Modeling of the Communication Protocol by Means of Colored Petri Nets. Case Study

Inga Titchiev

Abstract: In this paper we present a small example describing a communication protocol by which a sender can transfer a number of packets to two receivers, using Colored Petri Nets. The problem is that the medium may lost the packets and packets may overtake each other. Hence, it may be necessary to retransmit packets and to ignore doublets and packets that are out of order. Packets also cannot reach desired receiver.
Keywords: Colored Petri Nets, communication protocol, verification, State space, Model checking, behavioral modeling.

1 Introduction

Colored Petri Nets (CPN) is a language for the modeling and validation of systems in which concurrency, communication and synchronization plays a basic role. Many system development projects are concerned with concurrent systems. The behavior of concurrent systems is complex. The scheduling of the processes involved is difficult to investigate [2]. Concurrent systems are challenging to design, tests, and debug. In this connection Colored Petri nets have been developed as a modification of Predicate/transition-nets, in order to avoid some technical problems which arise when the method of place-invariants is generalized to apply for Predicate/transition-nets. A CPN model of a system is an executable model representing the states of the system and the events (transitions) that can cause the system to change state.

With Colored Petri Nets (CP-nets) it is possible to use *data types* and complex *data manipulation:*

1. Each token has attached a data value called the *token color.*

2. The token colors can be *investigated* and *modified* by the occurring transitions.

We use a communication protocol since it is easy to explain and understand, and because it involves concurrency, non-determinism, communication, and synchronization which are key characteristics of concurrent systems. During the construction of a CP-net, simulations

are used to validate the CPN model, i.e., to check that it has the expected behavior.

2 Colored Petri Nets. Definitions

Advantage of constructing a model by means of Colored Petri Nets:
1. Intuitive design and operation of the system.
2. Completeness: Results in a more complete design.
3. Correctness: Removing errors and ambiguities in the design stage.
4. Properties can be checked before implementation.

A CPN model of a system describes the states of the system and the events (transitions) that can cause the system to change state. By making simulations of the CPN model, it is possible to investigate different scenarios and explore the behaviors of the system. Very often, the goal of simulation is to debug and investigate the system design.

Definition 1 [3]: A **CP-net** is a tuple CPN = (Σ, P, T, A, N, C, G, E, I) where:

(i) Σ is a finite set of non-empty **types**, also called color sets.

(ii) P is a finite set of **places**.

(iii) T is a finite set of **transitions**.

(iv) A is a finite set of **arcs** such that:

• $P \cap T = P \cap A = T \cap A = \emptyset$.

(v) N is a **node** function. It is defined from A into $P \times T \cup T \times P$.

(vi) C is a **colour** function. It is defined from P into Σ.

(vii) G is a **guard** function. It is defined from T into expressions such that:

• $\forall t \in T$: [Type(G(t)) = B $\wedge$ Type(Var(G(t))) $\subseteq \Sigma$].

(viii) E is an **arc expression** function. It is defined from A into expressions such that:

• $\forall a \in A$: [Type(E(a)) = $C(p)_{MS}$ $\wedge$ Type(Var(E(a))) $\subseteq \Sigma$] where p is the place of N(a).

(ix) I is an **initialisation** function. It is defined from P into closed expressions such that:

• $\forall p \in P$: [Type(I(p)) = $C(p)_{MS}$].

Definition 2 [3]: A step Y is **enabled** in a marking M iff the following property is satisfied:

$\forall p \in P$: E(p,t)<b> $\leq$ M(p).

When a step Y is enabled in a marking M it may **occur**, changing the marking M to another marking M′, defined by:

$\forall p \in P: M'(p) = (M(p) - E(p,t)\langle b \rangle) + E(t,p)\langle b \rangle$.

M′ is **directly reachable** from M. This is written: $M[Y\rangle M'$.

3 Communication protocol example

The communication protocol consists of a *sender* transferring a number of *data packets* to two *receivers*. Communication takes place on an unreliable network, i.e., packets may be lost and overtaking is possible. Packets also cannot reach desired receiver. The protocol uses sequence numbers, acknowledgements, and retransmissions to ensure that the data packets are delivered exactly once and in the correct order at the receiving end. The protocol uses a stop-and-wait strategy, i.e., the same data packet is transmitted until a corresponding acknowledgement is received. The data packets consist of a sequence number and the data to be transmitted. An acknowledgement consists of a sequence number specifying the number of the data packet expected next by the receiver.

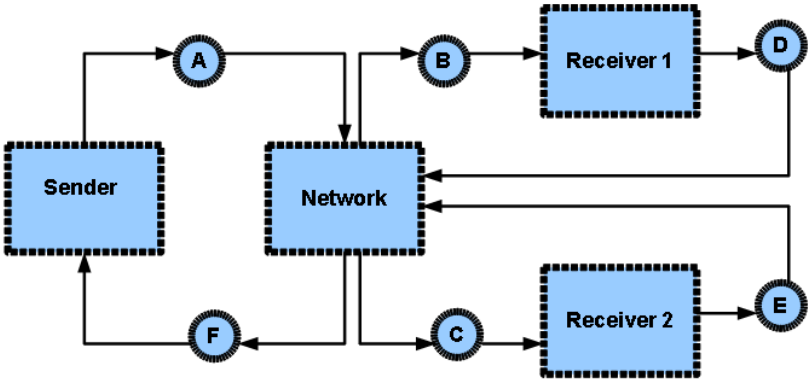

Figure 1. Conceptual scheme of the communication protocol

The simulator exploits the locality property of Petri nets, which ensures that the occurrence of a transition only affects its immediate surroundings. This ensures that the number of steps executed per second in a simulation is independent of the number of places and transitions in the CPN model. This means that simulation scales to large CPN models.

After simulation is provided information about the markings that are reached and the binding elements that are enabled and occur during a simulation. Simulation can only be used to consider a finite number of executions of the model being analysed. This makes simulation suited for detecting errors and for obtaining increased confidence in the correctness of the model.

4 Conclusion

To cope with the complexity of modern concurrent systems, it is important to provide methods that enable debugging and testing of central parts of the system designs before implementation and deployment. Developing an executable model usually leads to a more complete specification of the design and makes it possible to make a systematic investigation of scenarios which can significantly decrease the number of design errors. We have illustrated the use of CP-nets for modeling and validation of a communication protocol.

The reader interested in a complete treatment of the modeling language and analysis methods may consult the works [3, 4, 5]. A CPN Tools[6] was used for system checking.

References

[1] Guţuleac, E. *Evaluarea performanţelor sistemelor de calcul prin reţele Petri stochastice.* Ed. Tehnica-Info, Chişinău, 2004, 276p.

[2] Păstrăvanu, O. *Aplicaţii ale reţelelor Petri în studierea sistemelor cu evenimente discrete.* Ed. Gh. Asachi, Iaşi, 2002, 238p.

[3] Jensen. K. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1: Basic Concepts*. Springer-Verlag, 1992.

[4] Jensen. K. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 2: Analysis Methods*. Springer-Verlag, 1994.

[5] Jensen. K. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 3: Practical use*. Springer-Verlag, 1997.

[6] CPN Tools. www.daimi.au.dk/CPNTools/.

Inga Titchiev[1]

[1] Institute of Mathematics and Computer Science
E-mail:caminga2002@yahoo.com