Class 11

Learning Objectives

- Understand the basic concepts behind Petri nets
- Be able to model basic processes using Petri nets
- Be able to explain basic notions of well-formedness of Petri nets and apply them to Petri nets

Readings

This class will introduce you to a basic diagramming technique that has been used to describe, analyze, and simulate business processes and workflows. Petri nets are a technique that was invented in 1962 by Carl Adam Petri. Petri nets are conceptually very simple; there are only two kinds of symbols: circles and boxes. These are technically called “places” and “transitions”, respectively.

Petri nets are also very general, i.e. they can be, and have been, used to describe the behaviour of all sorts of things. However, what makes Petri nets really useful is the fact that people have developed formal ways of proving desirable properties. For example, when we describe a travel reservation process, we would like to be certain that the process will eventually finish. We might also like to ensure that at no time will a single travel agent have more than 10 applications in their in-box. These properties have technical terms: The first is called “liveliness”, the second is called “boundedness”. In this course, we are not concerned with how we can prove these properties; it's enough to know that we can (and that we have software tools that will do it for us).

Reisig, W. and Rozenberg, G.: Informal Introduction to Petri Nets

This article is a very elementary introduction to Petri nets. The first example introduces circles and boxes as representing passive and active components (also called by their technical names “places” and “transitions”, respectively.). While the article shows the application to a library system that deals with books and book borrowing, the generic nature of Petri nets means this can easily be applied to business processes as well. Boxes (“transitions”) represent generic active components and we can therefore represent the actions or activities of a process as boxes. In contrast, circles represent passive components. These are more difficult to apply to business processes. We could think of them as in-boxes (“pre-places”) and out-boxes (“post-places”) for the actions that will contain work items to be processed. However, we will see later (in the next classes) that we can describe business processes without the circles (“places”), but they serve an important role in explaining how things behave.

Section 1.2 then explains how things behave by introducing the idea that objects (technically called “tokens”) can occupy the passive components (circles, “places”). The active components (“transitions”) then take objects from their input places and put them into their output places. This behaviour is further explained in the second part of the article (Example 2) which introduces the concept of boxes (“transitions”) as being “enabled”. Being enabled means that there is at least one object (“token”) in each of the transition's input places (“pre-places”). Note that in a complicated Petri net, there may be multiple transitions that are enabled at the same time. A transition occurs (or “fires”) when it is enabled. It then takes one object (“token”) from each of the its input places (“pre-places”) and puts one object (“token”) onto each of its output places (“post-places”). Note that, as in Figure 2.5, places may contain more than one token.

As you read this article, you should pay attention to the different ways in which places and transitions are used in the two examples. In the library (Example 1), the places represented real (physical) places, e.g. the order-counter, the pick-up counter (Figure 1.6), the tokens represented real (physical) objects, e.g. books and index cards, and the transitions represented activities. In contrast, in the container station (Example 2), the places represented the current condition of the crane, e.g. loaded, moving left, etc. (Figure 2.2), the tokens did not represent anything other than in which of these states the crane was in, certainly no physical objects as in Example 1. In fact, the article uses terminology of “pre-condition” and “post-condition” to describe pre-places and post-places (page 8).

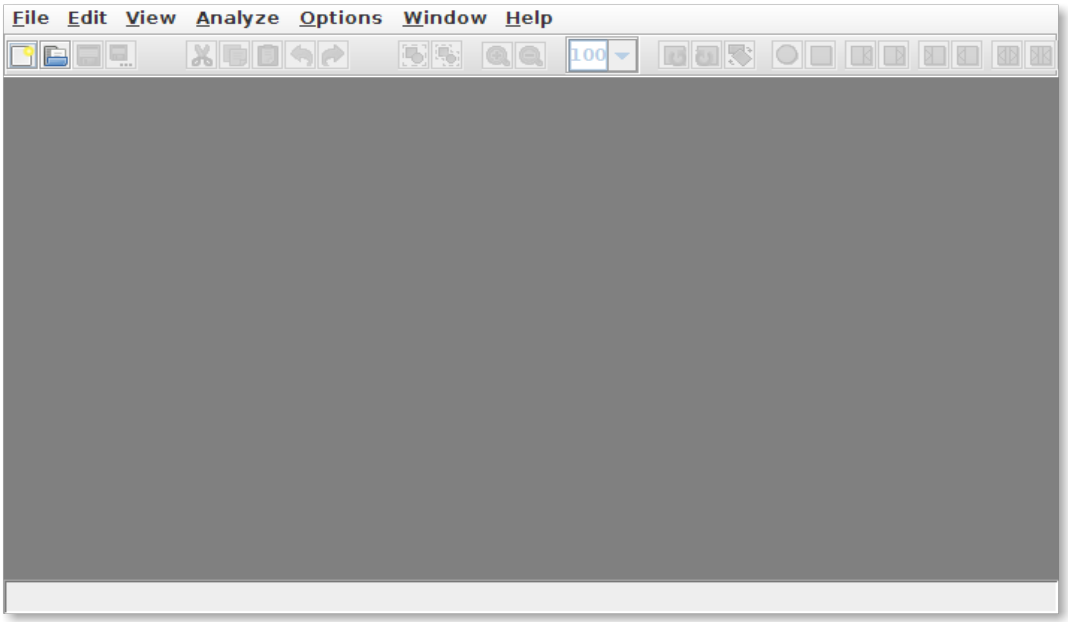
Here is a little table that summarizes the terminology in this article:

	
Circles	Boxes
Passive components	Active components
Places	Transitions

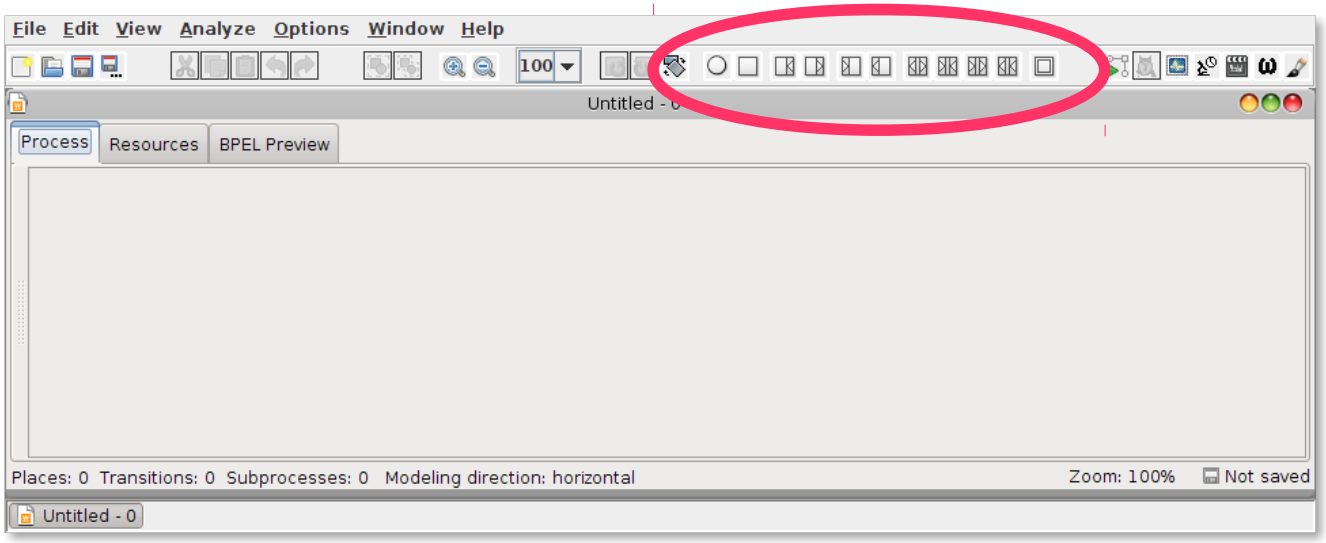
As you read the article, you may want to think about such issues as what else can be modelled with Petri nets and how are places and transitions interpreted? Is there something missing in Petri nets that you think might be needed when it comes to modelling business processes?

WoPeD Software

The WoPeD Software is a free, open-source software for drawing and analyzing workflows using Petri nets and Workflow nets. WoPeD stands for **Wo**rkflow **Pe**tri net **D**esigner. Download the version for your operating system (Windows or Mac) it from the course website, install it, and start it. You will see the basic WoPeD window:



To create a new diagram, select File -> New in the menu. You should now see a blank diagram window with a toolbar on top (highlighted in red below).

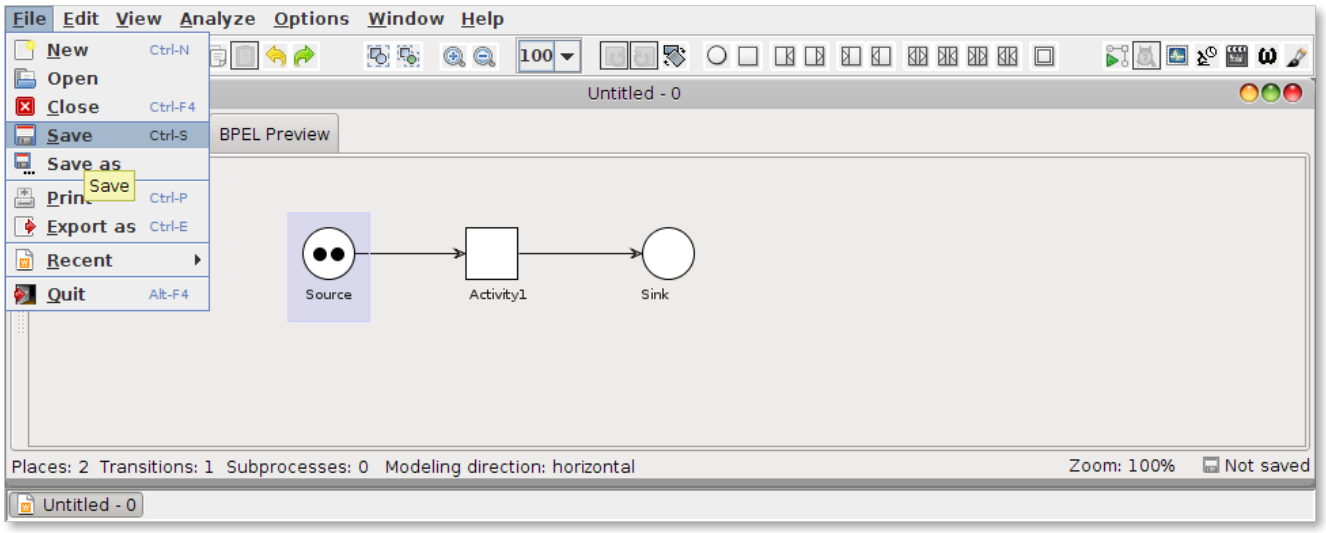


You can add places and transitions to the diagram by selecting them in the toolbar and clicking at the place in the diagram where you want to place them. Notice that the toolbar provides the basic Petri net place and transition symbol (circle and box) but also the Workflow net symbols shown in Section 2.3.2 (page 55, Figure 2.16). Once you have added a place or transition, you can give it a name.

You can add arcs or arrows between places and transitions by selecting the center of a place or transition, pressing and holding the mouse button and dragging to the place or transition that you want to connect. When you highlight the center of a place/transition, the mouse pointer changes to a pointing hand. If you drag to an empty place in the diagram and release the mouse, WoPeD will create a new place or transition for you, as appropriate.

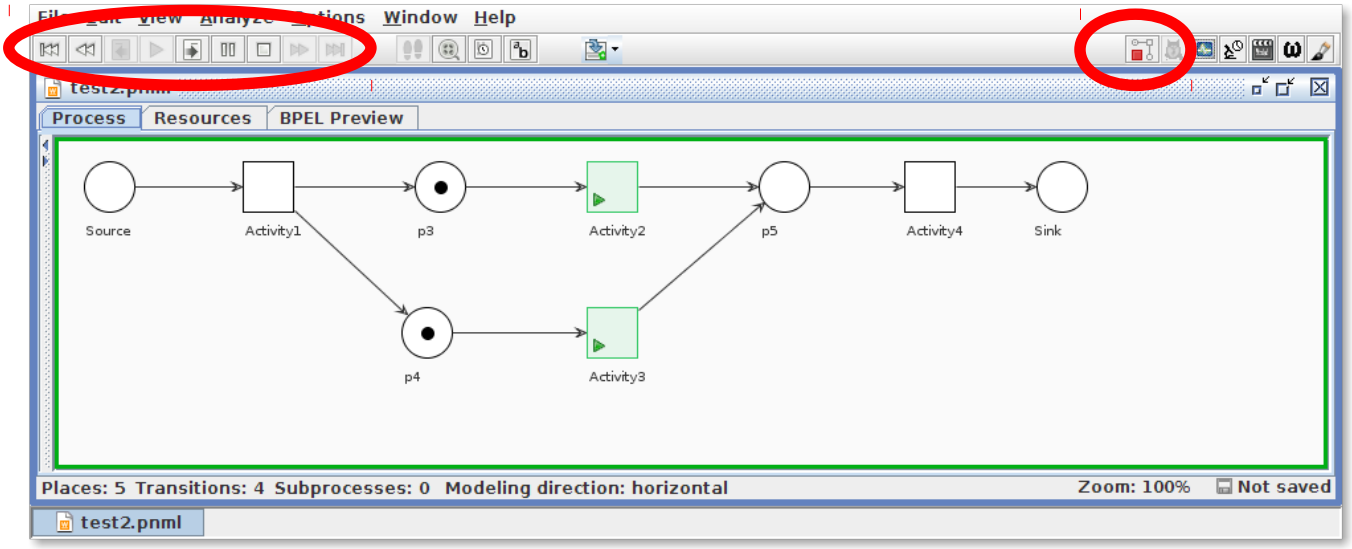
You can add tokens to places by right-clicking on a place and selecting “Add token” from the menu.

Save your diagram by selecting “File -> Save” from the main menu.



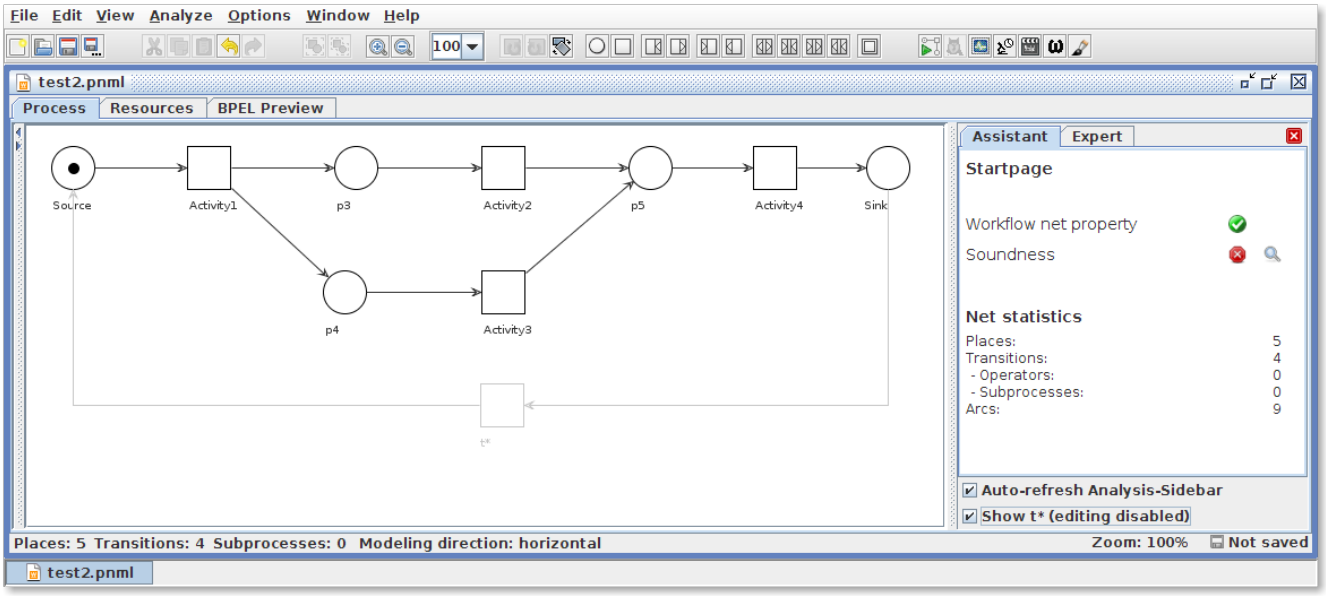
Once you have modelled a process/workflow (remember, a workflow net has exactly one source and one sink place), add a token to the source. You can now simulate the workflow and watch how transitions are enabled and fire. For this, click the “Start Tokengame” button in the toolbar, indicated below:

During the simulation, the active Petri net/Workflow net is highlighted in green;' enabled transitions are also highlighted in green and can be fired by clicking on them. The toolbar now shows controls to take a step backward, reset, pause the simulation, etc. (see below). The “Start tokengame” button in the toolbar has become a “Stop tokengame” button (see below).



Finally, WoPeD allows you to check whether the Petri net you have drawn is a workflow net (satisfies definition 7 on page 54) and is sound (satisfies definition 8 on page 55). Do do this, select the button “Semantic Analysis” from the toolbar:

The results of the semantic analysis are shown to the right of the diagram. Clicking on the magnifying glass symbol will provide more detail about the two conditions. Alternatively, you can switch to the expert view for the analysis results. At the bottom of the analysis results window is a check box with which you can show the t* transition that is mentioned in the second part of definition 7 (page 54).



Review questions and exercises:

Using the WoPeD software, do the following:

- Chapter 2, Exercise 1 (Instead of YAWL, assume Workflow nets)
- Construct a simple Petri net that is not live (Definition 3). Identify why it is not live. Verify this by means of the token game.
- Construct a simple Petri net that is not bounded (Definition 4). Identify why it is not bounded. Verify this by means of the token game.