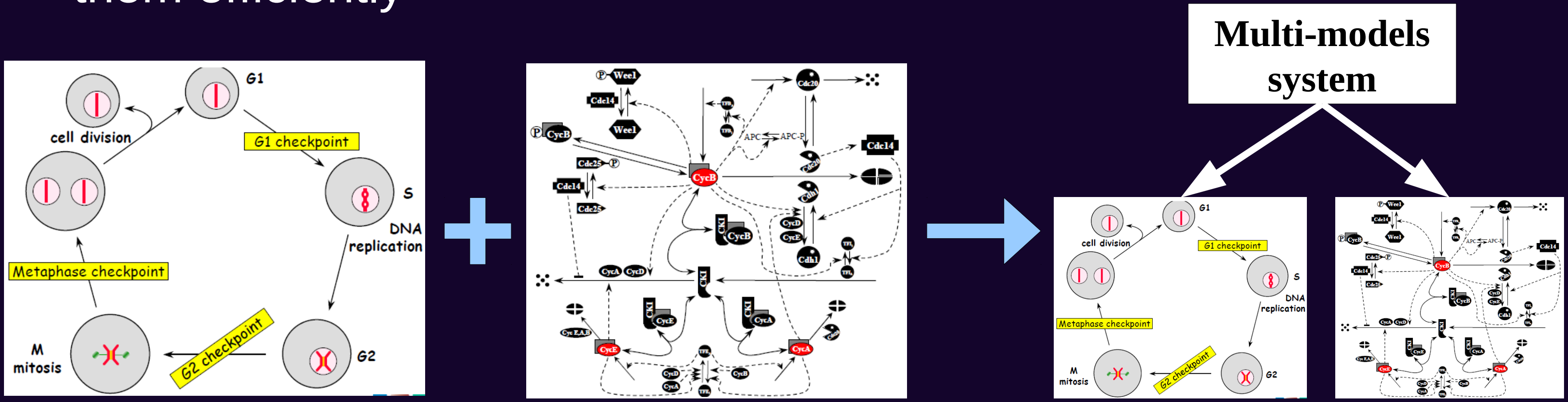
- BioRica : Why ? What is it ?
  - Context and interest
- BioRica to describe multi-models system
  - The BioRica syntax
- BioRica to simulate multi-models system
  - The BioRica simulation
- BioRica 1st release
  - Conception, development, tests and documentation
- BioRica future versions
  - Version in development and needs for next versions
- Examples and demo !

Plan

- BioRica : Why ? What is it ?
  - Context and interest
- BioRica to describe multi-models system
  - The BioRica syntax
- BioRica to simulate multi-models system
  - The BioRica simulation
- BioRica 1st release
  - Conception, development, tests and documentation
- BioRica future versions
  - Version in development and needs for next versions
- Examples and demo !

Context

- A lot of models in biology
  - To describe chemical process, biological process,...
  - Validated experimentally
  - Not developed in order to be reused
- Need to compose existing models hierarchically and to simulate them efficiently



What is BioRica ? (1/3)

- BioRica :
  - Software to describe and simulate multi-models in biology
  - Propose a semantic for models composition

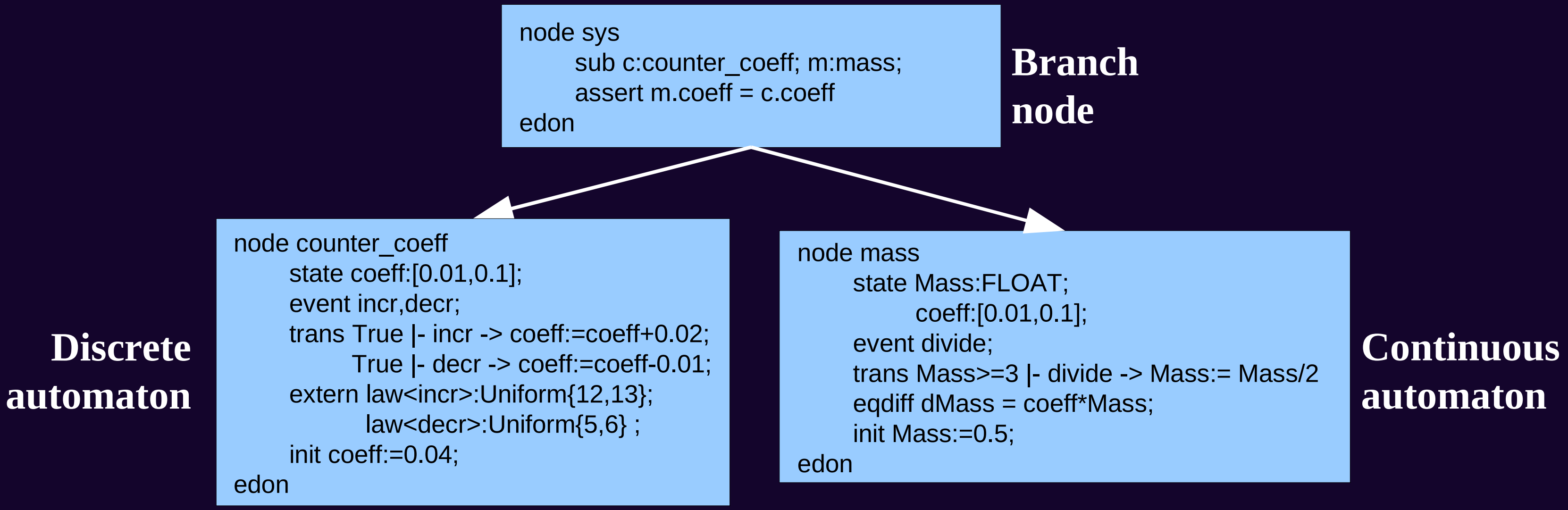
  - Adaptation for biology systems of AltaRica (Point and Rauzy, 1999)
    - AltaRica is a language used to describe functional and non-functional behaviors on critical systems
    - AltaRica is developed for industrial applications

  - Previous version of BioRica
    - Developed as an internal tool in the team (Soueidan et al, 2007)
    - The original software was completely rewrote for the new version

What is BioRica ? (2/3)

•Description of multi-models systems :

- Description of a syntax
  – Based on « AltaRica DataFlow Language Syntax » (Rauzy, 2002)
- Multi-models system represented by a hierarchy of nodes
  – 1 model = 1 node
  – Hierarchy with leaf nodes (automaton) and branches nodes



```
node sys
        sub c:counter_coeff; m:mass;
        assert m.coeff = c.coeff
edon
```

Branch node

```
node counter_coeff
        state coeff:[0.01,0.1];
        event incr,decr;
        trans True |- incr -> coeff:=coeff+0.02;
                 True |- decr -> coeff:=coeff-0.01;
        extern law<incr>:Uniform{12,13};
                   law<decr>:Uniform{5,6} ;
        init coeff:=0.04;
edon
```

Discrete automaton

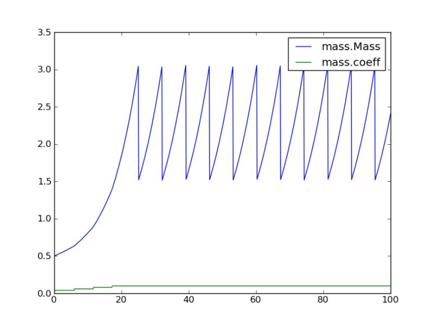
```
node mass
        state Mass:FLOAT;
                 coeff:[0.01,0.1];
        event divide;
        trans Mass>=3 |- divide -> Mass:= Mass/2
        eqdiff dMass = coeff*Mass;
        init Mass:=0.5;
edon
```

Continuous automaton

What is BioRica ? (3/3)

- Simulation of multi-models systems :
  - Work with continuous models and discrete times
  - Simulate a system
    - Resolve differential equations in continuous models
    - Change states from discrete transitions
  - Obtain results
    - Print variables values at each time interval, to standard output
    - Display plots

```
time    c.coeff    m.coeff    m.Mass     event
0       0.04       0.04       0.5        .
0.1     0.04       0.04       0.52125    .
                      ...
5.034   0.06       0.06       0.61005    c->incr
5,134   0.06       0.06       0,61437    .
                      ...
24,45   0.06       0.06       3.00072    m->divide
```



Plan

- BioRica : Why ? What is it ?
  - Context and interest
- BioRica to describe multi-models system
  - The BioRica syntax
- BioRica to simulate multi-models system
  - The BioRica simulation
- BioRica 1st release
  - Conception, development, tests and documentation
- BioRica future versions
  - Version in development and needs for next versions
- Examples and demo !

The BioRica syntax

- Multi models system represented by hierarchy of nodes
    - 1 model = 1 node
    - Hierarchy with leaf nodes (automaton) and branches nodes

- BioRica components
    - Constant
    - Domain
    - Formula
    - Node

The BioRica syntax : components

Constants

Definition domains

Formula to simplify equations

Node

Node

```
const C1 = 45; C2 = 2;
formula F = 0.56452 *s +C1;
domain D = [0,40]; D2 = {2,5,C1,48}

node A
      state s:D;
      flow f:{1,2};
      event ev1,ev2;
      trans
            f=1 |- ev1 -> s:=s+C1;
            f=2 |- ev2 -> s:=s+C2 ;
      eqdiff ds = F;
      extern law<ev1>:Exponential{0.3};
      init s:=10;
edon

node main
      sub a1,a2:A;
      assert a1.s=a2.s;
      sync <a1.ev1,a2.ev1>;
edon
```

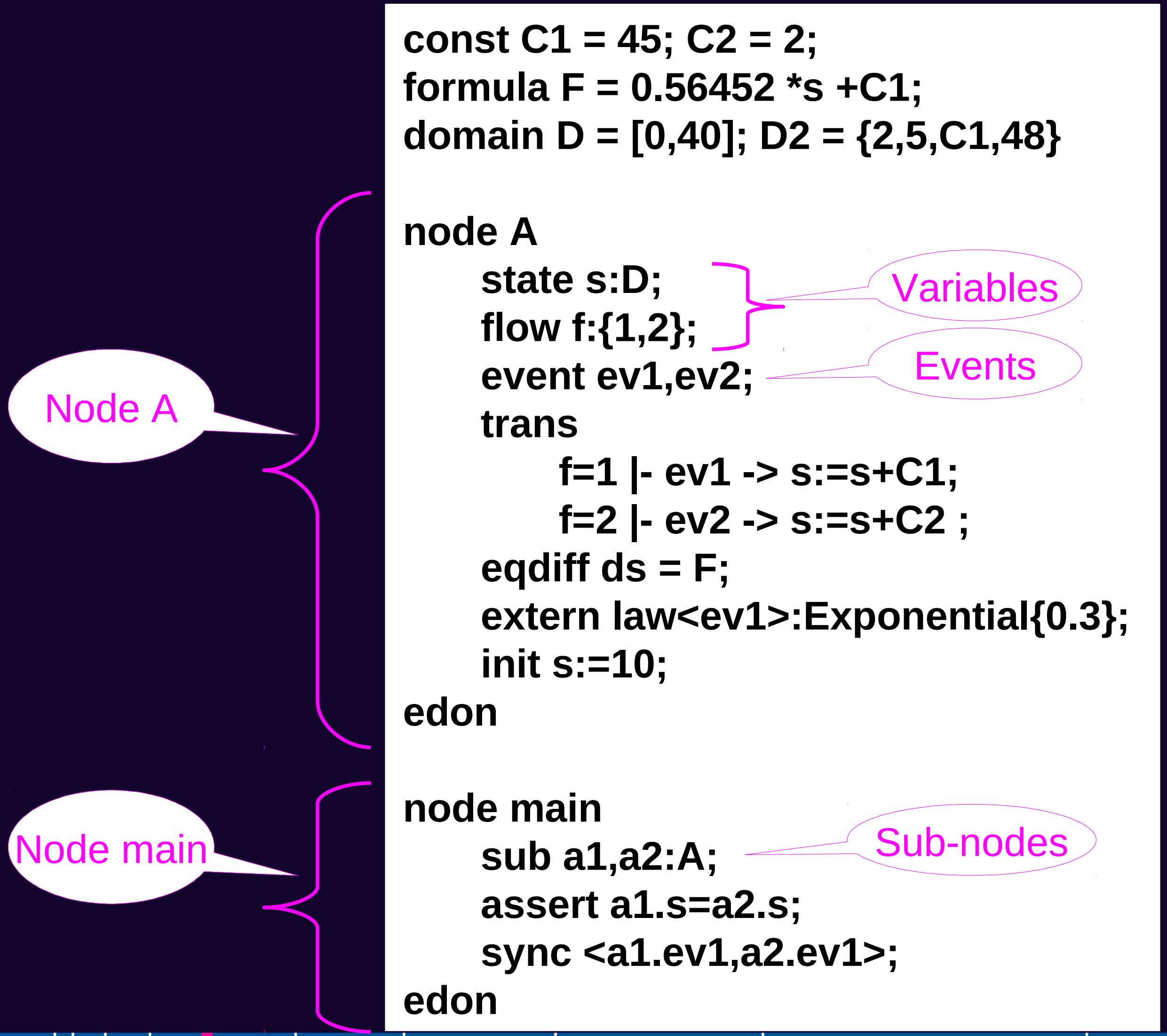
The BioRica syntax : node

Node A

Node main

```
const C1 = 45; C2 = 2;
formula F = 0.56452 *s +C1;
domain D = [0,40]; D2 = {2,5,C1,48}

node A
    state s:D;
    flow f:{1,2};
    event ev1,ev2;
    trans
        f=1 |- ev1 -> s:=s+C1;
        f=2 |- ev2 -> s:=s+C2 ;
    eqdiff ds = F;
    extern law<ev1>:Exponential{0.3};
    init s:=10;
edon

node main
    sub a1,a2:A;
    assert a1.s=a2.s;
    sync <a1.ev1,a2.ev1>;
edon
```

The BioRica syntax : node elements



```
const C1 = 45; C2 = 2;
formula F = 0.56452 *s +C1;
domain D = [0,40]; D2 = {2,5,C1,48}

node A
        state s:D;
        flow f:{1,2};
        event ev1,ev2;
        trans
                f=1 |- ev1 -> s:=s+C1;
                f=2 |- ev2 -> s:=s+C2 ;
        eqdiff ds = F;
        extern law<ev1>:Exponential{0.3};
        init s:=10;
edon

node main
        sub a1,a2:A;
        assert a1.s=a2.s;
        sync <a1.ev1,a2.ev1>;
edon
```

The BioRica syntax : node fields (1/4)

```
const C1 = 45; C2 = 2;
formula F = 0.56452 *s +C1;
domain D = [0,40]; D2 = {2,5,C1,48}

node A
      state s:D;
      flow f:{1,2};
      event ev1,ev2;
      trans
            f=1 |- ev1 -> s:=s+C1;
            f=2 |- ev2 -> s:=s+C2 ;
      eqdiff ds = F;
      extern law<ev1>:Exponential{0.3};
      init s:=10;
edon

node main
      sub a1,a2:A;
      assert a1.s=a2.s;
      sync <a1.ev1,a2.ev1>;
edon
```

Node A

Initialization

Node main

Assertion to describe invariant on variables

The BioRica syntax : node fields (2/4)

```
const C1 = 45; C2 = 2;
formula F = 0.56452 *s +C1;
domain D = [0,40]; D2 = {2,5,C1,48}

node A
      state s:D;
      flow f:{1,2};
      event ev1,ev2;
      trans
            f=1 |- ev1 -> s:=s+C1;
            f=2 |- ev2 -> s:=s+C2 ;
      eqdiff ds = F;
      extern law<ev1>:Exponential{0.3};
      init s:=10;
edon

node main
      sub a1,a2:A;
      assert a1.s=a2.s;
      sync <a1.ev1,a2.ev1>;
edon
```

Node A

External directive
for events
delay or weight

Node main

Synchronization
to synchronize events

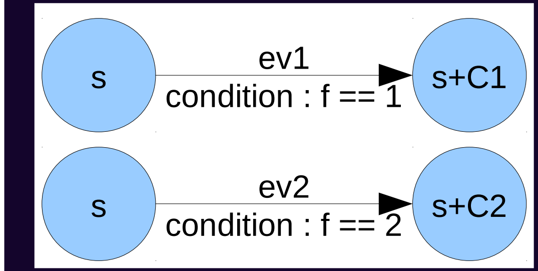
The BioRica syntax : node fields (3/4)

```
const C1 = 45; C2 = 2;
formula F = 0.56452 *s +C1;
domain D = [0,40]; D2 = {2,5,C1,48}

node A
      state s:D;
      flow f:{1,2};
      event ev1,ev2;
      trans
            f=1 |- ev1 -> s:=s+C1;
            f=2 |- ev2 -> s:=s+C2 ;
      eqdiff ds = F;
      extern law<ev1>:Exponential{0.3};
      init s:=10;
edon

node main
      sub a1,a2:A;
      assert a1.s=a2.s;
      sync <a1.ev1,a2.ev1>;
edon
```

Node A

Node main

Transitions for changes of state



The BioRica syntax : node fields (4/4)

```
const C1 = 45; C2 = 2;
formula F = 0.56452 *s +C1;
domain D = [0,40]; D2 = {2,5,C1,48}

node A
      state s:D;
      flow f:{1,2};
      event ev1,ev2;
      trans
            f=1 |- ev1 -> s:=s+C1;
            f=2 |- ev2 -> s:=s+C2 ;
      eqdiff ds = F;
      extern law<ev1>:Exponential{0.3};
      init s:=10;
edon

node main
      sub a1,a2:A;
      assert a1.s=a2.s;
      sync <a1.ev1,a2.ev1>;
edon
```

Node A

Node main

Differential equations

$$\frac{ds}{dt}=0.56452*s+45$$

Plan

- BioRica : Why ? What is it ?
  - Context and interest
- BioRica to describe multi-models system
  - The BioRica syntax
- BioRica to simulate multi-models system
  - The BioRica simulation
- BioRica 1st release
  - Conception, development, tests and documentation
- BioRica future versions
  - Version in development and needs for next versions
- Examples and demo !

The BioRica simulation (1/2)

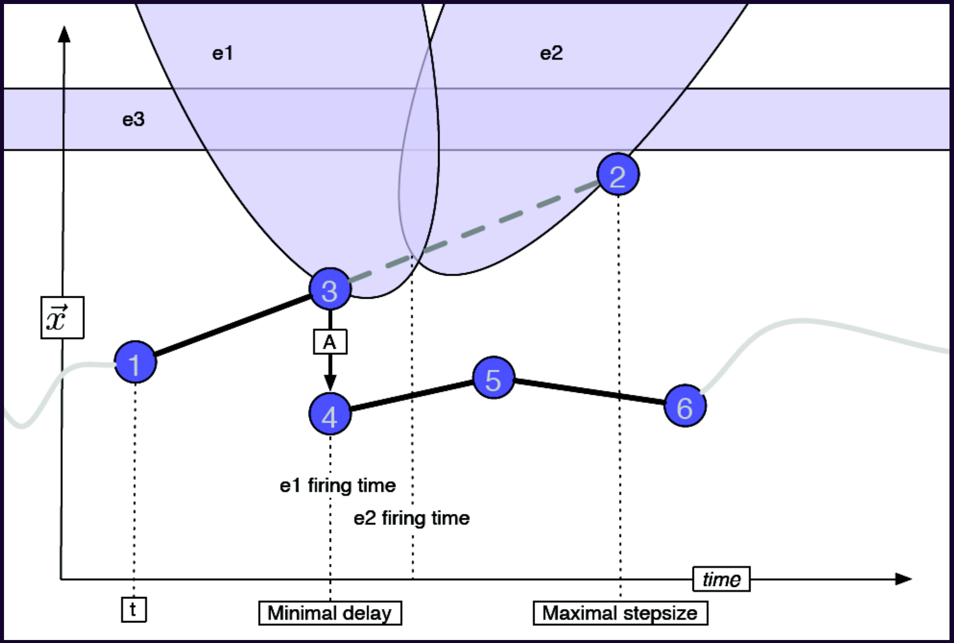
- Events scheduling
  - Use a scheduler to deal with events
  - Events temporized by random probability law

  - Search in all the sub-nodes reachable events
  - Select events with minimal delay
  - For concomitant events
    - Choose randomly via weighting

  - Memorize waiting events and manage their delay

The BioRica simulation (2/2)

- Simulation combining continuous and discrete models
  - From H. Soueidan's thesis (2009)
  - The simulation deals with continuous times and discrete events and runs back the time to consider discrete interruptions



- Resolution move from t (***point 1***) and maximal step (***point 2***).
- Events *e1*, *e2* guards are satisfied
- Simulator turns back at the initial activation point of the event (***point 3***)
- Discrete transition *A* is passed (***point 4***)

Plan

- BioRica : Why ? What is it ?
  - Context and interest
- BioRica to describe multi-models system
  - The BioRica syntax
- BioRica to simulate multi-models system
  - The BioRica simulation
- BioRica 1st release
  - Conception, development, tests and documentation
- BioRica future versions
  - Version in development and needs for next versions
- Examples and demo !

BioRica version 0.9 (1/3)

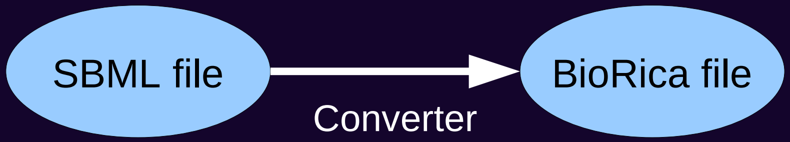
- Project web-page : http://biorica.gforge.inria.fr/

- Command line software
- Developed in Python
- Compatible with Windows, Unix and Mac OS X

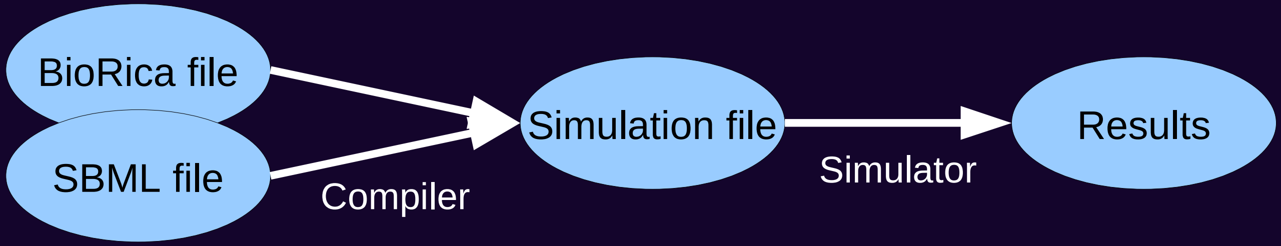
- License CeCILL (similar to license GNU GPL)
- Depot at the APP (Agence pour la Protection des Programmes) asking by the INRIA

BioRica version 0.9 (2/3)

- Software in 3 parts
  - A SBML (Systems Biology Markup Language) converter
    - Main program “biorica_convert_SBML”



  - A compiler
    - Main program “biorica_compil”
  - A simulator
    - Main program “biorica_simul”



BioRica version 0.9 (3/3)

- Development
  - In Python
  - With many tests (unit tests, validation tests, ...)
  - Development and tests as one goes along on each package independently
- Manuals
  - Maintenance manual from comments in code
  - User guide with syntax description and user cases
- Version progress
  - Version functional and tested
  - Version validated on complex examples
    - Working with Rodrigo Assar in our team to validate the software
    - Validation on hybrid models and wine fermentation models

Plan

- BioRica : Why ? What is it ?
  - Context and interest
- BioRica to describe multi-models system
  - The BioRica syntax
- BioRica to simulate multi-models system
  - The BioRica simulation
- BioRica 1st release
  - Conception, development, tests and documentation
- BioRica future versions
  - Version in development and needs for next versions
- Examples and demo !

New functionality for future versions

- Improve differential equations
  - Currently the software supports ODE (Ordinary Differential Equations)
  - Extend to allow other kind of differential equations
    - DAE (Differential Algebraic Equations)
    - Partial differential equations

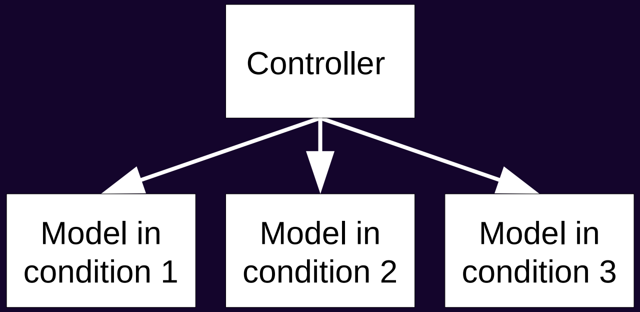
- Add user interface
  - We expect BioRica to be used by biologists and modelers, which are not necessarily familiar with command line programs
  - Need to design a user interface to make easier to create and simulate models

Plan

- BioRica : Why ? What is it ?
    - Context and interest
- BioRica to describe multi-models system
    - The BioRica syntax
- BioRica to simulate multi-models system
    - The BioRica simulation
- BioRica 1st release
    - Conception, development, tests and documentation
- BioRica future versions
    - Version in development and needs for future versions
- Examples and demo !

Example 1

- The SCHITTLER model
  - It models the dynamics of the differentiation from precursor cells into bone or cartilage cells in humans
  - A stimulus is responsible of the differentiation
- Interest :
  - Improving the stimulus model (how and when each differentiation is stimulated) we can improve the combined model.
- A hybrid system
  - A continuous model : SCHITTLER
    - With 3 different state conditions
  - A discrete model : the stimulus
    - Controller to change the states



Controller

Model in condition 1

Model in condition 2

Model in condition 3

Example 1

- The SCHITTLER model : a continuous automaton

$$\begin{cases} \dfrac{dxd}{dt} = \dfrac{0.2 \ast xd^2 + 0.5}{10 + zd + 0.1 \ast xd^2} - 0.1 \ast xd \\ \dfrac{dxo}{dt} = \dfrac{0.1 \ast xo^2 + 1 + zo}{1 + 0.1 \ast xo^2 + 0.1 \ast xc^2 + 0.5 \ast xd^2} - 0.1 \ast xo \\ \dfrac{dxc}{dt} = \dfrac{0.1 \ast xc^2 + 1 + zc}{1 + 0.1 \ast xc^2 + 0.1 \ast xo^2 + 0.5 \ast xd^2} - 0.1 \ast xc \end{cases}$$

$$xd, xo, xc \in \mathbb{R}$$
$$zd, zo, zc \in [0,1]$$

$$\begin{gathered} xd[0]=12 \\ xo[0]=0 \\ xc[0]=0 \\ zd[0]=0 \\ zo[0]=0 \\ zc[0]=0 \end{gathered}$$

```
node DIFF



edon
```

Example 1

- The SCHITTLER model : a continuous automaton

$$\begin{cases} \dfrac{dxd}{dt} = \dfrac{0.2*xd^2+0.5}{10+zd+0.1*xd^2} - 0.1*xd \\ \dfrac{dxo}{dt} = \dfrac{0.1*xo^2+1+zo}{1+0.1*xo^2+0.1*xc^2+0.5*xd^2} - 0.1*xo \\ \dfrac{dxc}{dt} = \dfrac{0.1*xc^2+1+zc}{1+0.1*xc^2+0.1*xo^2+0.5*xd^2} - 0.1*xc \end{cases}$$

$$xd, xo, xc \in \mathbb{R}$$
$$zd, zo, zc \in [0,1]$$

$$\begin{gathered} xd[0]=12 \\ xo[0]=0 \\ xc[0]=0 \\ zd[0]=0 \\ zo[0]=0 \\ zc[0]=0 \end{gathered}$$

```
node DIFF
      state xd,xo,xc:FLOAT;
              zd,zo,zc:[0,1];
      init
            xd:=12,xo:=0,xc:=0,zd:=0,zo:=0,zc:=0;




edon
```

Example 1

- The SCHITTLER model : a continuous automaton

$$\begin{cases}\dfrac{dxd}{dt}=\dfrac{0.2*xd^2+0.5}{10+zd+0.1*xd^2}-0.1*xd\\[2ex]\dfrac{dxo}{dt}=\dfrac{0.1*xo^2+1+zo}{1+0.1*xo^2+0.1*xc^2+0.5*xd^2}-0.1*xo\\[2ex]\dfrac{dxc}{dt}=\dfrac{0.1*xc^2+1+zc}{1+0.1*xc^2+0.1*xo^2+0.5*xd^2}-0.1*xc\end{cases}$$

$$xd\,,xo\,,xc\in\mathbb{R}$$
$$zd\,,zo\,,zc\in[0,1]$$

$$\begin{gathered}xd[0]=12\\xo[0]=0\\xc[0]=0\\zd[0]=0\\zo[0]=0\\zc[0]=0\end{gathered}$$

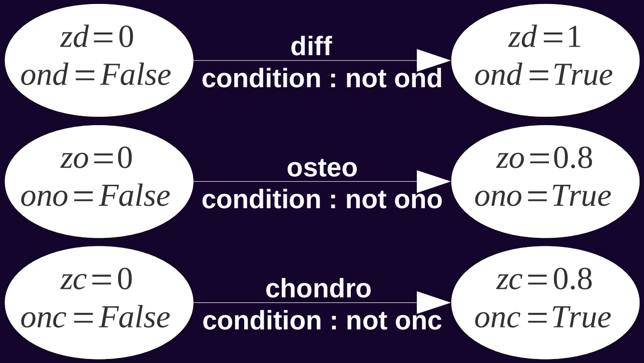
```
node DIFF
      state xd,xo,xc:FLOAT;
              zd,zo,zc:[0,1];
      init
            xd:=12,xo:=0,xc:=0,zd:=0,zo:=0,zc:=0 ;
      eqdiff
            dxd = (0.2*xd^2+0.5)/(10+zd+0.1*xd^2)-0.1*xd;
            dxo = (0.1*xo^2+1+zo)/(1+0.1*xo^2+0.1*xc^2+0.5*xd^2)-0.1*xo;
            dxc = (0.1*xc^2+1+zc)/(1+0.1*xc^2+0.1*xo^2+0.5*xd^2)-0.1*xc;
edon
```

Example 1

- The stimulus controller : a discrete automaton

$zd, zo, zc \in [0,1]$
$ond, ono, onc \in 0,1$

$$
\begin{gathered}
zd[0]=0\\
zo[0]=0\\
zc[0]=0\\
ond[0]=False\\
ono[0]=False\\
onc[0]=False
\end{gathered}
$$



Events diff, osteo and chondr with exponential law of parameters 0.01, 0.002 and 0.001

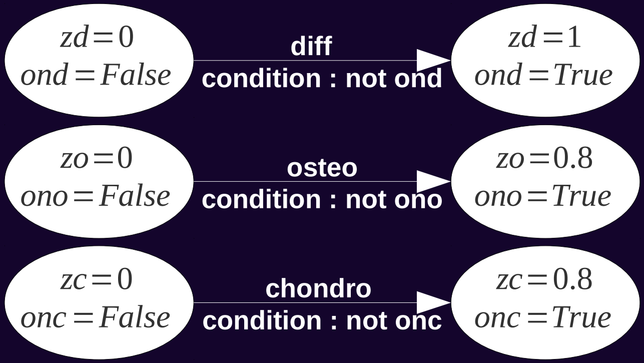
```
node STIMULUS






edon
```

Example 1

- The stimulus controller : a discrete automaton

$$zd, zo, zc \in [0,1]$$
$$ond, ono, onc \in 0,1$$

$$zd[0]=0$$
$$zo[0]=0$$
$$zc[0]=0$$
$$ond[0]=False$$
$$ono[0]=False$$
$$onc[0]=False$$



$zd=0$, $ond=False$ —diff, condition : not ond→ $zd=1$, $ond=True$

$zo=0$, $ono=False$ —osteo, condition : not ono→ $zo=0.8$, $ono=True$

$zc=0$, $onc=False$ —chondro, condition : not onc→ $zc=0.8$, $onc=True$

Events diff, osteo and chondr with exponential law of parameters 0.01, 0.002 and 0.001

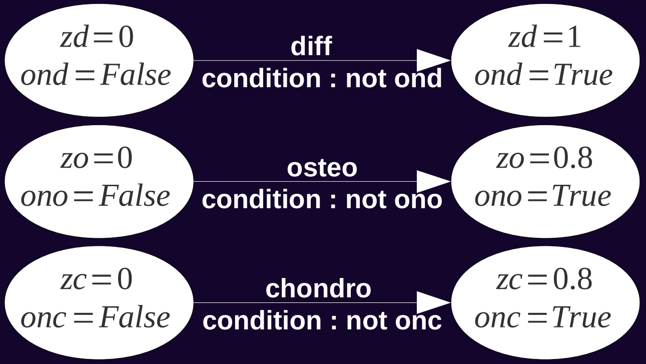
```
node STIMULUS
      state zd,zo,zc:[0,1];ond,ono,onc:BOOL;



      init zd:=0,zo:=0,zc:=0,ond:=False,ono:=False,onc:=False;



edon
```

Example 1

- The stimulus controller : a discrete automaton

$$zd\,,zo\,,zc\in[0,1]$$
$$ond\,,ono\,,onc\in 0,1$$

$$zd[0]=0$$
$$zo[0]=0$$
$$zc[0]=0$$
$$ond[0]=False$$
$$ono[0]=False$$
$$onc[0]=False$$



$zd=0$, $ond=False$ → diff, condition : not ond → $zd=1$, $ond=True$

$zo=0$, $ono=False$ → osteo, condition : not ono → $zo=0.8$, $ono=True$

$zc=0$, $onc=False$ → chondro, condition : not onc → $zc=0.8$, $onc=True$

Events diff, osteo and chondr with exponential law of parameters 0.01, 0.002 and 0.001

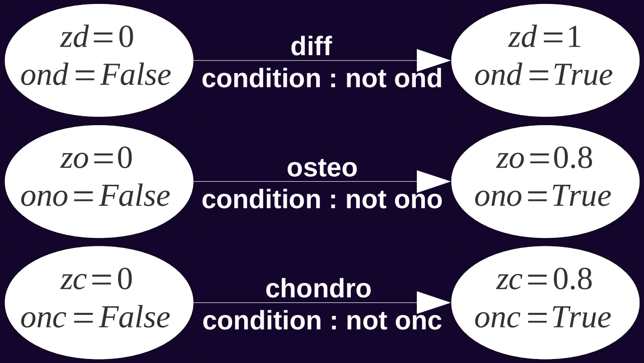
```
node STIMULUS
     state zd,zo,zc:[0,1];ond,ono,onc:BOOL;
     event diff,osteo,chondro;
     trans
          not ond  |- diff -> zd:=1,ond:=True;
          not ono  |- osteo -> zo:=0.8,ono:=True ;
          not onc  |- chondro -> zc:=0.8,onc:=True;
     init zd:=0,zo:=0,zc:=0,ond:=False,ono:=False,onc:=False;




edon
```

Example 1

- The stimulus controller : a discrete automaton

$zd, zo, zc \in [0,1]$
$ond, ono, onc \in 0,1$

$zd[0]=0$
$zo[0]=0$
$zc[0]=0$
$ond[0]=False$
$ono[0]=False$
$onc[0]=False$



$zd=0$, $ond=False$ — diff, condition : not ond → $zd=1$, $ond=True$

$zo=0$, $ono=False$ — osteo, condition : not ono → $zo=0.8$, $ono=True$

$zc=0$, $onc=False$ — chondro, condition : not onc → $zc=0.8$, $onc=True$

Events diff, osteo and chondr with exponential law of parameters 0.01, 0.002 and 0.001

```
node STIMULUS
      state zd,zo,zc:[0,1];ond,ono,onc:BOOL;
      event diff,osteo,chondro;
      trans
            not ond  |- diff -> zd:=1,ond:=True;
            not ono  |- osteo -> zo:=0.8,ono:=True ;
            not onc  |- chondro -> zc:=0.8,onc:=True;
      init zd:=0,zo:=0,zc:=0,ond:=False,ono:=False,onc:=False;
      extern
            law<diff>:Exponential{0.01};
            law<osteo>:Exponential{0.002};
            law<chondro>:Exponential{0.001};
edon
```

Example 1

- The complete system

Model DIFF with variables zd, zo, zc, ...
Model STIMULUS with variables zd, zo, zc, ...

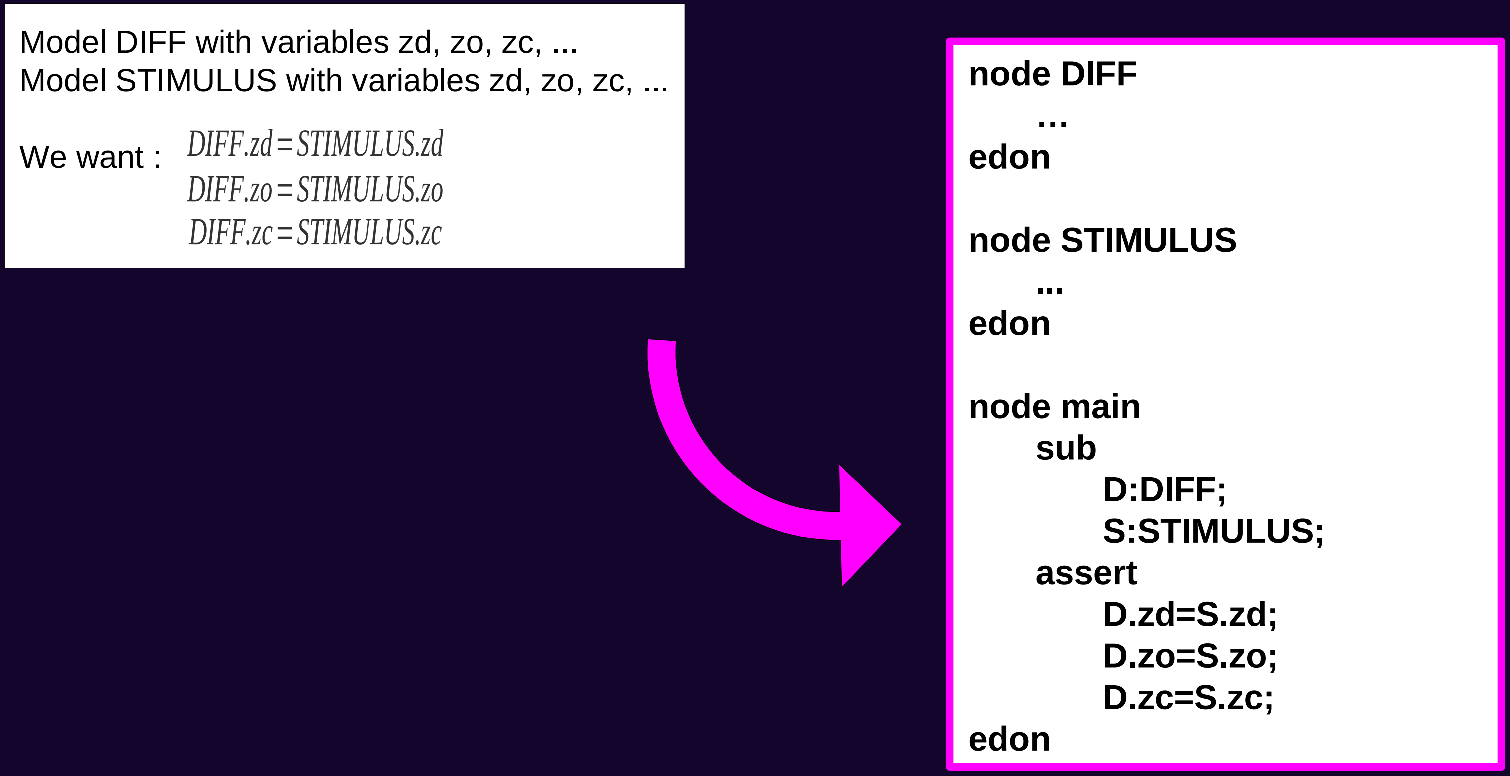
We want :

$$
\begin{aligned}
DIFF.zd &= STIMULUS.zd \\
DIFF.zo &= STIMULUS.zo \\
DIFF.zc &= STIMULUS.zc
\end{aligned}
$$

```
node DIFF
      ...
edon

node STIMULUS
      ...
edon

node main





edon
```

Example 1

- The complete system



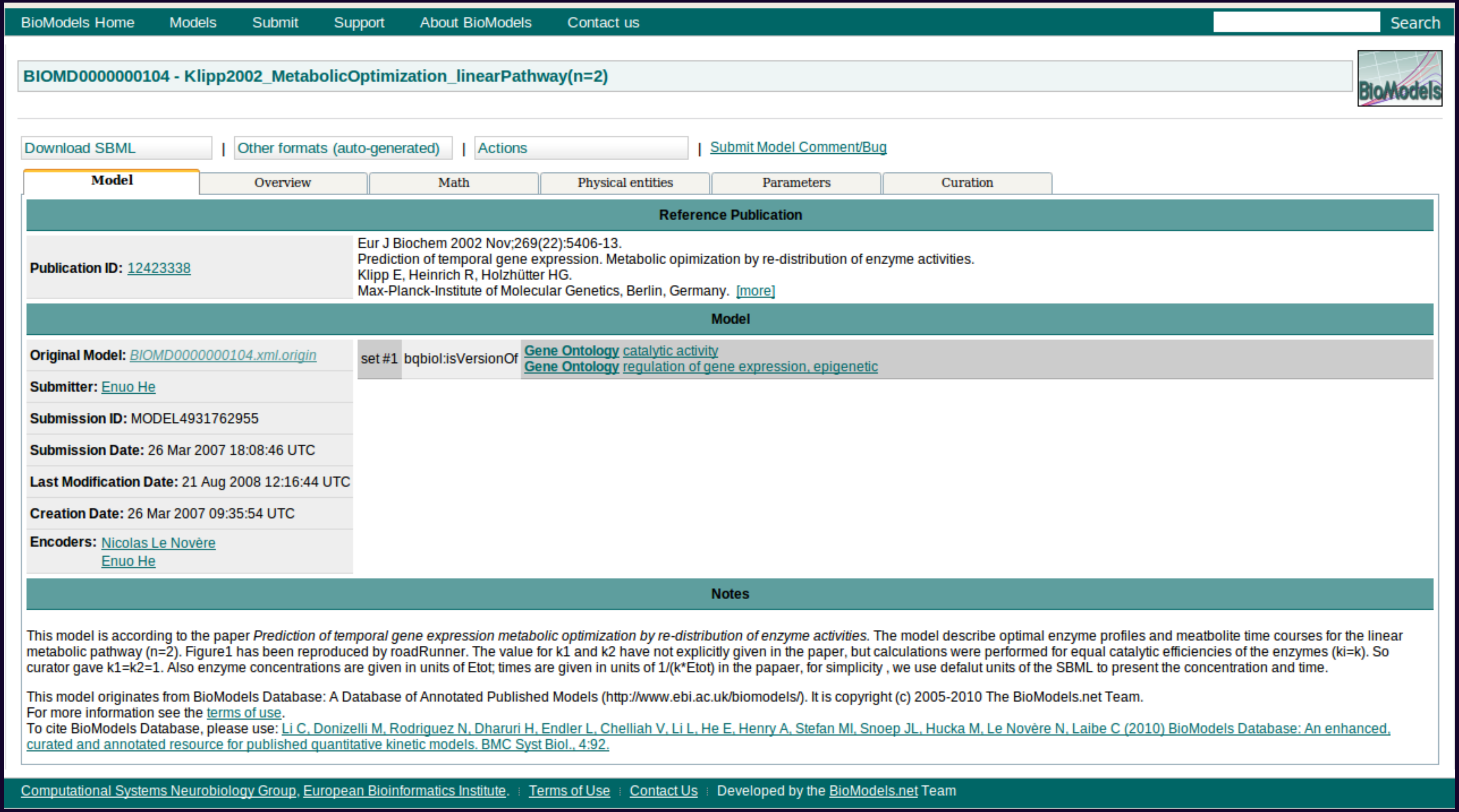
Model DIFF with variables zd, zo, zc, ...
Model STIMULUS with variables zd, zo, zc, ...

We want :

$$DIFF.zd = STIMULUS.zd$$
$$DIFF.zo = STIMULUS.zo$$
$$DIFF.zc = STIMULUS.zc$$

```
node DIFF
      ...
edon

node STIMULUS
      ...
edon

node main
      sub
            D:DIFF;
            S:STIMULUS;
      assert
            D.zd=S.zd;
            D.zo=S.zo;
            D.zc=S.zc;
edon
```

Example 2 : BioModels modèle 104



BIOMD0000000104 - Klipp2002_MetabolicOptimization_linearPathway(n=2)

Download SBML | Other formats (auto-generated) | Actions | Submit Model Comment/Bug
Model
Overview
Math
Physical entities
Parameters
Curation

Reference Publication

Publication ID: 12423338	Eur J Biochem 2002 Nov;269(22):5406-13. Prediction of temporal gene expression. Metabolic opimization by re-distribution of enzyme activities. Klipp E, Heinrich R, Holzhütter HG. Max-Planck-Institute of Molecular Genetics, Berlin, Germany. [more]

Model

Original Model: *BIOMD0000000104.xml.origin*

Submitter: Enuo He

Submission ID: MODEL4931762955

Submission Date: 26 Mar 2007 18:08:46 UTC

Last Modification Date: 21 Aug 2008 12:16:44 UTC

Creation Date: 26 Mar 2007 09:35:54 UTC

Encoders: Nicolas Le Novère
Enuo He

set #1	bqbiol:isVersionOf	**Gene Ontology** catalytic activity **Gene Ontology** regulation of gene expression, epigenetic

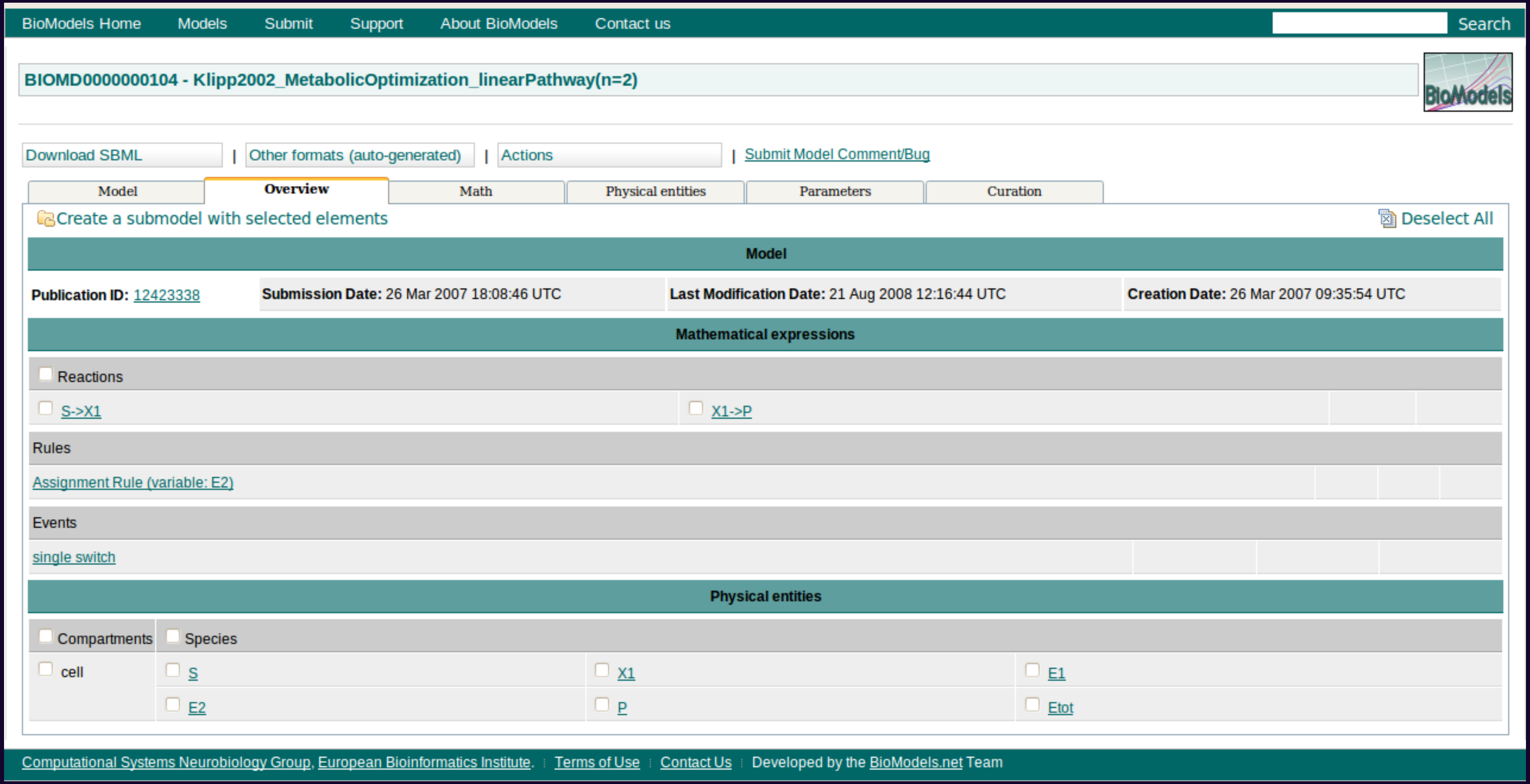
Notes

This model is according to the paper *Prediction of temporal gene expression metabolic optimization by re-distribution of enzyme activities.* The model describe optimal enzyme profiles and meatbolite time courses for the linear metabolic pathway (n=2). Figure1 has been reproduced by roadRunner. The value for k1 and k2 have not explicitly given in the paper, but calculations were performed for equal catalytic efficiencies of the enzymes (ki=k). So curator gave k1=k2=1. Also enzyme concentrations are given in units of Etot; times are given in units of 1/(k*Etot) in the papaer, for simplicity , we use defalut units of the SBML to present the concentration and time.

This model originates from BioModels Database: A Database of Annotated Published Models (http://www.ebi.ac.uk/biomodels/). It is copyright (c) 2005-2010 The BioModels.net Team.
For more information see the terms of use.
To cite BioModels Database, please use: Li C, Donizelli M, Rodriguez N, Dharuri H, Endler L, Chelliah V, Li L, He E, Henry A, Stefan MI, Snoep JL, Hucka M, Le Novère N, Laibe C (2010) BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Syst Biol., 4:92.

Computational Systems Neurobiology Group, European Bioinformatics Institute. | Terms of Use | Contact Us | Developed by the BioModels.net Team

Example 2 : BioModels modèle 104



BIOMD0000000104 - Klipp2002_MetabolicOptimization_linearPathway(n=2)

Download SBML | Other formats (auto-generated) | Actions | Submit Model Comment/Bug
Model
Overview
Math
Physical entities
Parameters
Curation
Create a submodel with selected elements
Deselect All

Model

Publication ID: 12423338	**Submission Date:** 26 Mar 2007 18:08:46 UTC	**Last Modification Date:** 21 Aug 2008 12:16:44 UTC	**Creation Date:** 26 Mar 2007 09:35:54 UTC

Mathematical expressions

□ Reactions

□ S->X1 □ X1->P

Rules

Assignment Rule (variable: E2)

Events

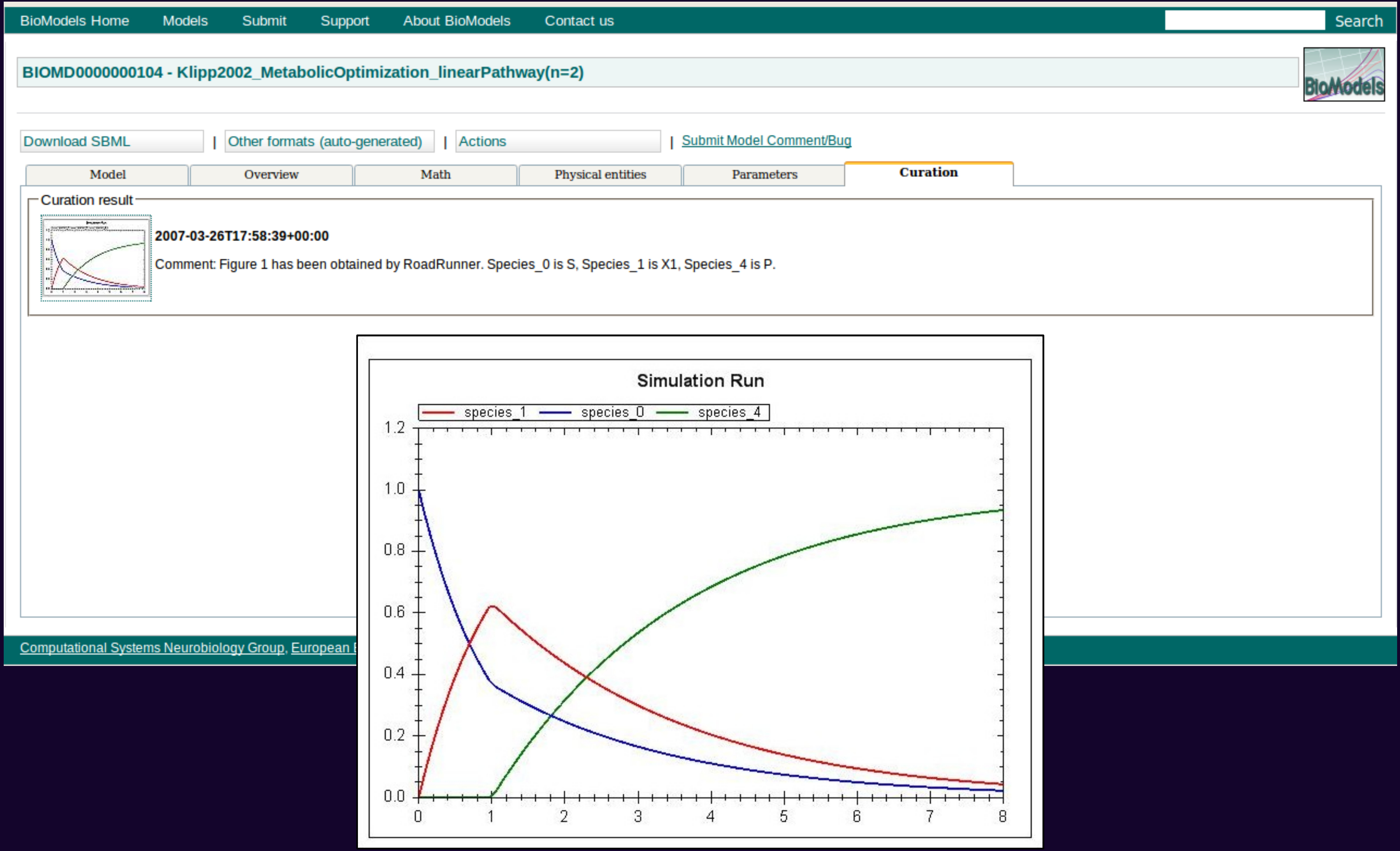
single switch

Physical entities

□ Compartments	□ Species		
□ cell	□ S	□ X1	□ E1
	□ E2	□ P	□ Etot

Computational Systems Neurobiology Group, European Bioinformatics Institute. Terms of Use Contact Us Developed by the BioModels.net Team

Example 2 : BioModels modèle 104



BIOMD0000000104 - Klipp2002_MetabolicOptimization_linearPathway(n=2)

Download SBML | Other formats (auto-generated) | Actions | Submit Model Comment/Bug
Model
Overview
Math
Physical entities
Parameters
Curation

Curation result

2007-03-26T17:58:39+00:00

Comment: Figure 1 has been obtained by RoadRunner. Species_0 is S, Species_1 is X1, Species_4 is P.

Thank you for your attention !!

Questions ?