**Introduction to hierarchical Petri nets**

**Motivation**

Creating large, intricate nets can be a cumbersome task. But similar to modular programming, the construction of CP-nets can be broken into smaller pieces by utilizing the facilities within CPN Tools for creating substitution transitions. Conceptually, nets with substitution transitions are nets with multiple layers of detail - you can have a somewhat simplified net that gives a broad overview of the system you are modelling, and by substituting transitions of this top-level net with more detailed pages, you can bring more and more detail into the model.

CP-nets can be organized into hierarchies of layers. This gives rise to hierarchical CP-nets that, although in all perfectly equivalent to ordinary CP-nets, can facilitate the management of large-scale models. In hierarchical CP-nets the modeler can construct a large model by using a number of small CP-nets. This is somewhat similar to the situation in which a programmer constructs a large program consisting of independent modules.

Here is a quick comparison between ordinary CP-nets and hierarchical CP-nets:

- Given hierarchical CP-net, it is easy to construct an equivalent CP-net and vice versa.
- A hierarchical CP-net and its equivalent non-hierarchical CP-net have exactly the same set of markings, steps and occurrence sequences.
- A hierarchical CP-net and its equivalent non-hierarchical CP-net are behaviorally equivalent.
- Computer analysis with many software tools takes almost the same time for both a hierarchical CP-net and its equivalent non-hierarchical CP-net.

Thus, we can describe the core of our constructions using ordinary CP-nets with the understanding that models for large and complicated MINs may be safely decomposed, on convenience, into layers as to form hierarchical CP-nets.

**Substitution transitions**

In hierarchical nets there is a method by which a transition can represent an entire piece of net structure, so that the net containing the transition executes as if the logic that the transition represents were physically present at the location of the transition. Such a transition is called a **substitution transition**.
Substitution transitions add nothing fundamentally new. Everything that can be done with them can also be done by using fusion places. But like fusion places, substitution transitions add so much convenience that they can make the difference between modelling feasibility and total impossibility.

Let us consider a substitution transition named Reverse which stands for a net that is used to reverse a list of integers. A small blue tag, called a subpage tag, is associated with the substitution transition Reverse.

Arcs surrounding substitution transitions may or may not have inscriptions. However, these inscriptions have no semantic meaning.

**Fusion places**

CPN Hierarchy supports a method for defining sets of places so that anything that happens to each place in a set also happens to all the other places in the set. The places are then functionally identical. Such places are called fusion places, and a set of fusion places is a fusion set.

Fusion adds nothing fundamentally new. If all the members of a fusion set are on the same page, we could replace the set with a single place and connect to it all the arcs that are connected to any member of the set. If the members are on different pages, we could copy everything on the several pages to a single page, and again collapse the set.

Conversely, if a net contains a place that has many arcs connecting to it, or requires very long arcs to reach it, we can unfold it into several places, on the same or different pages, and so simplify the net's graphical structure without changing its meaning. Such unfolding is a common event during the process of CP-net development. Frequently the need for it can be anticipated, and fusion places can be used from the beginning.

**7.1.3 Subpages and superpages**

A page that contains a substitution transition is called a superpage. The page named Top in the figure above is a superpage.
When a CP-net uses a substitution transition the logic that the transition represents must be kept somewhere. It is kept on a page called a subpage, and the logic on the subpage is called a subnet or a submodel.

The page named Reverse is the subpage associated with the substitution transition Reverse.

Each substitution transition is said to be a supernode of the corresponding subpage.

The hierarchical structure of a net, i.e. the relationships between subpages and superpages, is shown in the index entry for the overview of the net. The name of a subpage is always shown below the name of its superpage, and the name of the subpage is always indented with respect to the name of the superpage.
7.1.4 Ports and sockets

Superpages and subpages are connected by equating places on the two pages using special-purpose fusion sets. These special fusion sets always contain only two places. A place that belongs to such a fusion set is called a port if it is on a subpage, and a socket if it is on a superpage.

7.1.4.1 Sockets

A socket is a place that is a neighbor to a substitution transition, i.e. there is always at least one arc between a substitution transition and a socket. Places Begin and End on page Top are sockets.

7.1.4.2 Ports

A port-type tag is associated with each port on a subpage. Places Begin and End on page Reverse are ports.

There are three kinds of port-type tags: In-tags, Out-tags, and I/O-tags.
7.1.4.3 Port/Socket assignments

Port/socket assignments are used to define how a subpage should be "glued together" with the surroundings of its supernode. Each socket must be assigned to a port on the corresponding subpage. A port with an In-tag must be assigned to a socket which is an input node (and not an output node) of the substitution transition. Analogously, an Out-tag indicates that the port must be related to a socket which is an output node (and not an input node), while an I/O-tag indicates that the socket must be both an input and output node.

Note that the arcs surrounding a port do not have to correspond to the port-type tag. For example, there may be arcs from a transition to a port that has an In-tag, i.e. the port may be an output node for a transition on the subpage even though it has an In-tag.

To learn about how port/socket assignments are indicated, read about the graphical feedback for port-type tags.

7.1.4.4 Markings of ports and sockets

Tokens that are present on a socket in a superpage are simultaneously present on the corresponding port on the subpage, and vice versa.

For example, the initial marking of the place Begin on (super)page Top is the same as the initial marking of the place Begin on the (sub)page Reverse.

7.1.4.5 Using a subpage more than once

The real power of substitution transitions lies in the fact that a subpage need not be the value of only one substitution transition: it can be used repeatedly, as the value of any number of substitution transitions on any number of superpages.

If a subpage is the value of more than one substitution transitions, there are said to be multiple instances of the subpage. Analogously, there are also multiple instances of the places and transitions on the subpage. The marking of each instance of a subpage can be completely independent from the markings of other instances of the same subpage.

Here is a net with two instances of subpage Reverse, as indicated by the (1) and (2) following the page name. Note that the markings of the two instances of
the place Begin on the page Reverse are different: they are the same as the initial markings of the corresponding sockets.

To learn about how to find out which subpage instance is associated with a substitution transition, read about the graphical feedback for subpage tags.