In order to be able to test the algorithm for computing the evolution graph of a Timed Continuous Petri Net the users need the following files:

- **Rpcontinue.exe** - the executable that allows computing the evolution graph of a Timed Continuous Petri Net.
- ***.dat** file - file that contains the information about one Timed Continuous Petri Net.
- **Cw3211.dll**, **32rtm.exe**, **Windpmi.386** and **Lp_solve.exe** - files needed for solving a Linear Programming Problem (used by Rpcontinue.exe).
- **Sirphyco.exe** - the graphic interface used for drawing a Petri Net and exports it in a readable form for Rpcontinue.exe.
- ***.rdp** file - files created and saved using Sirphyco.exe.

The steps required for a simulation are:

a. preparing the input data;
b. running the simulation;
c. reading the output data.

There are two ways of preparing the data and testing the algorithm of computing the Evolution Graph:

1. Using Sirphyco for all three simulation steps.
2. Preparing the data manually (or using another program), running the Rpcontinue program and reading the output data (or using it in another program).

1. USING SYRPHYCO.

Sirphyco represents an program that allows users to simulate and evaluate the comportment of different types of Petri Nets. It allows users to build a Petri Net in a graphical manner.

a) Preparing the input data

The input data for the algorithm of computing the evolution graph is represented by the Timed Continuous Petri Net and all its components: the maximal speeds for transitions, the initial marking for places and the resolution rule of each conflict. In order to have a Timed Continuous Petri Nets we need that the following requirements are satisfied:

- if the user needs a specific type of element (place, transition or arc) he must choose this type before selecting to draw the element.
- all Net's places are continuous. The user must select from the ComboBox corresponding to places (located at the top) the option "Continue";
- all transitions must be continuous. The user must select from the ComboBox corresponding to transitions (also located at the top) the option "Vitesse max. constante";
- in order to use normal arcs, the option "Ordinaire" must be selected. If there is a need of generalised arc, the option "Generalise" must be selected. If the user wants to curve an arc, he must use the CTRL button an in the same time to click on an arc. Multiple curving points can be specified for an arc;
- double clicking on an element allows the user to change parameters of that element;
- if there are conflicts (more than one transition are fed by one place), the resolution rule must be given (for each place belonging to a conflict). The resolution rule can be based on priorities and/or sharing. The resolution rule can be given accessing "Simulation/Conflits" from the menu bar. If two transitions are in a sharing, they must have the same priority level. If so, the weight of sharing can be specified.;
- in order to have a valid example the maximal speed for each transition and also the initial marking for each place must be given;

b) Running the simulation

In order to run the simulation and to obtain the evolution graph, the user must select a simulator (for Timed Continuous Petri Nets – "RdP continu a vitesses constantes") and after that click on one of the following buttons (located also on the toolbar):  ,  or  . The first button () is used to initialise the Petri Net. The second () is used to view the evolution graph step by step and the third button () is used to view all the Petri Nets evolution. In order to be able to see the evolution of the place marking, it is recommendable to insert the evolution graphics (either from "Insertion/Courbes") or directly from the toolbar (between the arc types and text). Each graphic may contain information about one or many places.

c) Reading the results

The results are presented in Sirphyco in a graphical manner: for each IB-state the markings are updated (and the place's marking evolution is shown in the graphic – if there is one graphic for that place).

2. PREPARING DATA MANUALLY (OR USING A PROGRAM)

Using a text editor, the data needed by Rpcontinue.exe can be written by the user directly. There are some examples of data in the archive that contains the Rpcontinue.exe program. The file "explanation.txt" contains the data from "Algo5c5_1.dat" file and in addition (starting with "//") comments for each line.

a) Preparing the input data

The rules for writing a **.dat** file are presented as follows:

- first line contains the name of Petri Net. Do not include in this line spaces or tabs;
- the second line contains the number of places (an integer number). (number denoted **pos**);
- the third line contains the number of transitions (an integer number). (number denoted **tran**);
- it follows **pos*tran** lines, each one corresponding to an element of the PRE matrix (the matrix that contains the weights of arcs from places to transitions; an element $PRE_{i,j}$ corresponds to the weight of arc from place P_i to transition T_j), starting with $PRE_{1,1}$ and continuing by lines;
- it follows **pos*tran** lines, each one corresponding to an element of the POST matrix (the matrix that contains the weights of arcs from transitions to places; an element $POST_{i,j}$ corresponds to the weight of arc from transition T_j to place P_i), starting with $POST_{1,1}$ and continuing by lines;
- next there are **pos** lines, each one corresponding to an initial marking of one place (starting with P_1);
- **tran** lines, each one corresponding to a maximal speed of a transition (starting with V_1 – the maximal speed for transition T_1);
- next is the resolution rule for conflicts:
  - for each place P_i (i from 1 to **pos**):
    - if the place Pi belongs to a conflict:
      - for each T_j (j from 1 to **tran**):
        - if T_j belongs to the conflict of P_i:
          - write the priority level of T_j in the conflict for place P_i.
      - if there is at least one sharing (there are at least two transitions with the same priority level) the sharing weights must be given:
        - for each T_j (j from 1 to **tran**):
          - if T_j belongs to a sharing of the P_i conflict and **j** is the first index from the sharing:
            - write the weight of T_j for P_i sharing;
            - for each T_k (k from j to **tran**):
              - if T_k belongs to the same sharing as T_j:
                - write the weight of T_k for P_i sharing.

There are two special types of elements used for building a Timed Continuous Petri Net. One is immediate transition (a transition whose maximal speed is infinite) and the other is 0^+ marking of a place. Because there is no correspondence between those elements and real numbers, in order to allow algorithms to deal with the problems regarding those elements, they were denoted by **–1**. So, if for the maximal speed, a value of –1 is read from the input file means that the corresponding transition has an infinite maximal speed. If, for an initial marking of a place, a value of –1 is read from the input file, it means that the initial marking is 0^+.

Because the algorithm of writing directly the **.dat** file is not easy to understand (especially the resolution rule), it is recommendable that users should read firs the comments from "explanation.txt" file before building the files.

b) Running the simulation

In order to realise a simulation of a Timed Continuous Petri Net the user needs to copy the file **.dat** corresponding to the Petri Net with the name "**date.dat**". The Rpcontinue.exe program reads his input data from this file.

c) Reading the results

The output data of the Rpcontinue.exe program are of two types:

- HTML files
- TEXT files

There are two HTML files: **results.htm** and **short_results.htm**. These files are obtained even if the simulation takes place under Sirphyco.

The **results.htm** file contains all the information regarding the steps of the algorithms applied and all intermediary values for variables.

The **short_results.htm** file contains only information regarding each IB-state of the Petri Net evolution. This information is about the initial marking, the vector of transitions speeds and the evolution equation for each place during the respective IB-state.
The TEXT files contain information about the evolution of each place and each transition. Files are denoted P**i**.rez, where **i** takes values from 1 to the number of places and T**j**.rez, where **j** takes value from 1 to the number of transitions in the respective Petri Net.
Each P**i**.rez file contains the following information:

- on the first line, the number of IB-states. (denoted **IBnr**)
- IBnr-1 groups of lines with the following composition:
  - initial marking for the IB-state (one line);
  - IB-state duration (one line);
- initial marking of the last IB-state (one line);
- the balance of the place for the last IB-state (one line).

Using the information from these files, the evolution of each place marking can be presented.
There is also one file called **P.rez** that presents the evolution of all places in the following form:

- on the first line, the number of IB-states. (denoted **IBnr**)
- IBnr-1 lines with the following composition:
  - IB-state duration (first column) separated by one tab from the following columns;
  - initial marking for each place (starting from 1 to **pos** – number of places) for the IB-state (separated by one tab);
- one line with the following composition:
  - on tab (the last IB-state has an infinite duration)
  - initial marking for each place (starting from 1 to **pos** – number of places) of the last IB-state (separated by one tab);
- one line with the following composition:
  - on tab (the last IB-state has an infinite duration)
  - the balance of all the places for the last IB-state (starting from 1 to **pos** and separated by columns).

Each T**j**.rez file contains the following information:

- on the first line, the number of IB-states. (denoted **IBnr**)
- IBnr-1 groups of lines with the following composition:
  - the transition speed for respective IB-state (one line);
  - IB-state duration (one line);
- the transition speed for the last IB-state (one line);