Cell Illustrator Tutorial I

Basic Concepts

Human Genome Center,
Institute of Medical Science,
University of Tokyo

support@genomicobject.net

Contents

[1] This document is mainly translated from Japanese manual of Masao Nagasaki by Enua Jeong.

1 PREFACE

Cell Illustrator^{TM}(CI) is a software tool that enables biologists to model, elucidate and simulate complex biological processes and systems [13]. It allows researchers to model metabolic pathways, signal transduction cascades, gene regulatory pathways, microRNA regulatory networks, as well as dynamic interactions of various biological entities, such as genomic DNA, mRNA and proteins. Cell Illustrator models are used to visualize biopathways, interpret experimental data and test hypotheses. In addition, it provides researchers diagrams of publication quality, and simulation result charts. Cell Illustrator has been successfully utilized to model biological pathways like the gene regulation of circadian rhythms in *Drosophila* and *mouse*, glycolytic pathway of *E. coli*, the boundary formation by notch signaling in *Drosophila*, the apoptosis induced by *Fas* ligand, *p53* related pathways in mammals, and the cell fate of gustatory neurons with microRNA regulations in *C. elegans* [1, 2, 3, 4, 5, 6, 7, 8, 9, 15]. This manual describes version 3.0 of Cell Illustrator (CI3.0). Using the graphical user interface environment, researchers can create and simulate their own molecular pathway models. Some of the key features of Cell Illustrator are:

(a) Pathway construction and visualization - constructing models graphically by simply dragging-and-dropping entities representing bio-components in the biological pathways.

(b) Pathways simulation - building mathematical models of biochemical reactions.

(c) Simulation animation - Cell Illustrator's simulation results can be saved into an animation script file and then used as an input to Cell Animator - a module that produces high quality presentation of the studied mechanism's dynamic behavior.

(d) Models import - importing previously created models, or sub-models from public or proprietary libraries and databases like KEGG [11, 14]. Pathway models in other formats, e.g. SBML and CellML, can be also imported into CI3.0.

(e) Models distribution - a pathway model in CI3.0 can be saved as a XML (named Cell System Markup Language [16]) file with a simulation log. Cell Illustrator Player, which is a free software distributed together with CI3.0, can load the model from the log file, and replay the simulation. Thus, once you create a model on CI3.0, you can freely distribute the model to any person who has interest in your model.

2 INTRODUCTION TO CELL ILLUSTRATOR

2.1 Installation and Execution of Cell Illustrator

Cell Illustrator is available for Windows, Linux, Unix and Mac OS X platforms. In order to run Cell Illustrator, Java (JRE) version 1.5.0 or higher needs to be installed. Java Runtime Environments (JRE) are available for download at `http://java.sun.com/downloads/`. Alternatively, for Windows and Linux platforms, you may execute the Cell Illustrator installation package bundled with a Java installer. Cell Illustrator installation packages also offer an option to install a beta version of Cell Animator, a program for animating biological pathways.

2.1.1 Windows

Install Java-bundled version

Step 1 Unpack CI3.0_wj.zip in a temporary directory (to unpack it, use WinZip (`http://www.winzip.com`).

Step 2 Double-click on CI3.0_wj.exe to install Cell Illustrator. Java Virtual Machine (JVM) will also be installed.

Step 3 Launch Cell Illustrator from the Start menu.

Install non Java-bundled version (recommended)

Step 1 Unpack CI3.0_w.zip in a temporary directory.

Step 2 Double-click on CI3.0_w.exe to install Cell Illustrator.

Step 3 Launch Cell Illustrator from the Start menu.

2.1.2 Linux

Install Java-bundled version

Step 1 Copy CI3.0_lj.bin into a temporary directory .

Step 2 Execute "> chmod +x CI3.0_lj.bin" (JRE will be installed at the same time).

Step 3 Execute "> CI3.0_lj.bin"

To invoke Cell Illustrator, go into the newly created directory and run "> ./CI".

Install non Java-bundled version (recommended)

Step 1 Copy CI3.0_l.bin into a temporary directory.

Step 2 Execute "> chmod +x CI3.0_l.bin".

Step 3 Execute "> CI3.0_lj.bin".

To invoke Cell Illustrator, go into CI directory and execute "> ./CI".

2.1.3 Unix (excluding Linux)

Install non Java-bundled version (If you need JRE, download JRE from `http://java.sun.com`)

Step 1 Copy CI3.0_u.jar into a temporary directory.

Step 2 Execute "> chmod +x CI3.0_u.jar".

Step 3 Execute "> java -jar CI3.0_u.jar".

To invoke Cell Illustrator, go to the CI folder and execute "> ./CI".

2.1.4 Mac OS X (Classic is not supported)

Install non Java (JRE) version (the latest version of JRE can be executed by selecting update)

Step 1 Move CI3.0_m.tgz into the newly created directory.

Step 2 Double-click on CI3.0_m.tgz, start installation.

Step 3 After the installation, a Cell Illustrator icon is generated.

To invoke Cell Illustrator, double-click on the Cell Illustrator icon on the Desktop. To install Cell Animator on Mac OS X, move CellAnimator.dmg into the same directory thus Cell Illustrator is installed in, and double-click CellAnimator.dmg. The Cell Animator icon will appear on the Desktop.

2.2 License Term

If you're going to run Cell Illustrator, you need to submit a request for a license file from GNI. Please copy the obtained license file to the directory where Cell Illustrator is installed. For example, on Windows, Cell Illustrator is installed to "`C:\Program Files\GNI\Cell Illustrator`" if the default installation settings are used. The license file can be bound to your machine, or time-limited.

To obtain a hardware-bound license, you need to send your machine hardware identifier (Hardware ID) as follows:

Step 1 Select the **About Cell Illustrator** option from the **Help** menu.

Step 2 In the **About Cell Illustrator** dialog, select the "Serial Number" tab and find the "Hardware ID" item.

Step 3 Click the "Copy" button in the lower left corner of the dialog. The details will be copied to the clipboard.

Step 4 Paste this text into your message and send it to the Cell Illustrator sales and technical support team.

If the license is expired, this method cannot be used again. In this case, the use the "Copy" button in the dialog acknowledging that the license term is expired, and send the hardware ID to your CI vendor.

3 CELL ILLUSTRATOR ELEMENTS

3.1 Basic Concepts

Cell Illustrator uses a method based on the Petri net formalism (Hybrid Functional Petri net with extension, HFPNe) to model and simulate biological pathways. HFPNe inherits and enhances the properties of a Petri net such as:

- Pathways can be modeled easily by drawing a graph.
- The pathways are simulatable without any additional work.
- Models of different complexity can be built with the same constituent elements.

Biological pathways are built using three elements: entity, process, and connector. Figure 1 summarizes the elements used in Cell Illustrator.

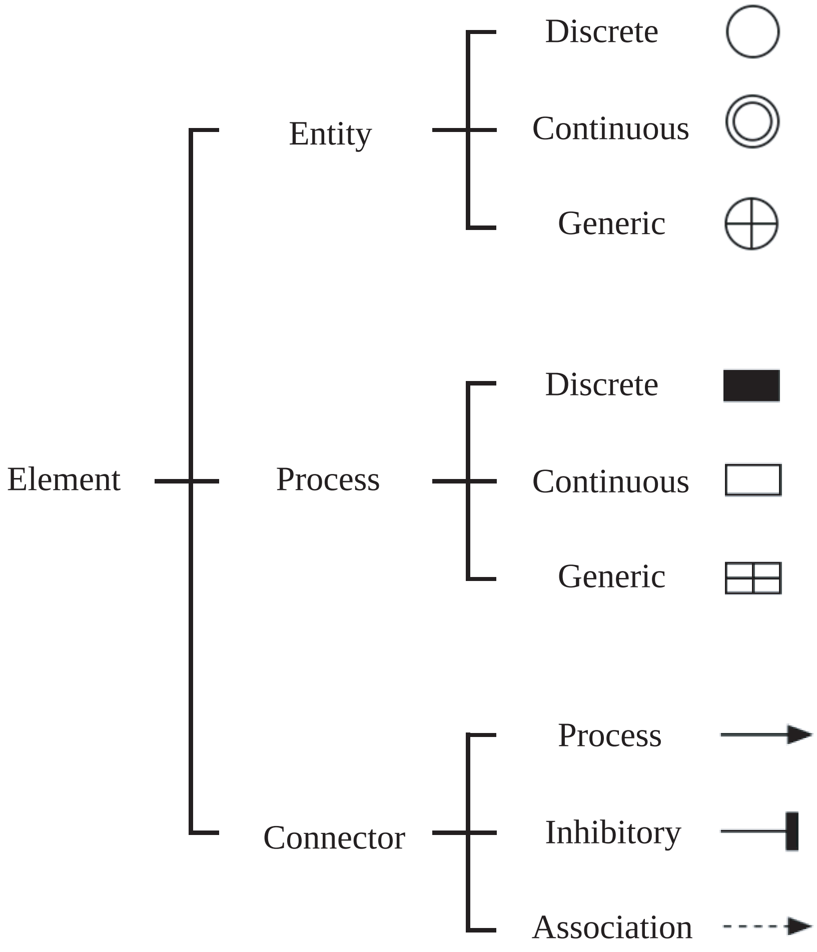


Figure 1

3.2 Entity

A biological object, such as protein or mRNA, is represented by an entity. There are three types of entities: *discrete*, *continuous* and *generic* (Figure 2). Using entities with different properties makes models more relevant to a biological pathway.

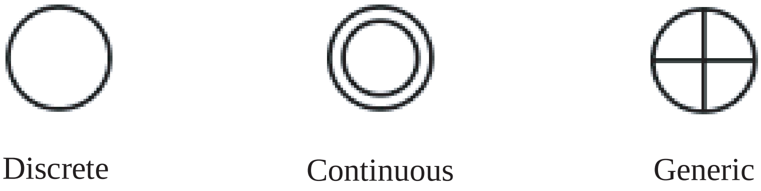


Figure 2

An entity has three properties. When you add an entity to the canvas in Cell Illustrator, three editable properties: *Name*, *Variable*, and *Initial Value* appear in the upper-left, upper-right, and lower-left corners, respectively, as shown in Figure 3. *Name* is indexed with the prefix 'e', *Variable* with 'm', and *Initial Value* has a default value of zero.

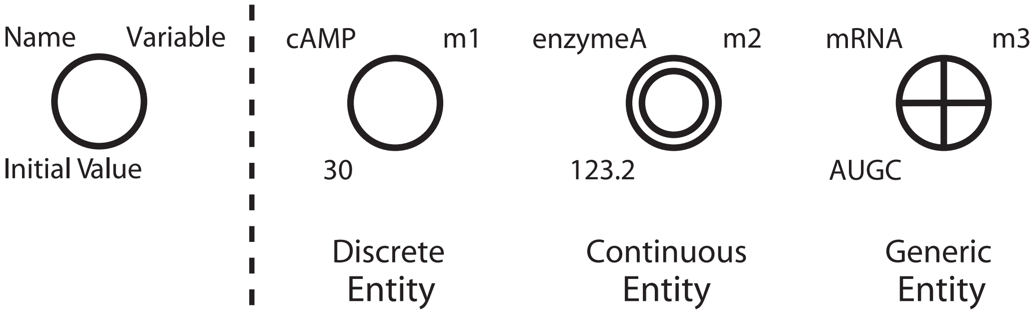


Figure 3

There are three ways to rename/modify a default name/value of a property:

1. Double-click the text you want to modify in the canvas and directly type a new name (Figure 4).
2. Click the entity in the canvas, select **Element Settings** menu option from the **Dialog** menu, and modify the value of *Name*/ *Variable*.
3. Click the entity in the canvas, select **Element Lists** menu option from the **Dialog** menu and then **Entity** tab, and modify the value of *Name*/ *Variable* by double-clicking the corresponding property.

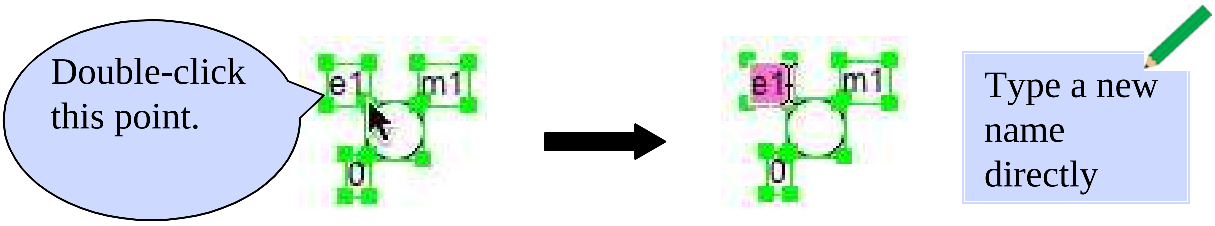


Figure 4

3.3 Process

Processes are used to represent protein responses, activation, binding, development, translocation, transcription, translation, etc.

Like entities, there are also three types of process: *discrete*, *continuous*, and *generic*. The *discrete* and *continuous* processes can be used together to represent phenomenon/response depending on the relationship between entities.

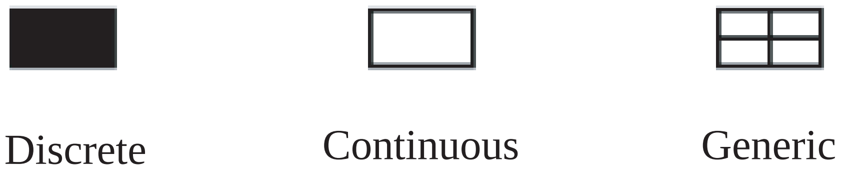


Figure 5

When a process is added to the canvas, three properties of the process: *Name*, *Delay*, and *Speed* appear in the upper-left, lower-left, and lower-right corners, respectively, as shown in Figure 6. *Name* is indexed with a prefix ‘p’, and both *Delay* and *Speed* have the default values of zero. *Delay* can be used only for *discrete* processes.

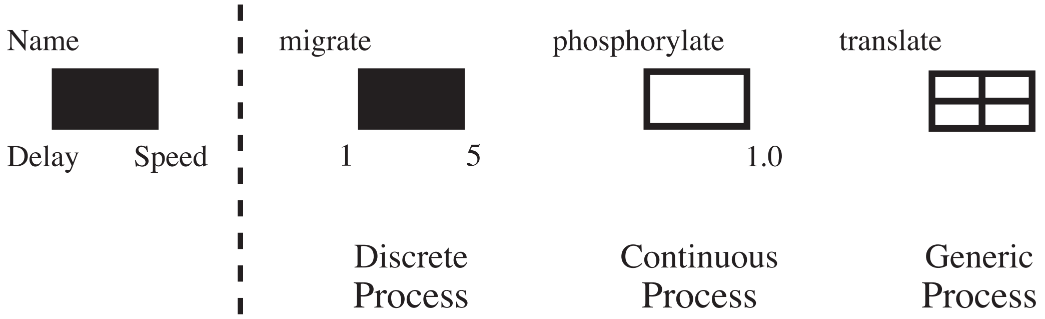


Figure 6

There are three ways to rename/modify a default name/value of a property:

1. Double-click the text you want to modify in the canvas and directly type a new name (Figure 7).
2. Click the process in the canvas, select **Element Settings** menu option from the **Dialog** menu, and modify the value of *Name*/*Delay*/*Speed*.
3. Click the process in the canvas, select **Element Lists** menu option from the **Dialog** menu and then **Process** tab, and modify the value of *Name*/*Delay*/*Speed* by double-clicking the corresponding property.

3.4 Connector

Connectors are used to connect entities with processes (Note that you cannot directly connect two entities or two processes).

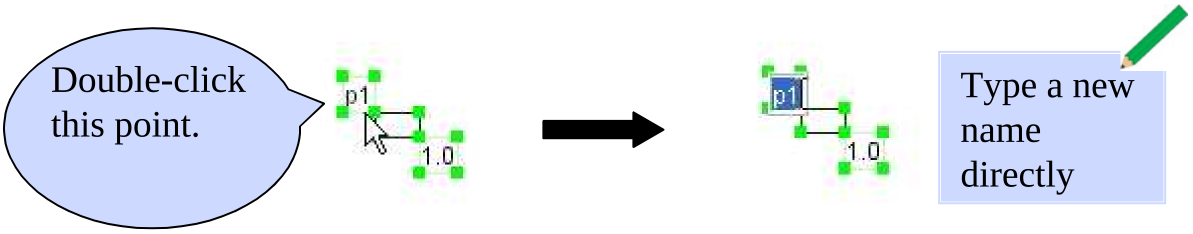


Figure 7

A connector from an entity to a process is called an *input connector*. One from a process to an entity is called an *output connector*. However, depending on the type of the entity and the process, some connections are not allowed (this will be explained later in Table 1).

There are three types of connectors: *process*, *inhibitory*, and *association*. All three types of connectors can be used as *input connectors*, while only a *process* connector can be used as an *output connector*.

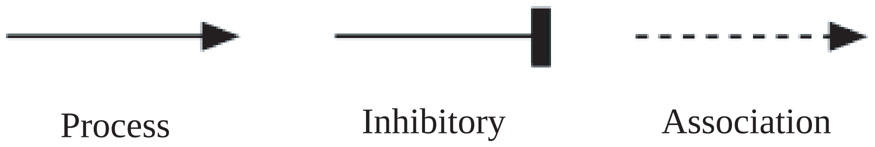


Figure 8

A *process* connector is used for any type of interaction. To repress any phenomenon/reaction, an *inhibitory* connector is used. *Association* connectors are used to assist/help some processes. The usage of connectors will be described in Section 7.

When a connector is added to the canvas, two properties of the connector, *Name* and *Threshold* appear above and below the connector line as shown in Figure 9. *Name* is indexed with the prefix ‘c’, and *Threshold* has the default value of zero.

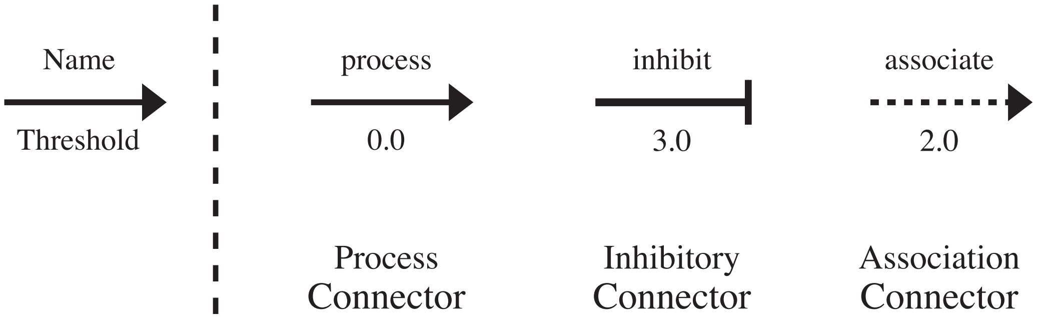


Figure 9

There are three ways to rename/modify the default names/values of the two properties:

1. Double-click the text you want to modify in the canvas and directly type a new name (Figure 10).

2. Click the connector, select the **Element Settings** menu option from the **Dialog** menu, and change the contents of the *Name/Threshold* fields.

3. Click the connector, select **Element Lists** menu option from the **Dialog** menu and then **Connector** tab, and give a new value to *Name/Threshold* by double-clicking the corresponding property.

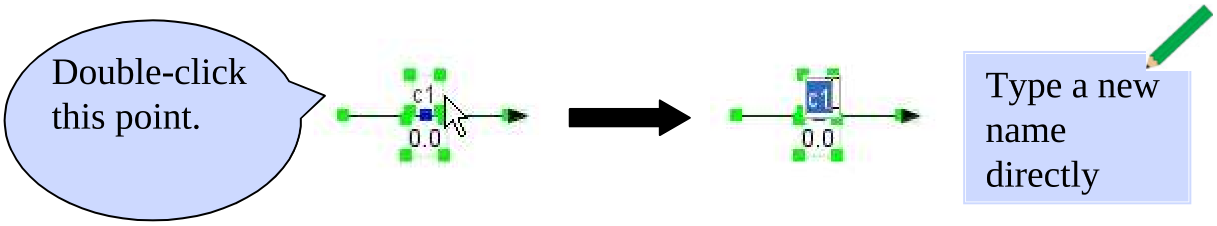


Figure 10

3.5 Graphical Elements

Cell Illustrator provides a library of graphical objects, representing biological entities and processes. Substituting the generic symbols of entities and processes with these elements improves the clarity of biopathway presentations.

3.5.1 Entity

Examples of entities are shown in Figure 11.

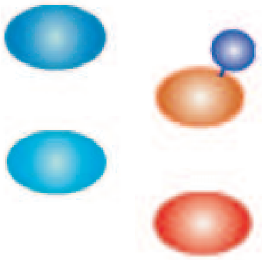
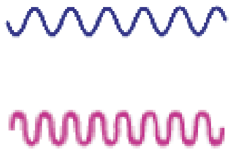
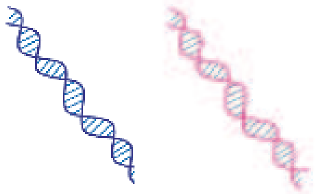
		
Proteins	MRNA	DNA

Figure 11

3.5.2 Process

Examples are shown in Figure 12.

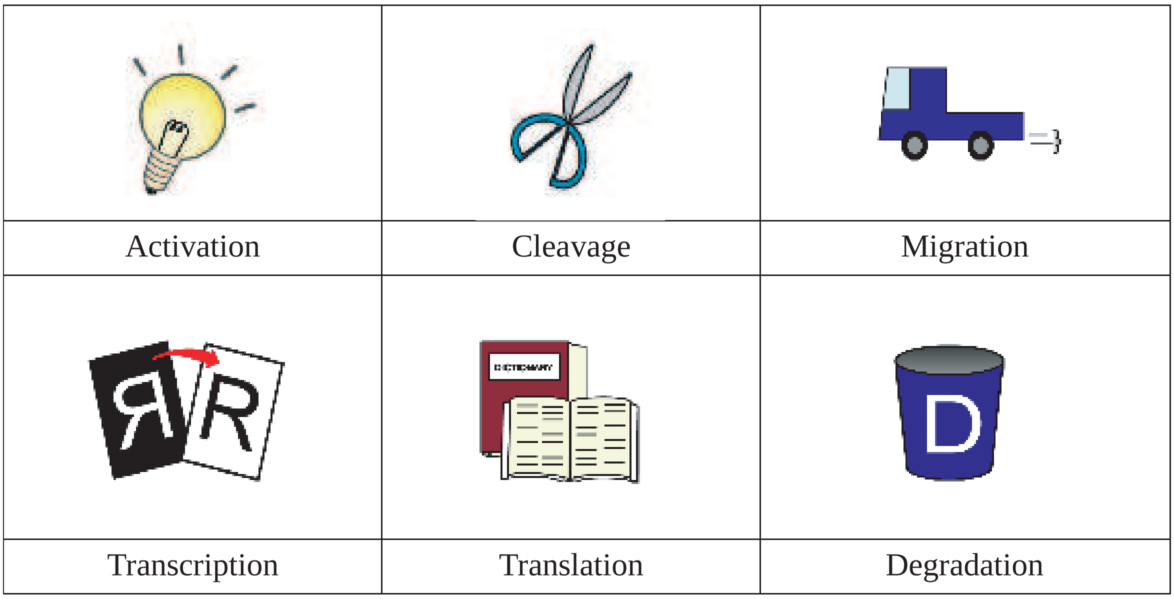


Figure 12

3.6 Element Combinations

There are several rules to follow when combining elements. As described in Section 3.4, it is impossible to directly connect an entity with an entity, or a process with a process. A connector is needed to connect an entity with a process. All combinations of entities and processes are presented in Table 1.

Some important rules:

Rule 1 Inhibitory connector and associate connector can be used in all cases, but process connector cannot connect an entity with a process in certain cases (see rules 2 and 3).

Rule 2 Discrete entities and continuous processes cannot be connected by process connectors.

Rule 3 Generic entities and discrete/continuous processes cannot be connected by input process connectors.

Figure 13 shows an example of a correct connection. On the other hand, if the connection is incorrect, the target element (p1) will not be highlighted, as shown in Figure 14. It is not necessary to remember the rules of combination between elements. Cell Illustrator will detect incorrect connections for you.

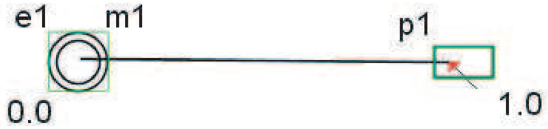


Figure 13

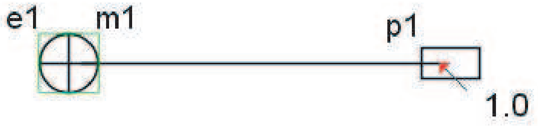


Figure 14

Table 1 Element combinations

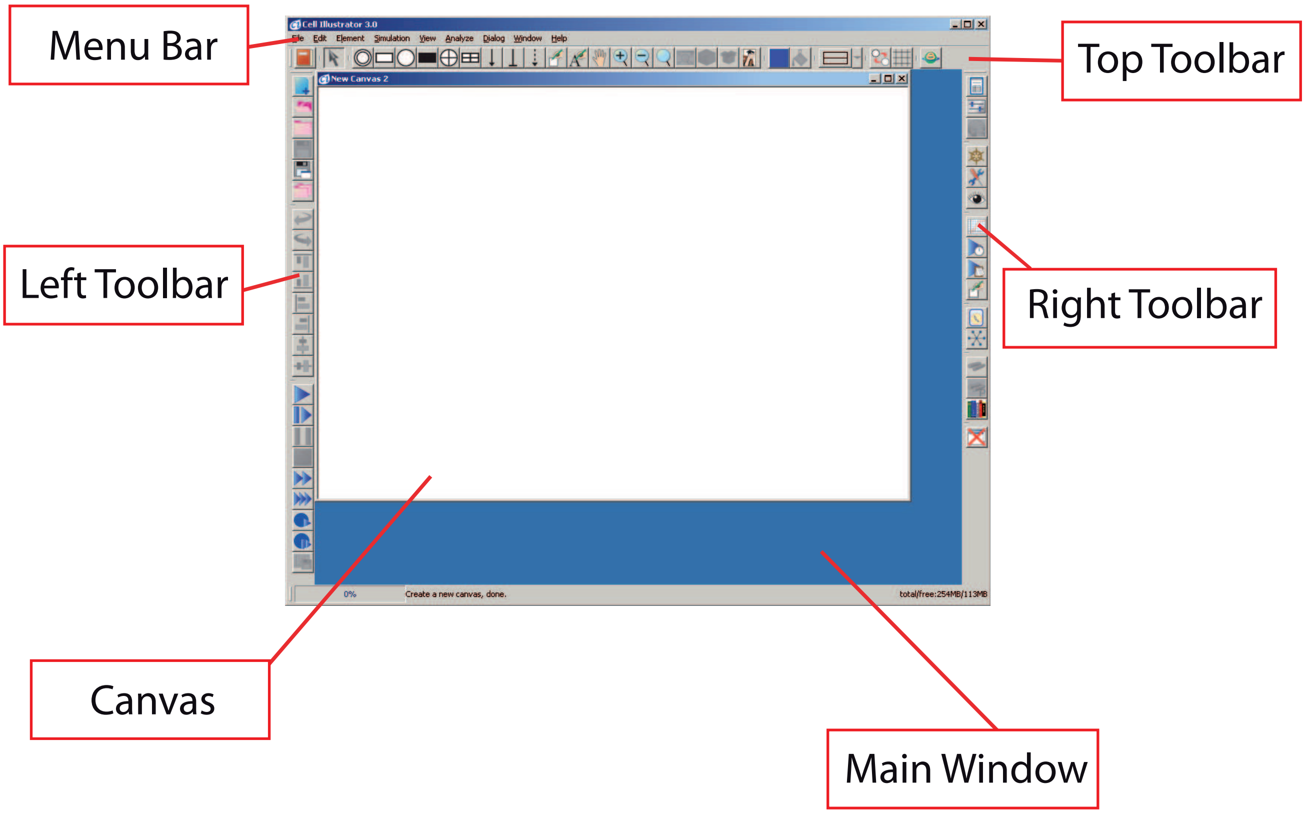
connector type		input connector (process)		
process type		discrete	continuous	generic
entity type	discrete	✓	–	✓
	continuous	✓	✓	✓
	generic	–	–	✓
connector type		input connector (inhibitary/associate)		
process type		discrete	continuous	generic
entity type	discrete	✓	✓	✓
	continuous	✓	✓	✓
	generic	✓	✓	✓
connector type		output connector (process)		
process type		discrete	continuous	generic
entity type	discrete	✓	–	✓
	continuous	✓	✓	✓
	generic	✓	✓	✓

4 INVOKING CELL ILLUSTRATOR AND CREATING A MODEL (BASIC)

Upon starting Cell Illustrator, the program pops up a single main window as shown in Figure 15. The first step in generating a new model is usually creating a new canvas. There are two ways of creating a canvas:

1. Select the first icon ("Create New Canvas") in the **Left Toolbar**.
2. Select the **New** menu option from the **File** menu in the **Menu Bar**.

You can save the created model by selecting **Save/Close** from **FILE** in the **Menu Bar**.



Create a new canvas, done.

Figure 15

4.1 Adding Elements

4.1.1 Adding an Entity/a Process

There are four ways of adding an entity/a process to the canvas:

1. Select one of the icons representing entities and processes from the **Top Toolbar** (for icons, see Figures 2 and 5).
2. Click the right mouse button on the canvas and select **Insert Entity/Insert Process** (Figure 16).
3. Select **Insert Entity/Insert Process** from **Edit** in the **Menu Bar**. The new element will be inserted in the center of the canvas (Figure 17).

4. In the **Menu Bar**, select **Biological Element** from **Dialog**. Drag and drop a new element.

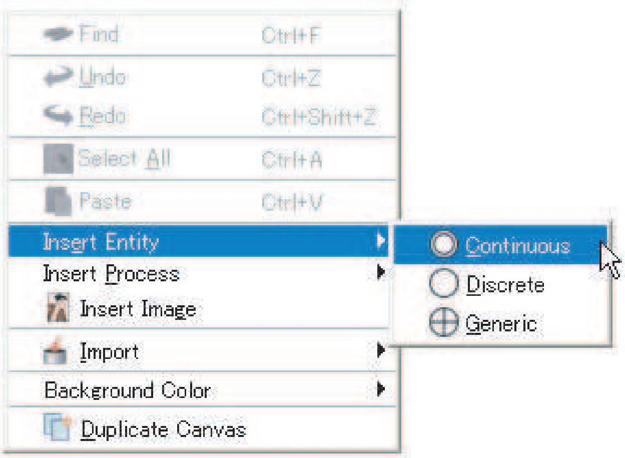


Figure 16

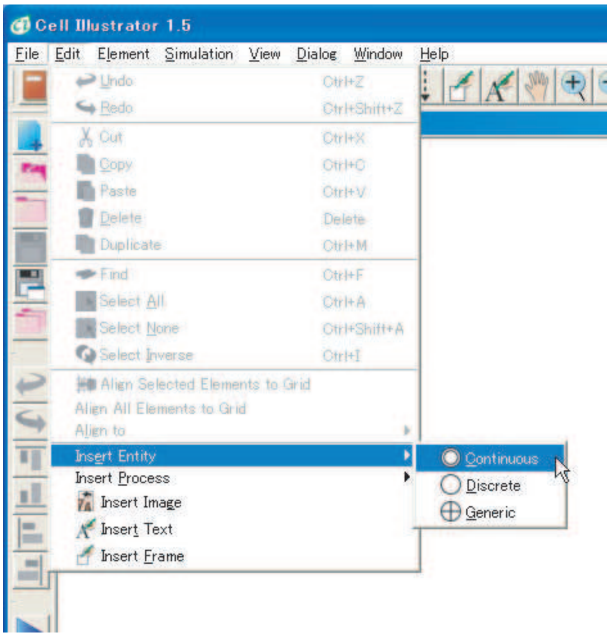


Figure 17

4.1.2 Adding a Connector

In order to connect an entity and a process, do the following steps:

Step 1 From the **Top Toolbar**, select one of the connector icons, as shown in Figure 18 (for connectors, see Figure 9).

Step 2 In the canvas, select an entity you want to connect (Figure 19).

Step 3 Move the mouse cursor to the target process and select it (Figure 20).

There are some limitations in connecting entities and processes. Refer to the rules of combining elements in Table 1.

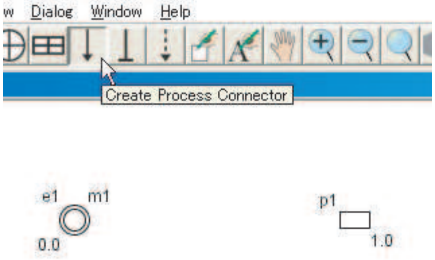


Figure 18

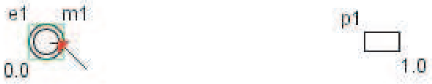


Figure 19

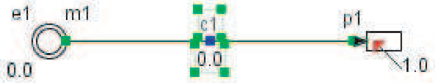


Figure 20

4.2 Modifying a Model

4.2.1 Modifying an Element

You can modify any element in the model. To do this, click on the **Selection Mode** button from the **Top Toolbar**.

4.2.2 Moving an Element

To move an element, click on the element in the canvas and drag it into the new location. You can also drag with the mouse to select multiple elements.

4.2.3 Cutting, Copying, and Pasting an Element

In the **Selection Mode**, after selecting an element, you can cut, copy and paste it in one of several ways:

1. Clicking the right mouse button brings up a popup menu, which allows you to cut, copy, and paste an element.

2. In the **Menu Bar**, select **Cut**, **Copy**, or **Paste** from **Edit**.

3. By using the keyboard, Ctrl+X, Ctrl+C, and Ctrl+V will cut, copy, and paste elements, respectively.

4.2.4 Other Edit Operations

The **Top Toolbar** contains buttons frequently used for other editing operations. The name and the function of each button are summarized in Table 2.

Table 2

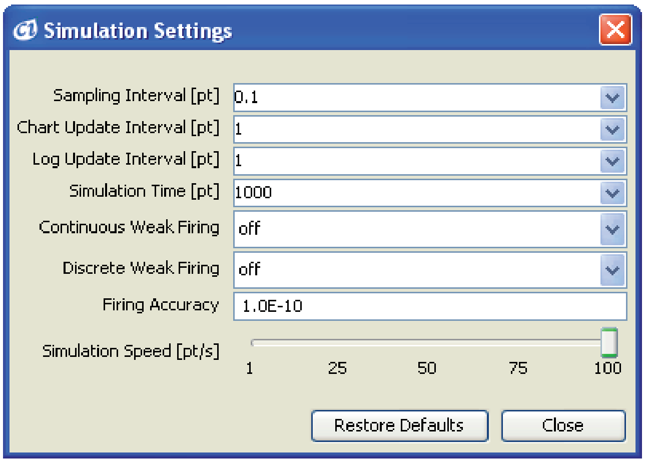
	"Edit Parts" – Organize and use a repository of reusable model parts.
□	"Create Frame" – Create a background frame.
A	"Create Note" – Annotate the model.
✋	"Manual Move" – Move the canvas without using the **Navigator** dialog, or the scroll bars.
	"Zoom In" – Zoom-in on the canvas.
	"Zoom Out" – Zoom-out from the canvas.
	"Reset Zoom" – Restore the default magnification.
	"Group" – Group together chosen elements to treat them as one for editing.
	"Ungroup" – Undo a group operation.
	"Load Image" – Load an image and insert it in the canvas.
□	"Set Color" – Apply a color to selected elements in the canvas as specified in the Select Color Scope.
, , A	"Select Color Tool" – Select the scope of Set Color operation: Fill Color, Line Color, and Text Color, respectively.
▭	"Set Stroke" – Change the stroke pattern of selected elements.
▦	"Toggle Grid Visible Status" – Turn the background grid on or off
	"Toggle Antialiasing Status" – Turn on or off antialiasing
	"Go to BioPACS URL" – Import a metabolic pathway from the public KEGG database.

5 SIMULATION (BASIC)

After creating a model, you can simulate the behavior of the modeled system. This chapter provides a brief explanation of setting-up and running a simulation.

5.1 Setting up a Simulation

You can use the Simulation Settings dialog in the Right Toolbar , which has five options: Sampling Interval, Plot Update Interval, Log Update Interval, Simulation Time, and Simulation Speed (Figure 21). The Simulation Time sets the duration of the simulation process. All the times are specified in Petri net time units (pt). For example, if the Simulation Time is set to 10 [pt], the simulation lasts for 10 [pt] starting from 0 [pt]. The default time is set to 2000 [pt]. In the model shown in Section 7, Simulation Time 100 [pt] is long enough.



Simulation Settings

Sampling Interval [pt] 0.1
Chart Update Interval [pt] 1
Log Update Interval [pt] 1
Simulation Time [pt] 1000
Continuous Weak Firing off
Discrete Weak Firing off
Firing Accuracy 1.0E-10
Simulation Speed [pt/s] 100

Figure 21

5.2 Setting up a Chart

Entities' values can change during simulation. These changes can be visualized in a chart. Although there are several ways of using a chart, we describe the simplest way to do this.

Step 1 Select entities whose values you want to display in a chart. It is possible to select multiple entities for a simulation (Figure 22).

Step 2 Click the right mouse button and select Create Chart . A chart appears, as shown in Figure 23.

However, a chart generated using this method cannot be deleted. The Chart Settings dialog can be used for editing and deleting the chart.

5.3 Running a Simulation

Once the simulation and the corresponding chart are set up, you can start and control the simulation. There are several buttons in the Left Toolbar , controlling the simulation. Their functions are summarized in Table 3.

Cell Illustrator executes a simulation at speed 1 [pt] per 1 second in real time. The maximum speed depends on the executing machine.

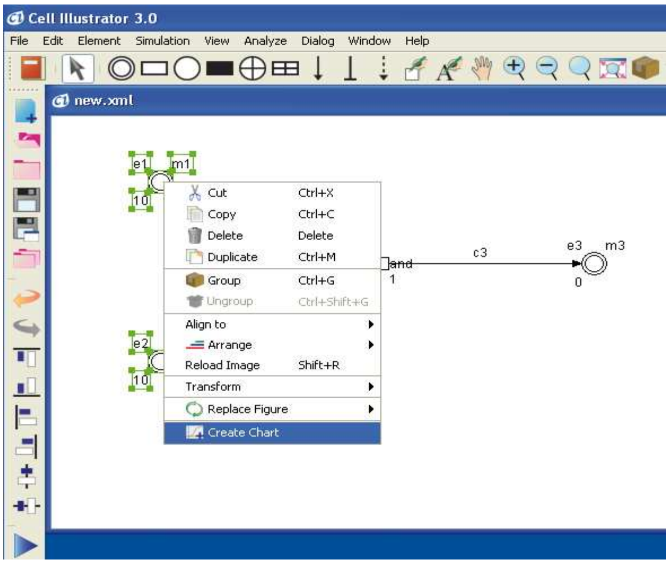


Figure 22

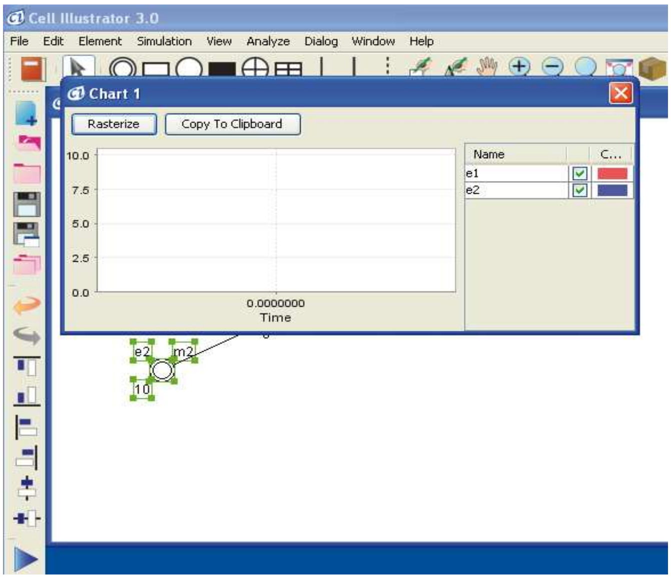


Figure 23

Table 3

▶	Start simulation.
●	Invoke a simulation with animated tokens moving between entities.
▶▶	Play at high speed.
▶▶▶	Play at maximum speed.
▮▶	Play one step.
●	Play one step with animated tokens.

6 SIMULATION RULES (BASIC)

It is easy to prepare a simulation by connecting entities and processes, as described in Sections 3 and 4. In this section, we explain how a simple model can become a more complicated one by adding properties of elements. After reading this section you will be able to easily create the model described in Section 7.

6.1 Model with Discrete Entities and Discrete Processes

First, generate the model shown in Figure 24:

Step 1 Place four discrete entities on the canvas.

Step 2 Place one discrete process on the canvas.

Step 3 Link each of the two entities on the left to the discrete process using process connectors.

Step 4 Connect the discrete process and each of the two remaining entities with connectors.

Step 5 Select all four entities and create a chart for tracking the changes of e1-e4 values.

The values in the entities are transferred from the sources of input connectors (i.e. entities e1 and e2) to the targets of output connectors (i.e. entities e3 and e4). In the model in Figure 24, all entities have default values of zero. Therefore, these values do not change in the course of simulation as shown in Figure 25.

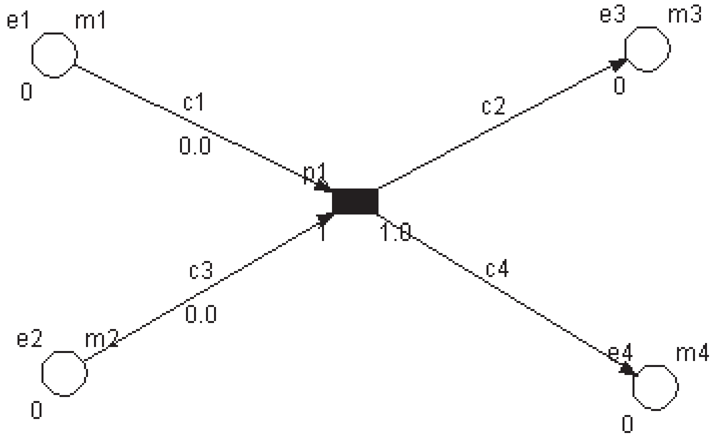
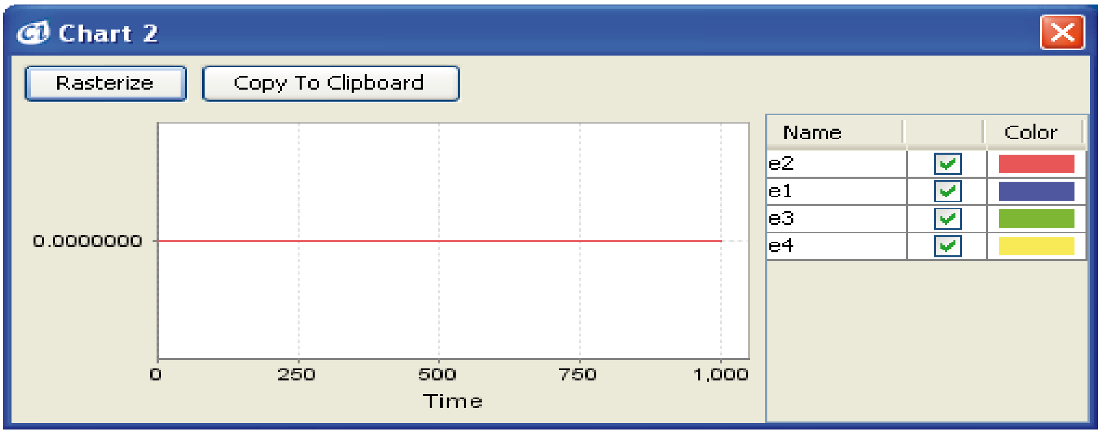


Figure 24

6.1.1 Initial Value

If the initial values of e1 and e2 are set to 10 (Figure 26), the simulation produces a chart shown in Figure 27. After the values of e1 and e2 become zero, the progress is stopped, and then the values of the entities do not change any more after becoming zero.



Name		Color
e2	☑	🟥
e1	☑	🟦
e3	☑	🟩
e4	☑	🟨

Figure 25

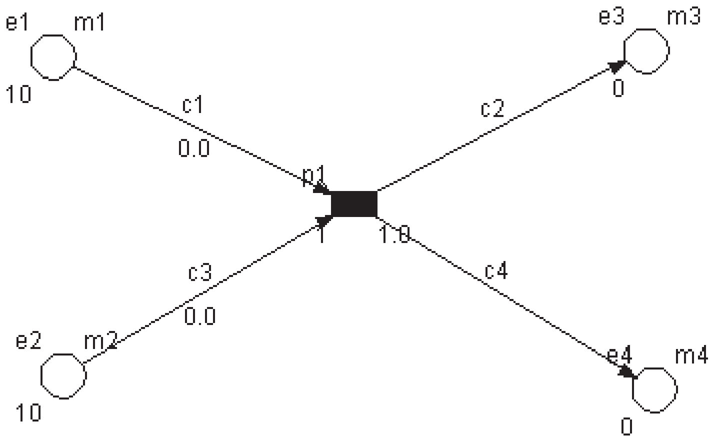
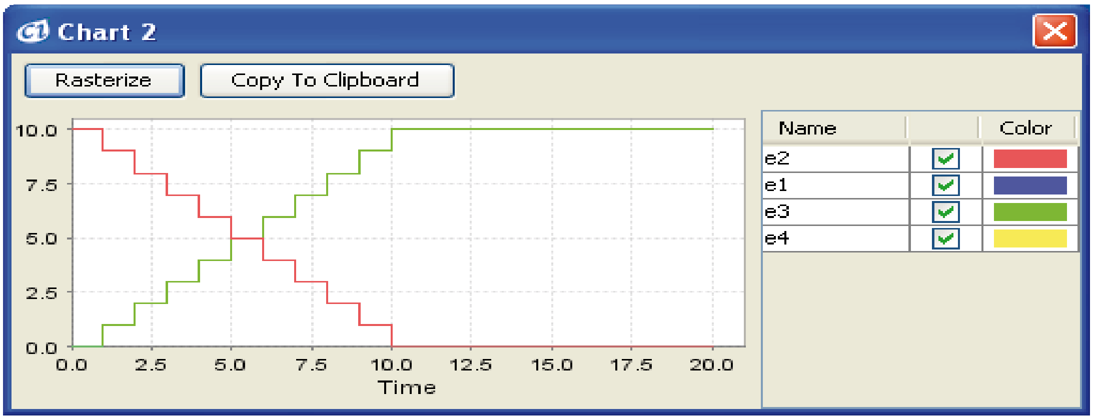


Figure 26

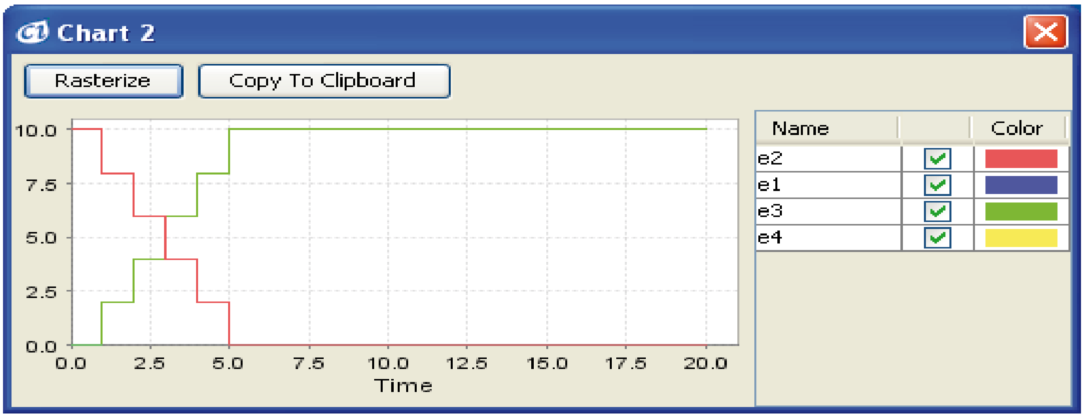


Name		Color
e2	☑	🟥
e1	☑	🟦
e3	☑	🟩
e4	☑	🟨

Figure 27

6.1.2 Speed

In Figure 27, at the 10 [pt] simulation time, the values of e1 and e2 become zero. If you change the speed parameter of process p1 from 1 to 2, the values of e1 and e2 will become zero at the 5 [pt], time as shown in Figure 28. The speed parameter controls the speed of the process.

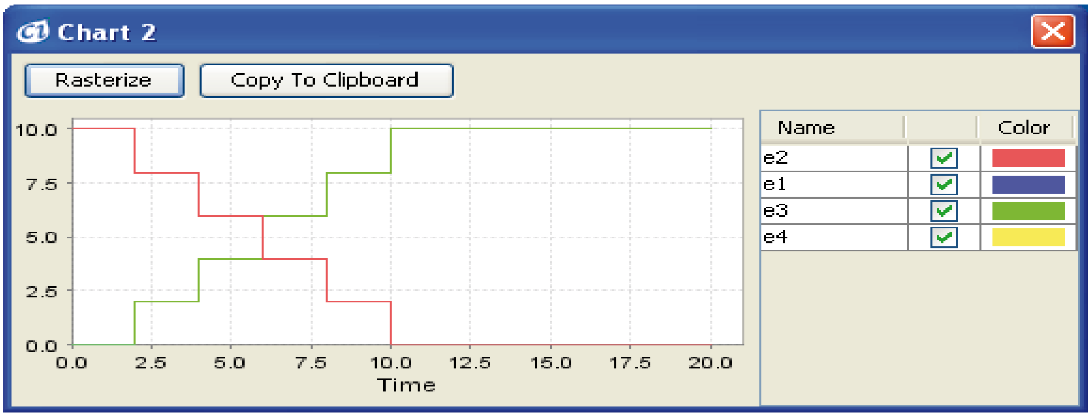


Name		Color
e2	☑	🟥
e1	☑	🟦
e3	☑	🟩
e4	☑	🟨

Figure 28

6.1.3 Delay

Let's change the delay parameter of the process p1 from 1 to 2. The delay parameter of the process controls the time during which this process is suspended. The simulation produces results shown in Figure 29. At the time of 10 [pt], the values of e1 and e2 become the same as the model in Figure 26. We see that the values are changed every 1 [pt] in Figure 27, while those in Figure 29 are changed at every 2 [pt]. The delay parameter is associated with the speed parameter and controls the interval between the changes of values controlled by the particular process.

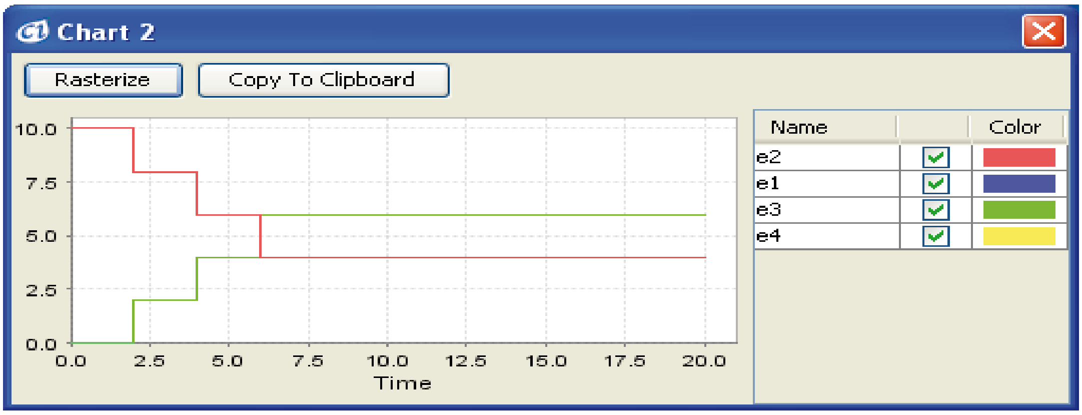


Name		Color
e2	☑	🟥
e1	☑	🟦
e3	☑	🟩
e4	☑	🟨

Figure 29

6.1.4 Threshold

Now, let's change the threshold of connector c1 to 5 and run the simulation. The resulting chart is shown in Figure 30. The values of e1 and e2 are not changed after the time of 6 [pt]. The reason is that if the value of an input entity is smaller than the threshold of the connector, the value of the entity cannot be transmitted to the target process. In order to activate a process, all connectors to the process must satisfy the above condition. In this example, at the time 6 [pt], the threshold of c3 satisfies the condition, because it is set to zero. However, c1 does not satisfy the condition because its threshold is set to 5, and becomes larger than the values of e1 and e2 in the course of the simulation. Therefore the process is stopped.



Name		Color
e2	☑	🟥
e1	☑	🟦
e3	☑	🟩
e4	☑	🟨

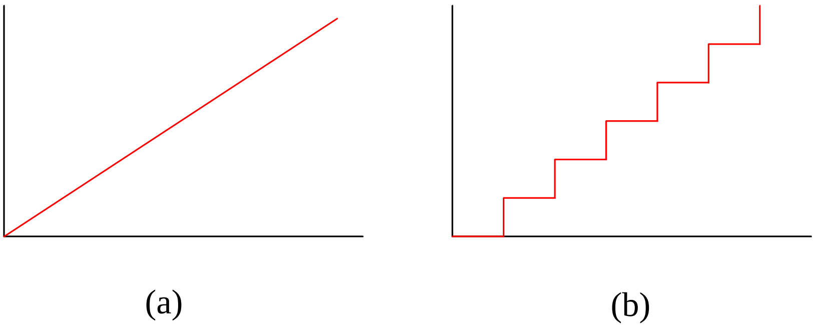
Figure 30

6.2 The Concepts of Discrete and Continuous

There are two types of objects in Cell Illustrator: discrete and continuous ones. Let us explain these concepts using the example of turning on a faucet. When you turn on the faucet (discrete event), water starts running, and its amount is represented by a continuous value.

Discrete objects values are represented by integer numbers (0, 1, 2, etc.) Cell Illustrator most often uses discrete objects to represent on/off actions, or phenomena which can be described without assuming continuity of their numerical representation. Conversely, continuous objects and processes are represented using real numbers. In CI, continuous values are usually associated with concentrations, process rates, time, or similar objects of continuous nature.

The plots in Figure 31 present the difference caused by using continuous vs. discrete objects in a simulation: continuous changes of some values produce smooth lines and curves representing the values (a), while discrete ones, represented by integers, yield jagged plots (b).


(a)

(b)

Figure 31

6.3 Model with Continuous Entities and Continuous Processes

Let's generate the model in Figure 32 by the following steps:

Step 1 Place four continuous entities on the canvas.

Step 2 Place one continuous process on the canvas.

Step 3 Link each of the two continuous entities on the left to the continuous process, using process connectors.

Step 4 Connect the continuous process and each of the two remaining continuous entities using process connectors.

Step 5 Select all four entities and create a chart.

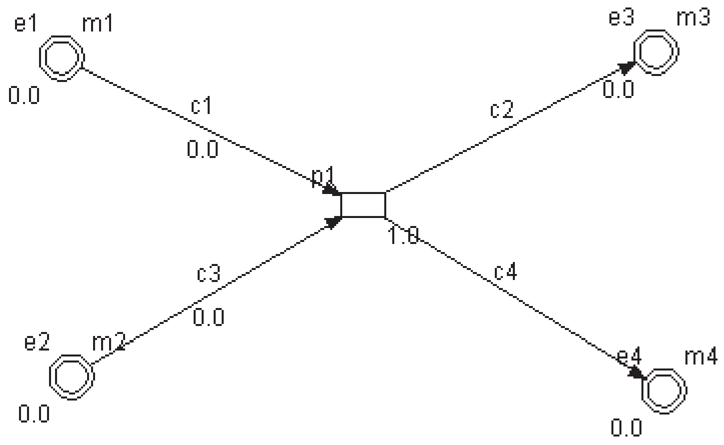


Figure 32

6.3.1 Initial Value and Threshold

The initial value of a continuous entity is different than that of a discrete one. A continuous entity's value can be a real number, like 1.5, 2.5, $\cdots$ while a discrete entity holds only integer numbers (0, 1, 2, $\cdots$). However, the threshold value of a discrete entity can hold real numbers too. The function of the threshold value in a continuous entity is the same as a discrete entity's threshold (Section 6.1.4).

6.3.2 Speed and Delay

Continuous processes have only the speed parameter, while discrete processes have two parameters: the speed and the delay. Let's execute the model presented in Figure 33, and compare the chart with the one in Figure 27.

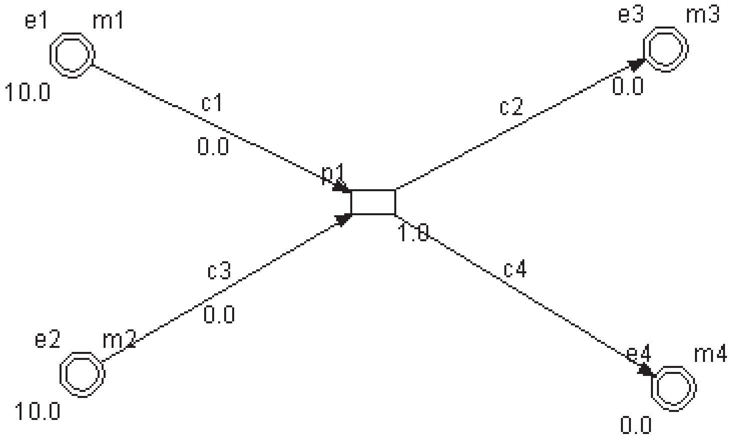
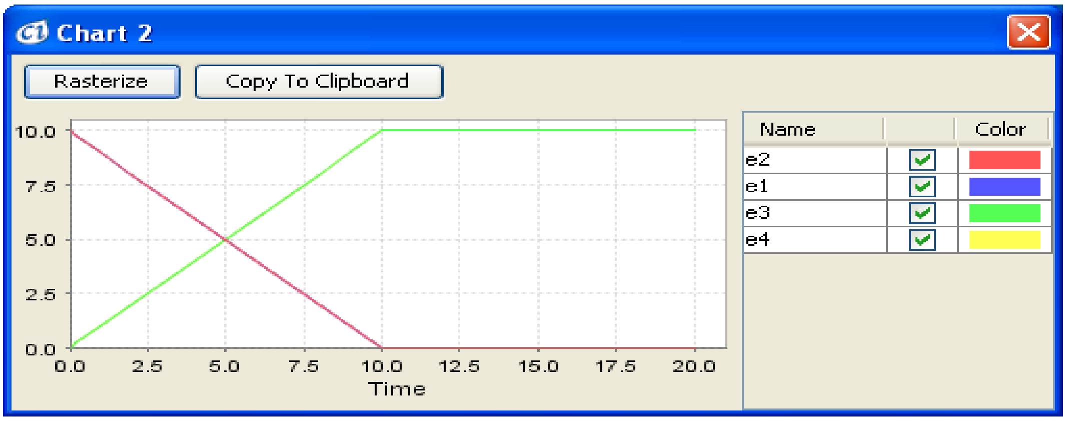


Figure 33



Name		Color
e2	☑	🟥
e1	☑	🟦
e3	☑	🟩
e4	☑	🟨

Figure 34

7 MODELING BASIC BIOCHEMICAL PROCESSES WITH CI

This section explains how to model biopathways using CI.

7.1 Degradation

In a cell, almost all mRNAs and proteins undergo fragmentation at certain speeds. This process is called degradation. Let's create a model of the degradation process:

Step 1 Create an entity, change its name to p53, and assign it the initial value of 100.

Step 2 Create a process and assign it the name "degradation".

Step 3 Connect p53 to the degradation process with a process connector.

Step 4 Set the degradation speed using the speed parameter of the process (the default speed is set to 1.0).

There are three possible combinations of the pair (entity, process): (discrete, discrete), (continuous, continuous), and (continuous, discrete), shown in Figure 35. The results of the simulation are presented in Figure 36.

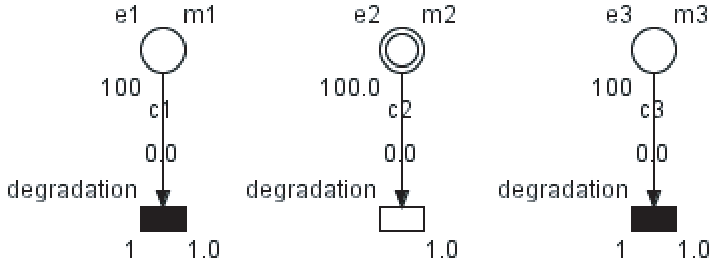
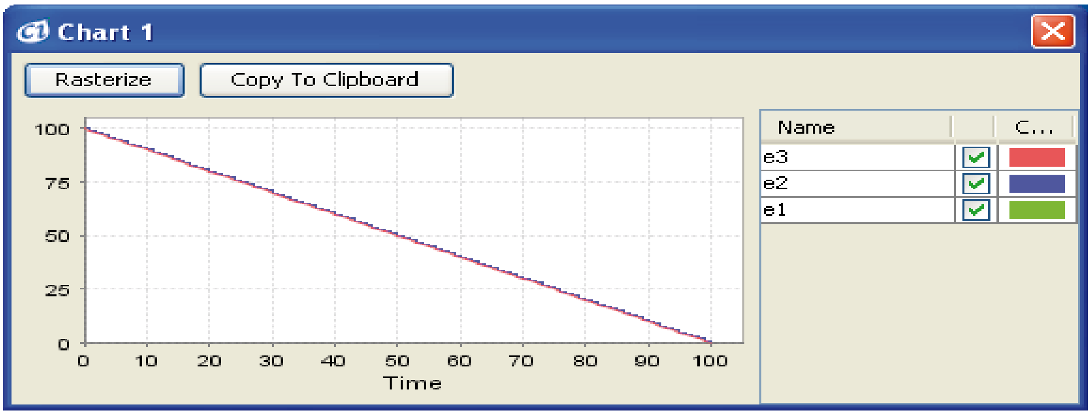


Figure 35



Name		C...
e3	☑	🟥
e2	☑	🟦
e1	☑	🟩

Figure 36

In this newly created model, the degradation speed of p53 does not depend on the value of p53. In real systems, reaction speeds are usually concentration-dependent. To make the degradation speed dependent on the concentration of the substrate - p53, the speed parameter should be changed as shown in Figure 37. The results of the simulation are presented in Figure 38.

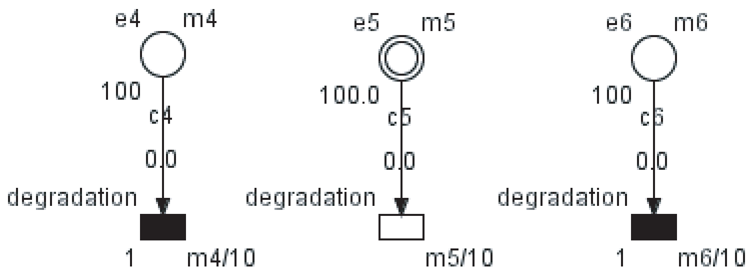
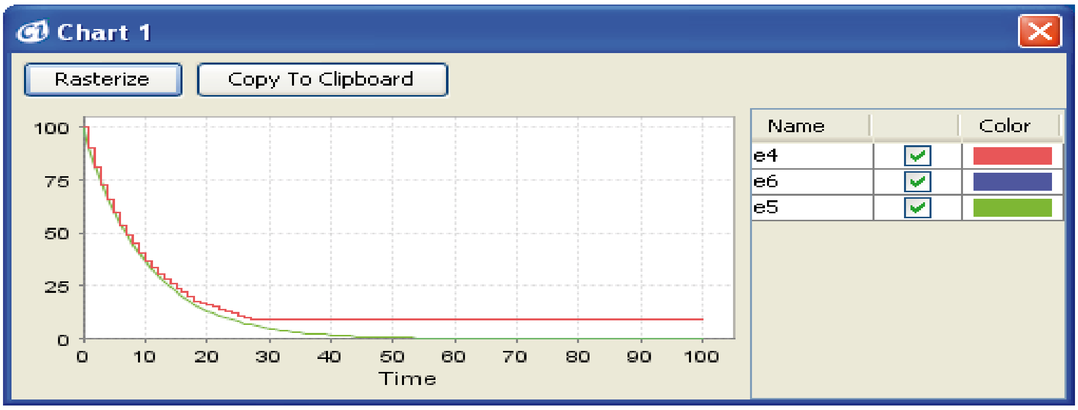


Figure 37



Name		Color
e4	☑	🟥
e6	☑	🟦
e5	☑	🟩

Figure 38

7.2 Translocation

In a cell, mRNAs and proteins move between organelles e.g. from the cytoplasm to the nucleus, and from the nucleus to the Golgi apparatus. A model of translocation can be created:

Step 1 Create two entities, assign them names (e.g. p53_nuclei and p53_cytoplasm) and values (e.g. 100 and 0), respectively.

Step 2 Create a process and assign it a name (e.g. translocation). Connect p53_nuclei to the process with an input connector, and the process to p53_cytoplasm with an output connector.

Step 3 Set the translocation speed using the speed parameter of the process (the initial speed is 1.0).

Possible combinations of the elements (p53_nuclei, translocation, p53_cytoplasm) are: (discrete, discrete, discrete), (discrete, discrete, continuous), (continuous, discrete, discrete), (continuous, discrete, continuous), and (continuous, continuous, continuous), as shown in Figure 39. The last one was chosen to be used in the model. The results of the simulation are presented in Figure 40.

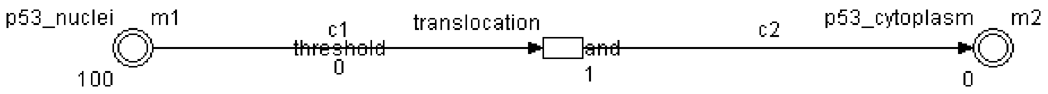
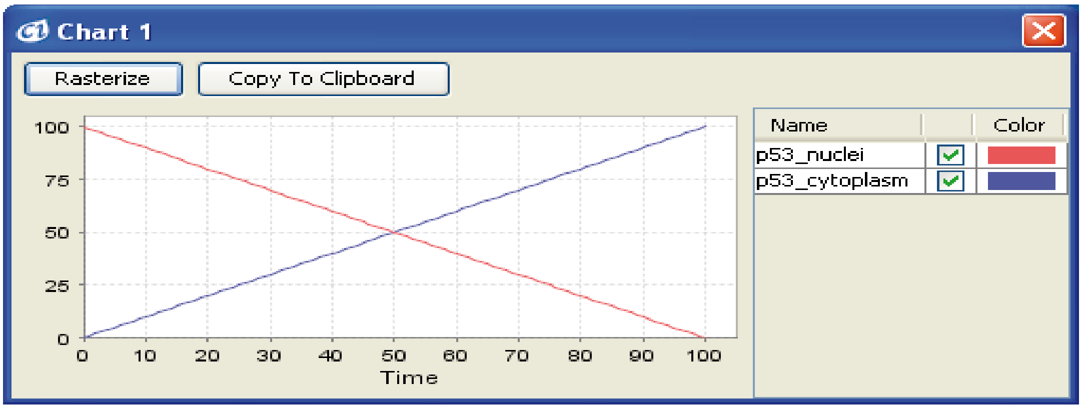


Figure 39



Name		Color
p53_nuclei	☑	🟥
p53_cytoplasm	☑	🟦

Figure 40

To make the translocation speed dependent on the value of the p53_nuclei entity (say, m1), as in the degradation example (Section 7.1), the process speed should be a function of m1 (e.g. m1/10) (Figure 41). The behaviour of this more realistic model is shown in Figure 42.

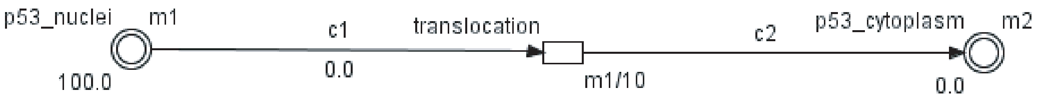
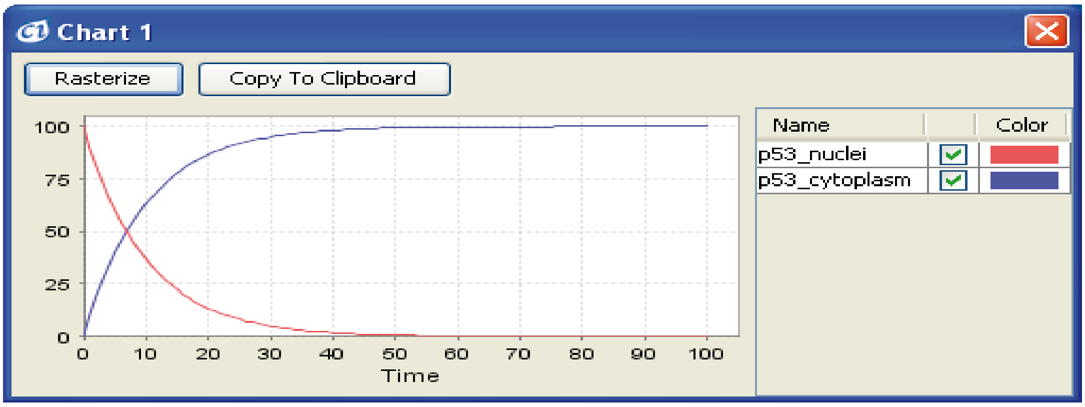


Figure 41



Name		Color
p53_nuclei	☑	🟥
p53_cytoplasm	☑	🟦

Figure 42

7.3 Transcription

Step 1 Create a process and assign it a name (e.g. transcription).

Step 2 Create an entity and assign it a name (e.g. mRNA_p53).

Step 3 Connect the transcription process to the mRNA_p53 entity with a process connector.

Step 4 Set the transcription speed using the speed parameter of the transcription process (the default speed is set to 1.0).

In **Step 1** and **Step 2**, the possible combinations of the pair (transcription, mRNA_p53) are (discrete, discrete), (discrete, continuous), and (continuous, continuous), as shown in Figure 43. The results of the simulation are presented in Figure 44.

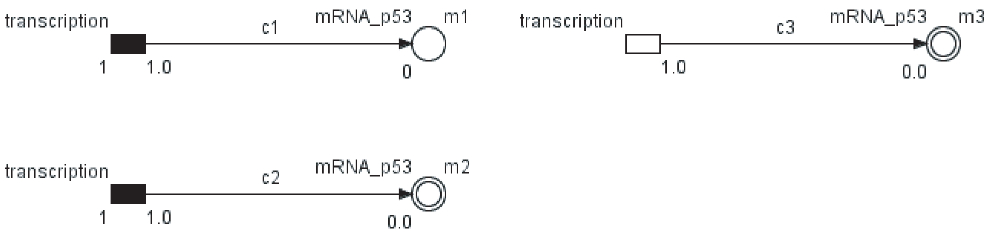
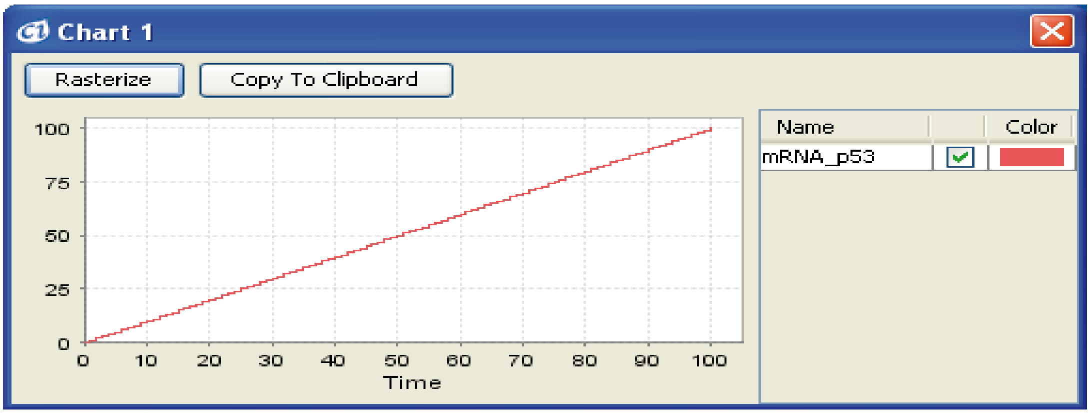


Figure 43



Name		Color
mRNA_p53	☑	🟥

Figure 44

In general, the mRNA synthesis and degradation occur simultaneously. The models presented in Figure 45 combine the transcription process with the degradation process. The results of the simulation are presented in Figure 46.

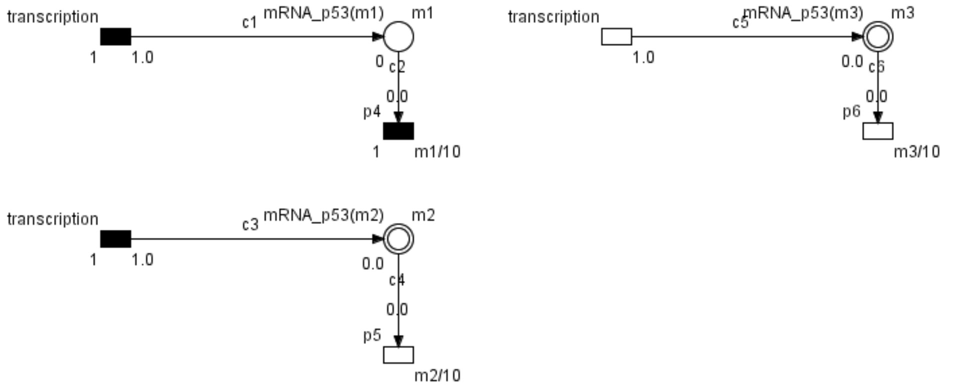
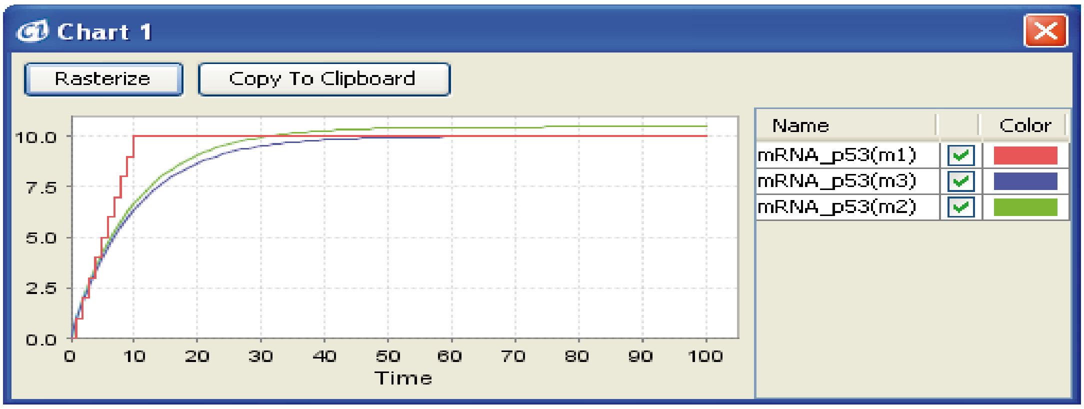


Figure 45



Name		Color
mRNA_p53(m1)	☑	🟥
mRNA_p53(m3)	☑	🟦
mRNA_p53(m2)	☑	🟩

Figure 46

7.4 Binding

In a cell, two proteins can bind together to create a complex. A model for binding can be created:

Step 1 Create three continuous entities and assign them names (e.g. p53, mdm2, p53_mdm2) and values (e.g. 100, 50, 0 respectively).

Step 2 Create one continuous process and assign it a name (e.g. binding).

Step 3 Connect two entities, p53 and mdm2, to the binding process, and the binding process to p53_mdm2.

Step 4 Set the binding speed using the speed parameter of the binding process (the binding speed defaults to 1.0).

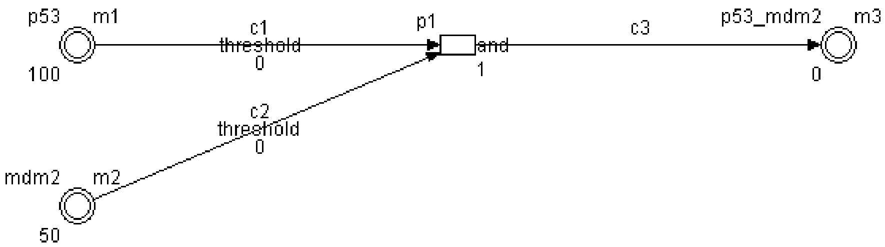
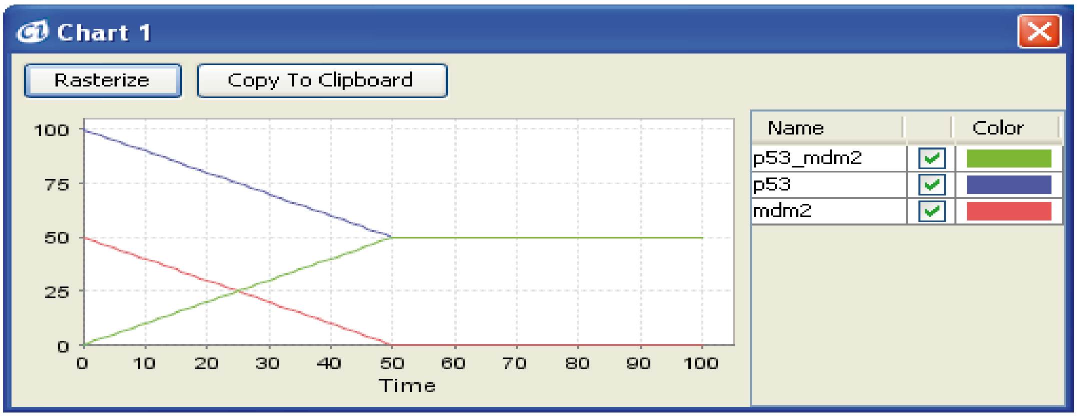


Figure 47

Test the created model - the values of the entities should change as shown in Figure 48



Name		Color
p53_mdm2	☑	🟩
p53	☑	🟦
mdm2	☑	🟥

Figure 48

The kinetics of binding depends on the concentrations of the proteins, so the model needs to be modified accordingly. Set the speed of process to m1*m2/400. Note that this formula is the well-known kinetic equation: v=[m1]*[m2]*k, where: 'v' is the process speed, '[m1]' and '[m2]' are the proteins' concentrations, and 'k' is the kinetic constant. Such a system behaves in the way shown in Figure 48.

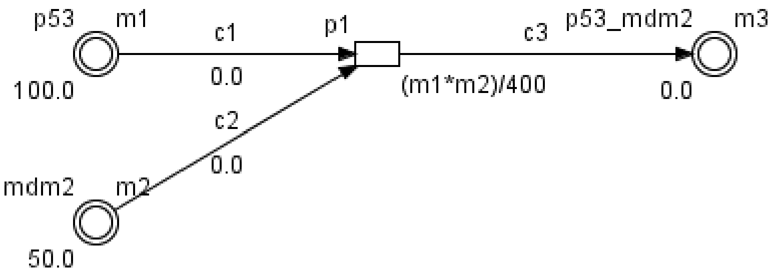
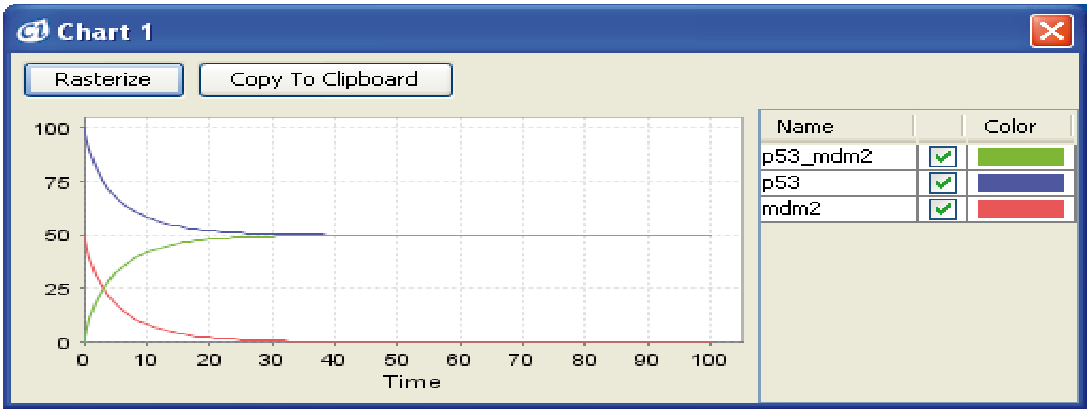


Figure 49



Name		Color
p53_mdm2	☑	🟩
p53	☑	🟦
mdm2	☑	🟥

Figure 50

7.5 Dissociation

Dissociation is a process reverse to binding (7.4). Create a model:

Step 1 Create three continuous entities and assign them names (e.g. p53, mdm2, and p53_mdm2) and values (e.g. 50, 0, and 50 respectively).

Step 2 Create one process and assign it a name (e.g. dissociation).

Step 3 Connect p53_mdm2 to the dissociation process, and the dissociation process to p53 and mdm2.

Step 4 Set the dissociation speed using the speed parameter of the process (the defaults is 1.0).

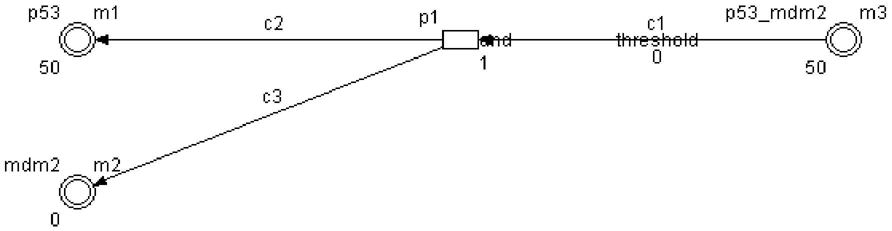
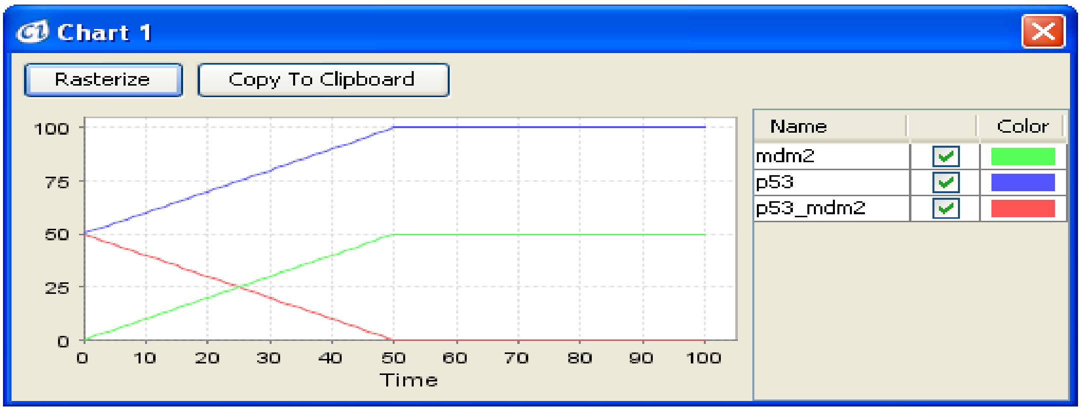


Figure 51

Test the created model - Figure 52 shows its expected behavior.



Name		Color
mdm2	☑	🟩
p53	☑	🟦
p53_mdm2	☑	🟥

Figure 52

In the real system the dissociation speed is described by the formula: v=k*[m3], where: 'v' is the speed '[m3]' is the concentration of 'm3' and 'k' is a kinetic constant.

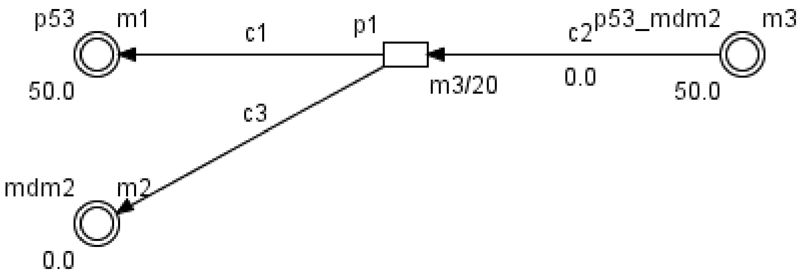
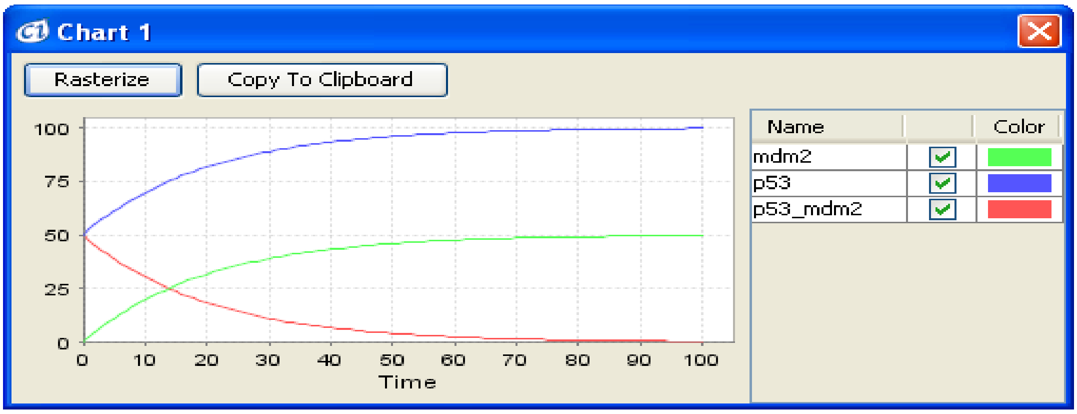


Figure 53



Name		Color
mdm2	☑	🟩
p53	☑	🟦
p53_mdm2	☑	🟥

Figure 54

7.6 Inhibition

So far we have described models using process connectors only. In this example an inhibitory connector will be explained.

A specific drug sometimes inhibits the activation of the transcription. The transcription model from Section 7.3 can be modified to include inhibition:

Step 1 Create an entity and assign it a name (e.g. doxorubicine).

Step 2 Add one more process (p2) and connect it to the doxorubicine entity with a process connector.

Step 3 Connect the doxorubicine entity to the transcription process with an inhibitory connector (The threshold of the inhibitory connector is set to zero).

Several combinations of element type (process p1, doxorubicine) are possible in **Step**s **1** and **3**, i.e. (discrete, discrete), (discrete, continuous), and (continuous, continuous). Figure 55 shows only two of the combinations: (continuous, continuous) and (discrete, discrete).

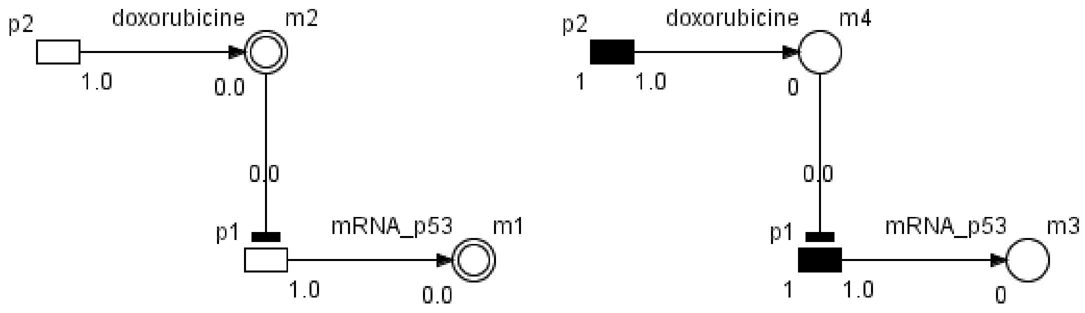
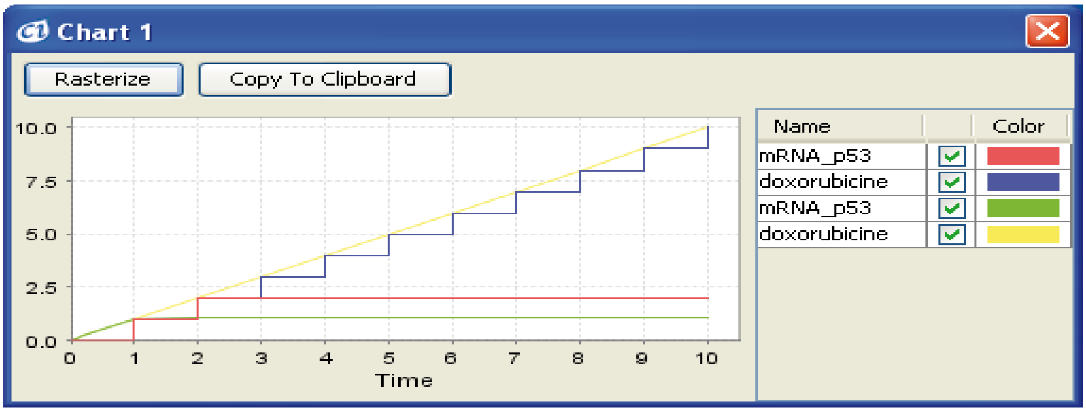


Figure 55



Name		Color
mRNA_p53	☑	🟥
doxorubicine	☑	🟦
mRNA_p53	☑	🟩
doxorubicine	☑	🟨

Figure 56

An inhibitor connector represses the response of the connected process. Whether the process is inhibited by the inhibitory connector depends on the relationship between the threshold value of the connector, and the value of the entity. If the value of the entity is larger than the threshold value of the connector, the repression is enabled.

In this model (Figure 55), the transcription progresses at the starting point, as the initial value of doxorubicin is set to zero. Then, the value of doxorubicin becomes larger than zero, and the transcription process is stopped immediately. The results of the simulation are presented in Figure 56.

The strength of a drug is represented by the threshold value of the inhibitory connector. Let's set the threshold value of the inhibitory connector to 5 (Figure 57). The drug will be effective if the value of the entity representing the drug is above 5. If the value of the drug is under 5, the effect of the drug will be weak, as shown in Figure 58.

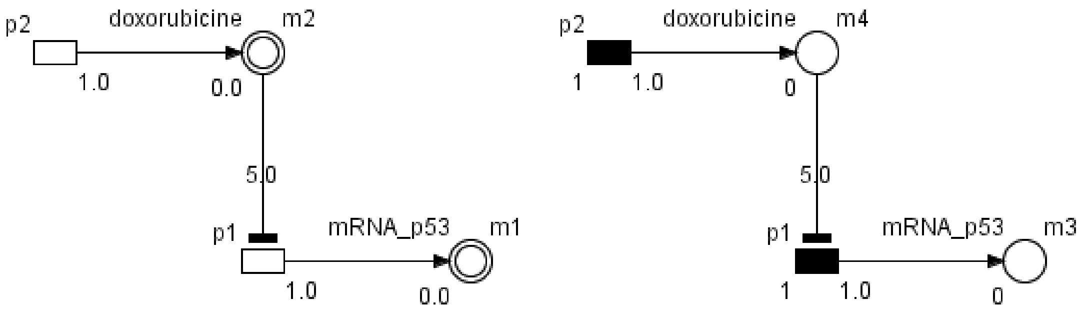
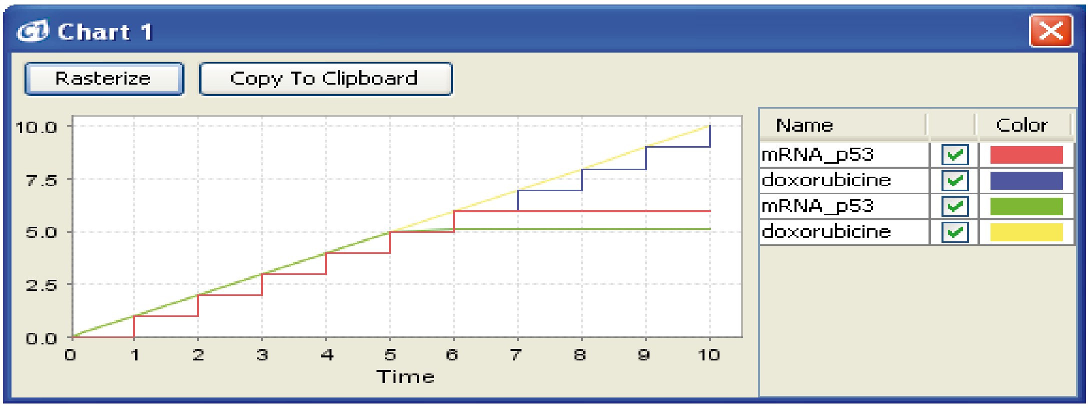


Figure 57



Name		Color
mRNA_p53	✓	🟥
doxorubicine	✓	🟦
mRNA_p53	✓	🟩
doxorubicine	✓	🟨

Figure 58

7.7 Phosphorylation Mediated by Enzymatic Reactions - Association Connectors

Enzymatic reactions are important events in living organisms. An enzyme promotes or influences chemical reactions without itself being destroyed or changed in any way. A model of an enzymatic reaction can be created easily by using association connectors. To create a model, in which an enzymatic reaction speeds up phosphorylation, execute the following steps:

Step 1 Create two entities and assign them names (e.g. p53 and p53-P) and values (e.g. 100 and 0), respectively.

Step 2 Create a process and set its name (e.g. phosphorylation). Connect p53 to the process, and the process to p53-P.

Step 3 Set the phosphorylation speed using the speed parameter of the phosphorylation process (the phosphorylation speed defaults to 1.0).

Step 4 Create one entity and set its name (e.g. CAK).

Step 5 Create a process and connect it to CAK with a process connector.

Step 6 Connect CAK to the phosphorylation process with an association connector.

Step 7 Change the threshold of the association connector to 1 (the default value is 0).

Then, combinations of a process and an entity (CAK) added in **Step 1** and **Step 3** are (discrete, discrete), (discrete, continuous), and (continuous, continuous). Figure 59 shows one model of combinations, i.e, (continuous, continuous).

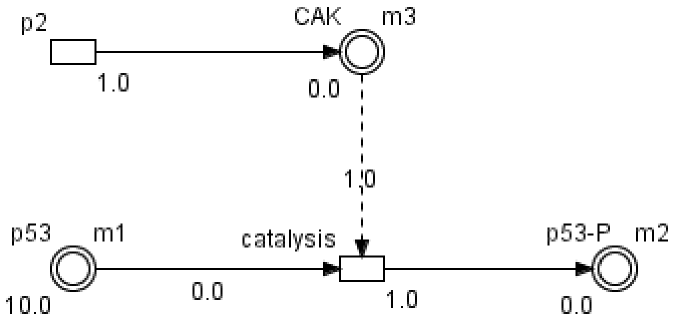
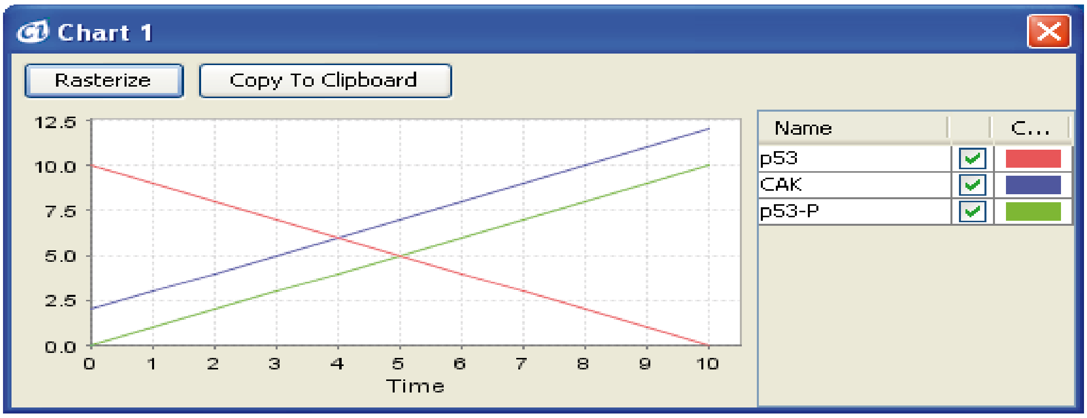


Figure 59

The association connector can promote the execution of a process. Whether the connected process will be promoted depends on the comparison between the initial value of the input entity and the threshold value of the association connector. If the initial value of the entity is larger than the threshold value of the connector, the target process is enabled. The association connector, unlike a process connector, does not change the value of its input entity. The association connectors are usually used to express enzymatic reactions within the body. The results of the simulation are presented in Figure 60.



Name		C...
p53	☑	🟥
CAK	☑	🟦
p53-P	☑	🟩

Figure 60

The model in Figure 59 shows that the initial value of the entity (CAK) is set to zero. Because it is smaller than the threshold of the association connector, the phosphorylation process does not proceed at the starting point. After the simulation is executed, however, the phosphorylation process is repressed as shown in Figure 60, because the value of the entity (CAK) becomes larger than 1.

The strength of an enzyme can be controlled by setting its threshold value. For example, if the threshold value is changed from 1 to 5 (Figure 61), the effect of the enzyme will be apparent when the value becomes larger than 5. Otherwise, the effect of the enzyme will be weak (Figure 62).

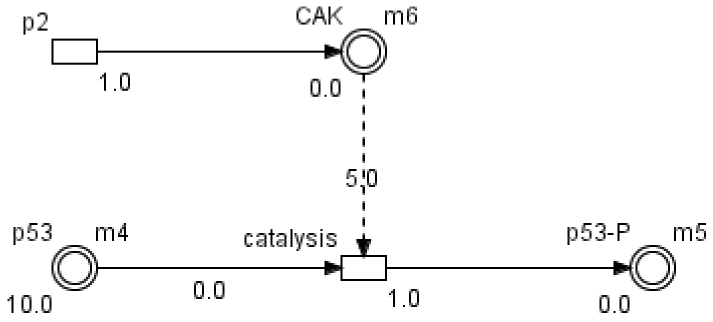
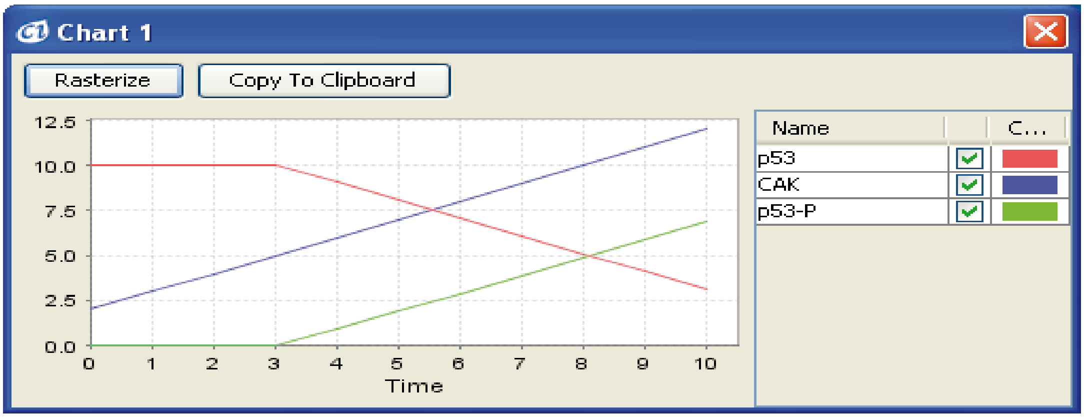


Figure 61



Name		C...
p53	☑	🟥
CAK	☑	🟦
p53-P	☑	🟩

Figure 62

In most systems, the reaction rate depends on enzyme (m3) and substrate (m1) concentrations. In the example in Figure 63, the speed of the phosphorylation process is defined as the equation (m1*m3)/10 (10 is an example). Figure 63 shows a model composed of all continuous elements. The results of the simulation are presented in Figure 64.

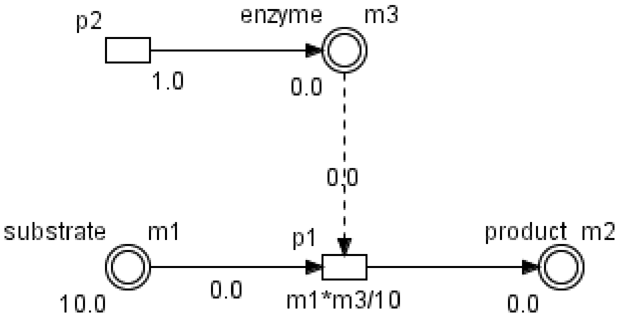
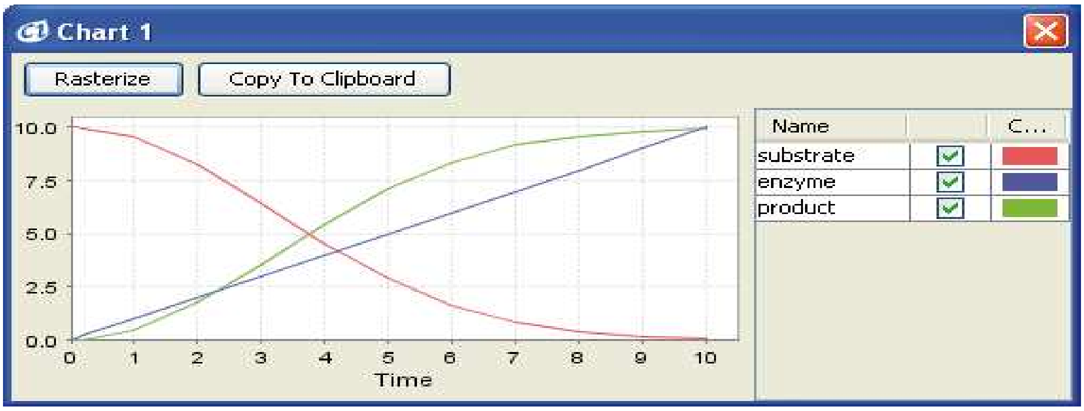


Figure 63



Name		C...
substrate	☑	🟥
enzyme	☑	🟦
product	☑	🟩

Figure 64

7.8 Enzymatic Catalysis - Michaelis-Menten Kinetics

This section is devoted to creating a model of enzymatic catalysis according to Michaelis-Menten theory.

Entities

There are four constituent elements in a simple enzymatic catalysis system: the **substrate**, the **enzyme**, the **enzyme-substrate complex**, and the **product**. Create four continuous entities representing these elements, and assign them descriptive names Figure 65

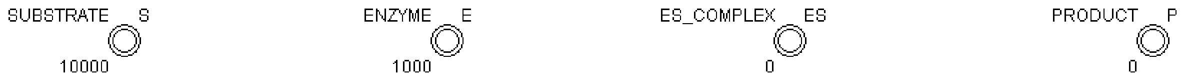


Figure 65

Processes

The first step of the catalysis is the formation of the enzyme-substrate complex. Place a continuous process on the canvas, connect it to the entities with process connectors, as shown in Figure 66. Name the process appropriately ('ES_COMPLEX_CREATION'). Assign to the SUBSTRATE and ENZYME entities the values of 10,000 and 1,000, respectively. Assign to the process the speed given by the formula S*E*2.5E-6. It represents the kinetic equation: v=k[S][E], where the rate constant k=2.5*10-6, [S] and [E] are the concentrations of the substrate and the enzyme, respectively.

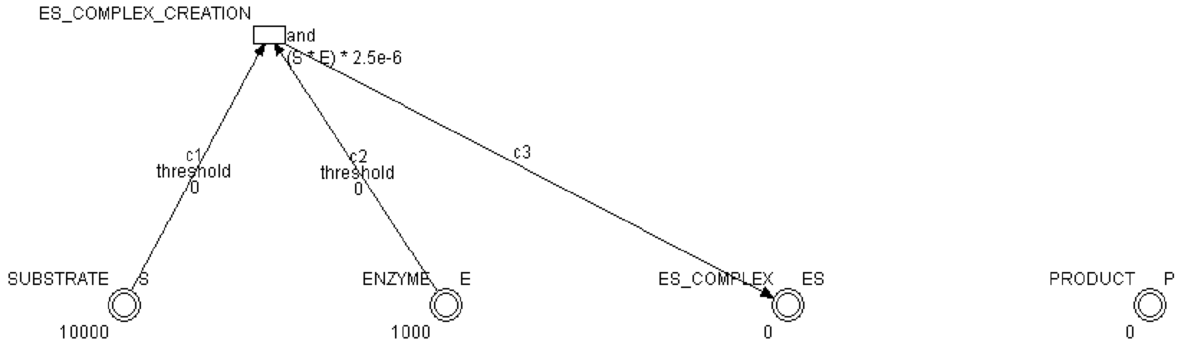


Figure 66

Once the substrate-enzyme complexes are formed, they start breaking down back to their constituent elements. Add the process representing this breakdown to the model. Assign it the speed of ES*1E-2 Figure 67.

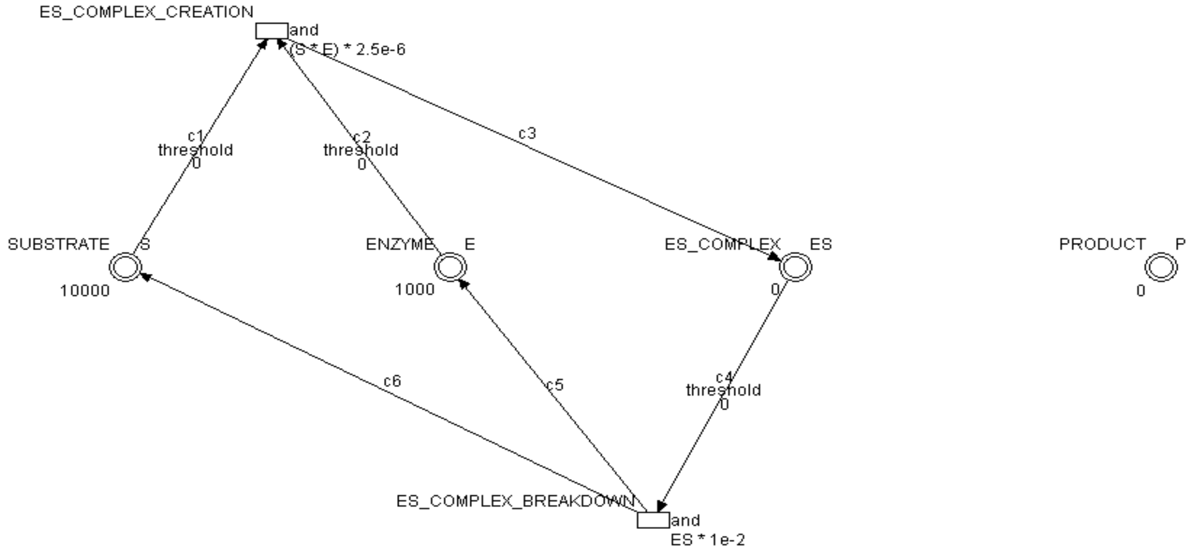


Figure 67

Finally, the catalyzed reaction may occur in a part of the enzyme-substrate complex population, forming the product and re-creating the enzyme. Add the process representing this reaction to the model. Assign it the speed of ES*1.5E-2 Figure 68.

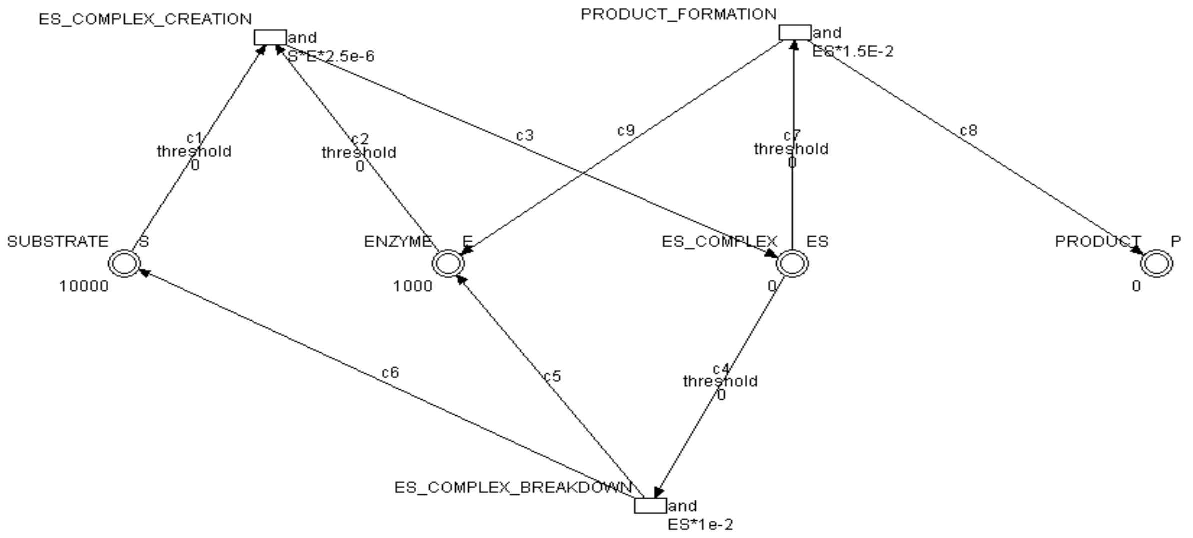
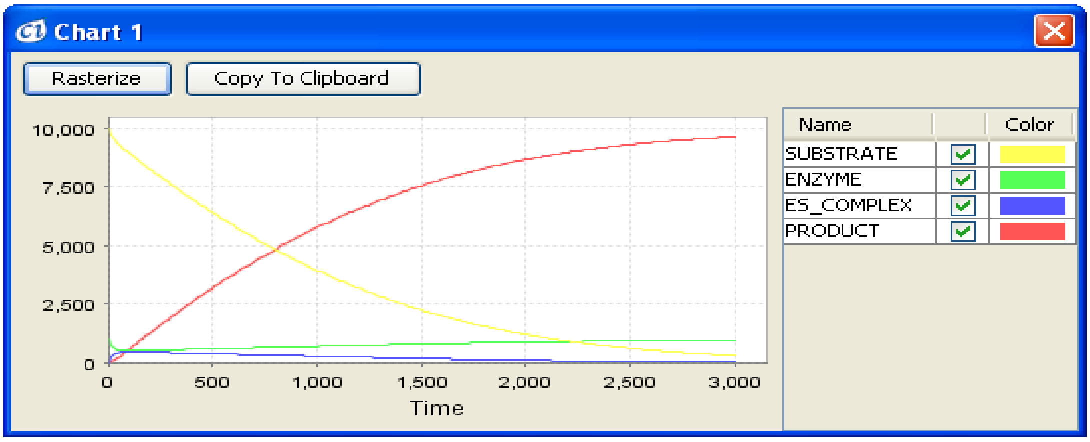


Figure 68

Running the model

In the time settings dialog, increase the simulation time to 3,000 [pt]. Add a chart displaying the values of all the entities in the model. Start the simulation. The chart should look like the one on Figure 69.



Name		Color
SUBSTRATE	☑	🟨
ENZYME	☑	🟩
ES_COMPLEX	☑	🟦
PRODUCT	☑	🟥

Figure 69

Congratulations! Modify the values of the entities, the speeds of the processes, and the simulation time. Study the system's behavior. Under some circumstances the simulation may not run (especially in the case of a very high substrate to enzyme ratio). If this happens, decrease the time step, or turn on the 'continuous weak firing' option in the simulation time settings dialog. Try to explain what caused the simulation to fail in the first place (hint: what is the criterion for firing a process in the classical Petri net formalism?).

8 SUMMARY

This tutorial has explained the installation and invoking of CI, on overview of concepts, and the editing and simulating of models. Furthermore, you have also learned how to create models based on each of reactions in living organisms.

In Tutorial II, you will learn how to create a passway combining each of reactions learned in Tutorial I, to modify models, and to give feedback of biological experiments.

References

[1] Doi A, Fujita S, Matsuno H, Nagasaki M, Miyano S (2004) Constructing biopathway models with hybrid functional Petri nets, In Silico Biology 4(3).

[2] Doi A, Nagasaki M, Fujita S, Matsuno H, Miyano S (2004) Genomic Object Net: II, Modeling biopathways by hybrid functional Petri net with extension, Applied Bioinformatics 2:185–188.

[3] Doi A, Nagasaki M, Matsuno H, Miyano S (2006) Simulation-based validation of the p53 transcriptional activity with hybrid functional Petri net, In Silico Biology 6(1-2):0001.

[4] Doi A, Nagasaki M, Ueno K, Matsuno H, Miyano S (2006) A combined pathway to simulate CDK-dependent phosphorylation and ARF-dependent stabilization for p53 transcriptional activity, Genome Informatics 17(1):112-123.

[5] Matsuno H, Doi A, Hirata Y, Miyano S (2001) XML documentation of biopathways and their simulations in Genomic Object Net, Genome Informatics 12:54–62.

[6] Matsuno H, Doi A, Nagasaki M, Miyano S (2000) Hybrid Petri net representation of gene regulatory network, Pacific Symposium on Biocomputing 5: 341–352.

[7] Matsuno H, Fujita S, Doi A, Nagasaki M, Miyano S (2003) Towards biopathway modeling and simulation, In: 24th International Conference on Applications and Theory of Petri Nets (ICATPN 2003), Volume 2679, Lecture Notes in Computer Science, 3–22.

[8] Matsuno H, Murakami R, Yamane R, Yamasaki N, Fujita S, Yoshimori H, Miyano S (2003) Boundary formation by notch signaling in Drosophila multicellular systems: experimental observations and gene network modeling by Genomic Object Net, Pacific Symposium on Biocomputing 8:152–163.

[9] Matsuno H, Tanaka Y, Aoshima H, Doi A, Matsui M, Miyano S (2003) Biopathways representation and simulation on hybrid functional Petri net, In Silico Biol, 3:389–404, `http://www.bioinfo.de/isb/toc_vol_03.html/`.

[10] Nagasaki M, (2004) A Platform for biopathway Modeling/Simulation and Recreating Biopathway Databases Towards Simulation, Ph.D Thesis, The University of Tokyo, `http://www.csml.org/download/publication/nagasaki_phd.pdf`.

[11] Nagasaki M, Doi A, Matsuno H, Miyano S, (2003) Recreating biopathway databases towards simulation, In: Computational Methods in Systems Biology, Volume 2602 of Lecture Notes in Computer Science, Springer-Verlag, 191–192.

[12] Nagasaki M, Doi A, Matsuno H, Miyano S, (2004) A versatile Petri net based architecture for modeling and simulation of complex biological processes, Genome Informatics 15(1): 180-197.

[13] Nagasaki M, Doi A, Matsuno H, Miyano S, (2004) Genomic Object Net:I, A platform for modeling and simulating biopathways, Applied Bioinformatics 2:181–184.

[14] Nagasaki M, Doi A, Matsuno H, Miyano S, (2004) Integrating biopathway databases for large-scale modeling and simulation, In: The Second Asia-Pacific Bioinformatics Conference, Volume 29 of Conferences in Research and Practice in Information Technology, Australian Computer Society, 43–52.

[15] Saito A, Nagasaki M, Doi A, Ueno K, Miyano S, (2006) Cell fate simulation model of gustatory neurons with microRNAs double-negative feedback loop by hybrid functional Petri net with extension, Genome Informatics, 17(1): 100–111.

[16] `http://www.csml.org/`