Abstract

It is a common practice in system design to build a system from its given behaviour. It is the general purpose in system analysis to get to know system behaviour from its structure. A common problem in both the two areas is how to specify, or to represent, system behaviour? This paper propose PROCESS PERIOD as the basic means of describing system behaviour, provided Petri Nets are used to model the system. A finite system can only have finite periods, and the total number of its periods is also finite. All processes, finite or not, can be constructed from, or decomposed into, periods.

Algorithm XY, contained in the appendix, is used to develop periods from a given system. The problem of constructing or reconstructing a system from the given behaviour is also discussed.

1 Introduction

Many systems can be modeled by Petri nets, and once this is done, the execution of such systems can also be recorded by Petri nets, or to be precise, by occurrence nets. Processes are just morphisms (mappings) from occurrence nets to the net modeling the system. Thus, occurrence nets are the basis in Net Theory to describe system behaviour while the full set of processes is the complete description. The problems with occurrence nets are:

1. A finite system can have infinitely many occurrence nets,

2. In case transitions without input places or output places are included in a system, the system may produce uncountably many occurrence nets.

3. Different systems can share exactly the same occurrence nets, e.g. all N-season systems \((N \geq 3)\) share the net shown in figure 1 and its sub-nets as their occurrence nets.
To surprise people by inventing some extreme situations is not our purpose, thus, we are confined to consider only those finite system nets whose transitions do have input places as well as output places. Besides, all system nets should be both simple and pure, i.e. every two elements in a system are structurally distinguishable from each other (being simple) and no element can be both input and output of some other element (being pure). (See [3].) As for occurrence nets, only those which are observable, or equivalently which have full state spaces (See [4] and [9]) are of interest.

With all these restrictions imposed on system nets and occurrence nets respectively, we still have to deal with infinitely many occurrence nets in describing the behaviour of a finite system. It is good on the one hand, because this means that finite nets can be used to model very complicated behaviour. On the other hand, however, this brings with it difficulties into system behaviour description.

Is it possible to construct infinitely many occurrence nets with finitely many basic ones? The answer is definitely positive because all occurrence nets of a particular finite system are constructed from the finitely many elements of that system.

We are not satisfied with this answer, because, although individual elements are structurally basic at the level of implementing the system, they are usually not basic at the level of describing system behaviour. They are details. It is not rare that many parts are combined to perform a certain function, thus these parts will stay together in all processes. It is such combinations which should be taken as being behaviourally basic, not the individual parts. Roughly speaking, this is the idea of process periods, not in the sense of being periodical, but in the sense of standing for a period of time in all processes. It is an extreme situation that, no matter what the initial marking is, each individual transition should be taken as a period. To the author’s belief, this means that the system is modeled at a rather high level with many details omitted.

In the next section we will recall some definitions and terminology from the literature. In section 3, Algorithm XY is introduced to see how process periods can be generated from given system nets. Section 4 shows various situations in executing XY. Theorems and propositions are given in section 5 together with necessary definitions. The problem of system reconstruction is discussed in section 6 followed by acknowledgement and references.

2 Basic Definitions and Terminology

2.1 (Directed) Petri Nets

Definition $N = (S, T; F)$ is called a (directed) Petri Net or Net for short if:

1. $S \cup T \neq \emptyset$
2. $S \cap T = \emptyset$