"$QN + PN = QPN$" Combining Queueing Networks and Petri Nets

Falko Bause
Informatik IV, Universität Dortmund
Postfach 500 500, FRG - W 4600 Dortmund 50

Topics: higher-level net models, timed and stochastic nets.

Abstract

System analysis is often done due to qualitative and quantitative aspects. Queueing networks and Petri nets are suitable and wide spread model worlds for such a form of analysis, because for both a well founded theory exists. The problem is that Petri nets and queueing networks are only suitable for qualitative analysis resp. quantitative analysis. So the analyst has to create two models of the same system. In recent years several model worlds have been developed, combining qualitative and quantitative aspects thus reducing the modelling effort. The stress nowadays is on timed and stochastic Petri net models. The main disadvantage of these model worlds lies in the difficulties of describing scheduling strategies with Petri net elements.

This paper describes a new model world, the QPN world, combining queueing networks and Petri nets for qualitative and quantitative analysis eliminating this disadvantage. It is shown that queueing networks, Petri nets and timed and stochastic Petri nets are special cases of the QPN world.

Contents

1 Introduction

Queueing networks and Petri nets are well known model worlds for the description and analysis of a variety of systems, e.g. computer systems [22] and flexible manufacturing systems [23]. Petri nets are used for qualitative analysis (in the sense of global correctness) whereas queueing networks are used for performance evaluation (quantitative analysis). In practice systems are analysed due to their qualitative and quantitative aspects. For performance analysis Petri nets can't be used, because of the lack of time. On the other side queueing networks are not suitable for qualitative analysis, for they miss e.g. the possibility of describing forking and joining of processes.

Some model worlds, like e.g. timed and stochastic Petri nets and extended queueing networks, try to eliminate these deficits. In [24] timed and stochastic Petri nets and extended queueing networks are compared due to their expressive power and evaluation efficiency. One important result of the discussion there is that timed and stochastic Petri nets have a greater modelling power if the description of scheduling strategies is integrated in an appropriate way.

The subject of this paper is to combine (timed and stochastic) Petri nets and queueing networks thus integrating an easy description of scheduling strategies into the Petri net world. The article is structured as follows. Section 2 presents basic definitions. In section 3 we describe existing model worlds for the combined description of qualitative and quantitative aspects and list their advantages and disadvantages. Section 4 gives the definition of a new model world, the QPN world, combining queueing networks and Petri nets. Section 5 gives some examples of QPN models showing their suitability for the modelling of systems. In section 6 we show that queueing networks, Petri nets and timed and stochastic Petri nets are special cases of the QPN world.

2 Basic definitions

We assume that the reader is familiar with Petri nets, so we will put the stress on the explanation of queueing networks.

2.1 Petri nets

In this paper coloured Petri nets defined by K. Jensen [16] are used.

Definition 1 (CPN) *A coloured Petri net (CPN) is a 6-tuple CPN=(P,T,C,I^-,I^+,M_0), where*

- *P is a finite and non-empty set of places,*
- *T is a finite and non-empty set of transitions,*
- *$P \cap T = \emptyset$,*
- *C is a colour function defined from $P \cup T$ into non-empty sets,*
- *I^- and I^+ are the backward and forward incidence functions defined on $P \times T$ such that $I^-(p,t), I^+(p,t) \in [C(t) \to C(p)_{MS}], \forall\ (p,t) \in P \times T$[1],*
- *M_0 is a function defined on P describing the initial marking such that $M_0(p) \in C(p)_{MS}, \forall p \in P$.*

[1]The subscript MS denotes multi-sets. $C(p)_{MS}$ denotes the set of all finite multi-sets of $C(p)$.

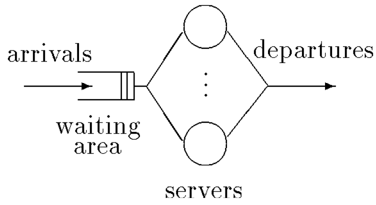

Figure 1: A queue (station)

Every CPN can be unfolded into an uniquely determined place/transition net[2]. Thus all well known notions of place/transition nets [18] can be defined for CPNs via this transformation.

Definition 2 (Unfolding of CPNs) *The unfolding of a CPN=(P,T,C,I^-,I^+,M_0) is performed in the following way:*

1. *$\forall\ p \in P, c \in C(p)$ create a place (p,c) of the place/transition net.*
2. *$\forall\ t \in T, c' \in C(t)$ create a transition (t,c') of the place/transition net.*
3. *Define the incidence functions of the place/transition net as*
   $I^-((p,c)(t,c')) := I^-(p,t)(c')(c)$[3]
   $I^+((p,c)(t,c')) := I^+(p,t)(c')(c)$.
4. *The initial marking of the place/transition net is given by $M_0((p,c)) := M_0(p)(c)$.*

$PN = (\bigcup_{p\in P}\bigcup_{c\in C(p)}(p,c), \bigcup_{t\in T}\bigcup_{c'\in C(t)}(t,c'), I^-, I^+, M_0)$ *is the unfolded CPN.*

A well founded theory exists for the qualitative analysis of CPNs. In [15] the reduction method for place/transition nets [10] is transferred to CPNs. Invariant analysis of CPNs is e.g. described in [13] and an efficient technique for construction of the reachability graph exploiting symmetries in the CPN is presented in [11].

2.2 Queueing networks

A queueing network [22] consists of a set of interconnected queues. Each queue (station; figure 1) represents a service center where arriving customers receive service by a server. Each station has one or more servers. For customers arriving at a station, service will begin immediately if there is a free server, otherwise the customer is forced to wait in a waiting area or the service of another customer is preempted, e.g. if the arriving customer has a higher priority. The order specifying which customer will receive its service next is described by a scheduling strategy. Typical scheduling strategies are e.g. FCFS (first

[2] A place/transition net is a 5-tuple $PN = (P,T,I^-,I^+,M_0)$ where

- P is a finite and non-empty set of places,
- T is a finite and non-empty set of transitions,
- $P \cap T = \emptyset$,
- $I^-, I^+ : P \times T \to \mathbb{N}_0$ are the backward and forward incidence functions,
- $M_0 : P \to \mathbb{N}_0$ is the initial marking.

[3] $I^-(p,t)(c')$ is a multi-set of $C(p)_{MS}$ and $I^-(p,t)(c')(c)$ denotes the number of occurrences of the element c in this multi-set.

come first served), which is the usual situation in real life or RR (round robin), which is the usual scheduling for jobs requesting service at the CPU.

Each customer demands a certain amount of service, which is specified by the length of time the server is occupied by the customer. This service time is described by a random variable thus representing the load the station has to cope with, due to this customer. Customers with the same service time distribution are grouped together in a *class*. The usual notation for queues is *A/B/m-"scheduling strategy"*, where A denotes the probability distribution function (pdf) specifying the interarrival times of customers, B is the pdf of service times and m is the number of servers. Typical examples are M/M/1-FCFS and M/G/1-FCFS, where M[4] denotes the negative exponential distribution function and G denotes general pdfs. If the interarrival times are not known a priori we will omit their description, e.g. -/M/1-FCFS.

The interconnection of queues in a queueing network is described by the paths customers may take determined by routing probabilities. Being served at station i, a customer of class s proceeds to station j becoming a class t customer with probability $r_{i,s;j,t} \in [0,1]$. The set of classes can be partitioned into *chains*. Two classes of customers belong to the same chain, if a customer of one class might become a member of the other class and vice versa. Thus an equivalence relation is defined on the set of classes.

Queueing networks are classified into two categories, a queueing network is

- open, if arrivals and departures from and to the external environment of the network are possible. This can be described by introducing a ficticious node 0 representing the environment, which creates customers usually due to a Poisson process. Together with the routing probabilities $r_{0;j,t}$ and $r_{i,s;0}$ the arrival and departure at each station is specified.
- closed, if no external arrivals and departures are allowed.

A chain h with $h \cap \{t \mid \exists \text{ queue } i : (r_{0;i,t} > 0 \text{ or } r_{i,t;0} > 0)\} \neq \emptyset$ or $= \emptyset$ is also said to be open resp. closed. Note that closed chains might exist in an open queueing network.

Performance measures of common interest are information on queue length, throughput and utilization of stations. Analysis is usually done by examination of the Markov process' steady state distribution [12, 22]. Queueing networks are very popular in the field of performance evaluation (quantitative analysis), because for the calculation of this steady state distribution very efficient algorithms exist for certain classes of queueing networks (product form solution, approximation [22]). The main problem is that queueing networks are only capable of describing the routing of customers in the net, but there is no facility to model e.g. forking and joining of processes.

To summarize this section we will list the main advantages and disadvantages of both model worlds for a starting point of the discussion in the next section (see figure 2). The list shows that it is desirable to create a model world which inherits the best of these two worlds.

3 Model worlds for qualitative and quantitative analysis

To overcome the mentioned disadvantages several efforts were undertaken in the last two decades. The most important and nowadays very popular trend is to add time and probability characterizations into Petri nets [1, 2, 3, 4, 5, 17].

[4] M stands for Markov.

Advantages	Disadvantages
Petri nets	
• easy description of – splitting and – synchronization and – blocking of processes. • efficient algorithms for proving certain qualitative properties	• timing aspects are not considered • description of scheduling strategies might get too complex (cf. section 3)
Queueing networks	
• timing aspects are considered • easy description of scheduling strategies • efficient algorithms for calculating performance measures for certain classes of queueing networks	• no constructs for the description of – splitting and – synchronization and – blocking of customers

Figure 2: Advantages and disadvantages of Petri nets and queueing networks

3.1 Timed and stochastic Petri nets

There are two principal ways of integrating timing aspects into Petri nets:

- specification of a dwelling time for tokens on a place (Timed places Petri nets; TPPNs) [21, 25],
- specification of a firing delay for enabled transitions (Timed transition Petri nets; TTPNs) [1, 17].

In recent years the stress is on TTPNs. TTPNs can be classified into two categories due to their handling of enabled transitions. In TTPNs with *preselection policy*, input tokens needed for firing are destroyed by the transition before starting the firing phase. After a certain firing delay the firing phase terminates by creating the output tokens. So the firing process of a transition in TTPNs with preselection is not atomic like in ordinary Petri nets. If several transitions are in conflict the one to destroy its tokens first is chosen randomly. In TTPNs with *race policy* each enabled transition competes for its input tokens with eventually conflicting transitions and the fastest transition will fire first ("race"). Here firing remains an atomic action.

The most important representatives of TTPNs are stochastic Petri nets, e.g. generalized stochastic Petri nets (GSPNs) [2, 3], describing Markov processes. Due to their wide spread use we will give the definition of GSPNs as a representative of the class of stochastic Petri nets with race policy.

Definition 3 (GSPN, [1]) [5] *A GSPN is a 6-tuple GSPN=*(P,T,I^-,I^+,M_0,W) *where*

- *PN=*(P,T,I^-,I^+,M_0) *is the underlying place/transition net*

[5] For simplicity we omit the definition of inhibitor arcs and priority levels.

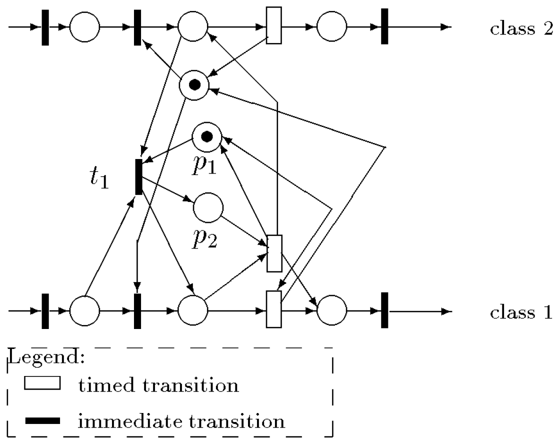

Figure 3: -/M/1-"preempt-resume priority" station modelled by a GSPN

- $W = (w_1, \ldots, w_{|T|})$[6] *is an array whose entry* w_i
  - *is the rate* $\in \mathbb{R}^+$ *of the negative exponential probability distribution function specifying the firing delay, if transition* t_i *is a timed transition or*
  - *is a weight* $\in \mathbb{R}^+$ *specifying the relative firing frequency, if transition* t_i *is an immediate one.*

The firing of enabled immediate transitions has priority on that of enabled timed transitions and happens immediately meaning in zero time.

The main advantages of timed and stochastic Petri nets are

- the possibility for qualitative analysis using efficient techniques from Petri net theory and
- the possibility for quantitative analysis.

The main disadvantage of these time augmented Petri nets is the very difficult description of scheduling strategies. Figure 3 shows a GSPN model of a station serving two classes of customers due to a "preempt-resume priority"[7] - strategy using standard Petri net elements. Transition t_1 is enabled if preemption will occur. Firing of t_1 preempts costumers of class 2. The marking of p_1 and p_2 keeps track of the occurrence of such a situation. This is the reason for modelling the service of a customer of class 1 by two identical parameterized timed transitions. Note that this realization of a preemptive priority scheduling strategy with resumption is only correct, if we assume negative exponentially distributed service times. For easier description of scheduling strategies additional elements are often integrated into the stochastic Petri net world, like e.g. inhibitor arcs, with the problem of raising the modelling power up to Turing machines even for the untimed model causing the undecidability of important analysis problems [18].

3.2 Extended Queueing Networks

There are only a few efforts undertaken in the past to add features for the description of qualitative aspects into queueing networks.

[6] $|T|$ denotes the number of elements in set T.

[7] A newly arriving customer with higher priority (class 1) preempts the service of the customer being served at that moment. The preempted customer will proceed his task at the interrupted point after completion of service of all customers with higher priority.

In extended queueing networks like e.g. RESQ ([20, 24]) some elements for the description of synchronization aspects are added to the queueing network world. Additional elements are passive resources, fork, join and split nodes (see figure 4). Each passive resource consists of a number of tokens representing the resource units available for customers arriving at the allocation node. If no token is available the customer is forced to wait, otherwise he passes the allocation node taking a free token. At the release node the token is given back to the passive resource, so that it is available for other customers arriving or being blocked at the allocation node. Figure 5 presents a central server model with memory constraints modelled by a passive resource (cf. [24]).

Further elements are fork and join nodes, where a customer creates several child processes resp. synchonizes with former generated processes. At split nodes child processes are created, but no synchronization takes place in the future.

The advantages of this form of extended queueing networks are similar to that of queueing networks:

- easy description of scheduling strategies,
- possibility for quantitative analysis using efficient algorithms from queueing network theory,
- possibility for description of some forms of synchronization aspects.

The main disadvantages of extended queueing networks are

- the lack of algorithms for qualitative analysis, although it looks easy to borrow them from Petri net theory,
- not all forms of parallel systems can be modelled. In [24] the reader will find an example demonstrating the difficulties in specifying the time-out behaviour of protocols using this model world.

The author has no knowledge of further directions in adding features to queueing networks, which have become as popular as RESQ. The reason is that most researchers are interested in new efficient techniques for performance evaluation and do not bother about expressive power. A combination of (generalized stochastic) Petri nets and queueing networks is given in [7], where one model world is used to calculate parameters for the other model world. So there is no combination on the description level.

To summarize this section we list the disadvantages of the above described model worlds:

Disadvantages:	
Timed and stochastic Petri nets	**Extended queueing networks**
difficult description of scheduling strategies	some parallel systems can not be described appropriately

The next section presents the QPN world, which combines the advantages of (timed and stochastic) Petri nets and queueing networks, so that these disadvantages disappear.

4 The QPN world

The main idea in creation of the QPN world [8, 9] was to add timing aspects to the places of a (coloured) Petri net (cf. figure 6). This idea reminds the reader of the TPPN

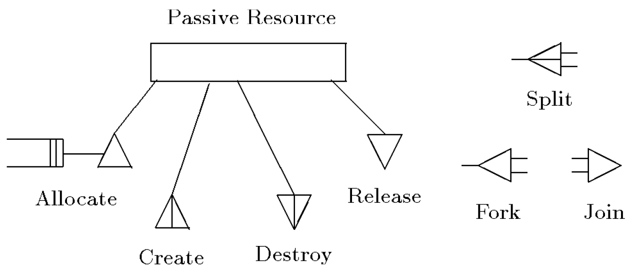

Figure 4: Primitives of the RESQ world

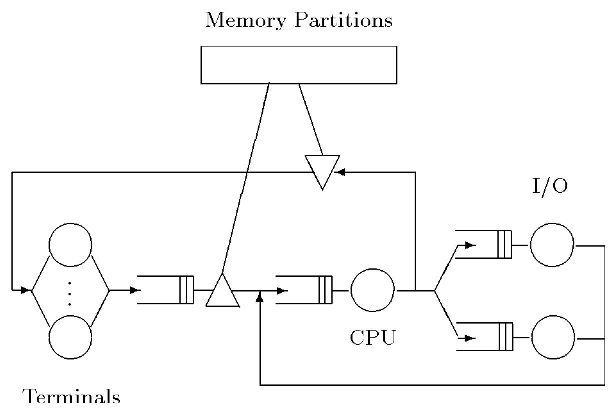

Figure 5: Central server system with memory constraints modelled by RESQ primitives

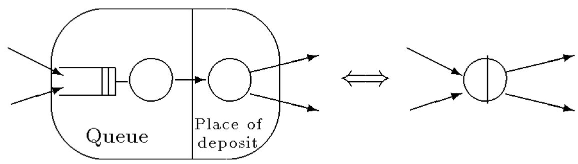

Figure 6: Timed place of a QPN

approach mentioned in section 3.1. In QPNs time is integrated in a more powerful way, because we don't restrict our model to the specification of a dwelling time for tokens. In QPNs a whole queue (station) may be integrated into the definition of a place. Such a timed place consists of two components, the queue (station) and a "place of deposit" for served tokens (customers)[8]. The behaviour of the net is like follows. Tokens, fired by the input transitions of such a timed place, are inserted into the queue due to the specified scheduling strategy. Tokens in a queue are not available for the transitions of the QPN. After completion of service the token (customer) is placed on the "place of deposit". Tokens on this "place" are available for all output transitions of the timed place. In an older version of the QPN world [8, 9] all transitions fired immediately after enabling. In the version described here we also integrate timed transitions to receive greater modelling convenience. Like in TTPNs an enabled timed transition will fire after a certain delay[9] specified by a random variable. Enabled immediate transitions will fire due to relative firing frequencies. We assume that no token is generated or destroyed in a queue, so that qualitative analysis can be partially done by analysing the underlying coloured Petri net.

Definition 4 (QPN) *A queueing Petri net (QPN) is a 8-tuple* $QPN=(P,T,C,I^-,I^+,M_0,Q,W)$ *where*

- $CPN=(P,T,C,I^-,I^+,M_0)$ *is the underlying coloured Petri net*
- $Q=(q_1,\dots,q_{|P|})$ *is an array whose entry* q_i
  - *denotes the description of a queue taking all colours of* $C(p)$ *into consideration, if* p_i *is a timed place or*
  - *equals the keyword* untimed, *if* p_i *is an untimed place.*
- $W=(w_1,\dots,w_{|T|})$ *is an array of functions whose entry* w_i *is defined on* $C(t_i)$ *and* $\forall\ c \in C(t_i): w_i(c)$ *is*
  - *the description of a probability distribution function specifying the firing delay, if transition* t_i *is a timed transition due to colour* $c \in C(t_i)$ *or*
  - *is a weight* $\in \mathbb{R}^+$ *specifying the relative firing frequency, if transition* t_i *is an immediate one due to colour* $c \in C(t_i)$.

The firing of immediate transitions has priority on that of timed transitions. The graphical representation of transitions and untimed places is similar to that of GSPNs and a pictorial representation of a timed place is given in the right part of figure 6.

5 Examples of QPNs

The QPN world is a very powerful, although easy to handle and understand, model world. Some examples will demonstrate its usefulness.

Figure 7 presents the central server model with memory constraints, which was modelled with RESQ primitives in figure 5:

QPN=(P,T,C,I^-,I^+,M_0,Q,W) where

- CPN=(P,T,C,I^-,I^+,M_0) is the underlying (coloured) Petri net[10] given in figure 7,

[8] Do not mix this notion of a place with the term "place" in Petri nets.
[9] We assume the race policy here.
[10] In this example we consider only one class of customers (colour of tokens).

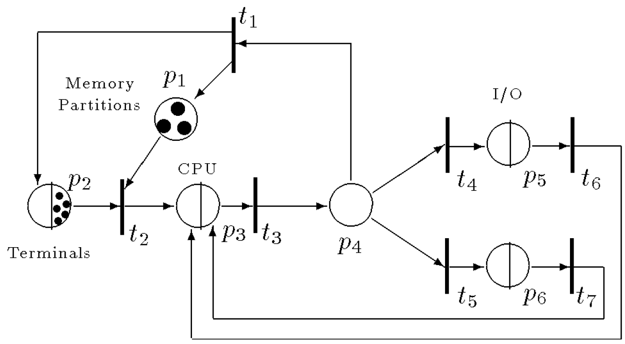

Figure 7: QPN model of the central server model

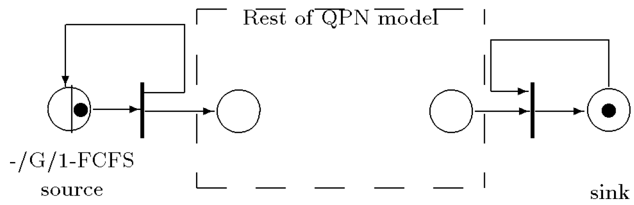

Figure 8: Open queueing network modelled by a QPN

- Q = (untimed, -/M/∞-IS, -/M/1-PS, untimed, -/M/1-FCFS, -/M/1-FCFS) [11],
- $W = (w_1, \ldots, w_{|T|})$ is the array of functions whose entry w_i is defined on $C(t_i)$ and (for simplicity we define) $\forall c \in C(t_i) : w_i(c) := 1$. Note that all transitions are immediate.

The structure of this model is distinctly described by the underlying (coloured) Petri net, so that qualitative analysis can be easily performed using techniques from Petri net theory. The place invariants of the Petri net e.g. prove that the number of tokens (customers) in the CPU and I/O stations is limited by the number of initially available memory partitions, thus giving the analyst more confidence in the "correctness" of his model.

The next example demonstrates that open queueing networks with generally distributed interarrival times can be modelled by QPNs (figure 8).

Furthermore time-outs, which aren't possible to model using RESQ primitives ([24]) can be easily described with QPNs (figure 9). After a certain time the host deposits a message in the buffer with probability w_2 or will proceed his local task with probability w_1. The sender takes the message from the buffer and transmits it to the receiver. t_3 represents the timer, t_4 and t_5 represent the reception of an incorrect resp. correct acknowledgement.

6 Expressive Power

A key issue for new model worlds is their expressive power in comparison to other model worlds. Looking at the examples in section 5 one might suppose that the QPN world is more prowerful than most of the other model worlds. In this section we show that the QPN world is a superset of (coloured) Petri nets, queueing networks and timed and stochastic Petri nets.

[11] IS (infinite server) assumes that every arriving customer will be immediately served with full server capacity, so the waiting area of a station with such a scheduling strategy is always empty. PS (processor sharing) is a special case of the RR strategy. Here all customers are served simultaneously with $\frac{1}{number\_of\_customers\_actually\_present}$ of the servers capacity.

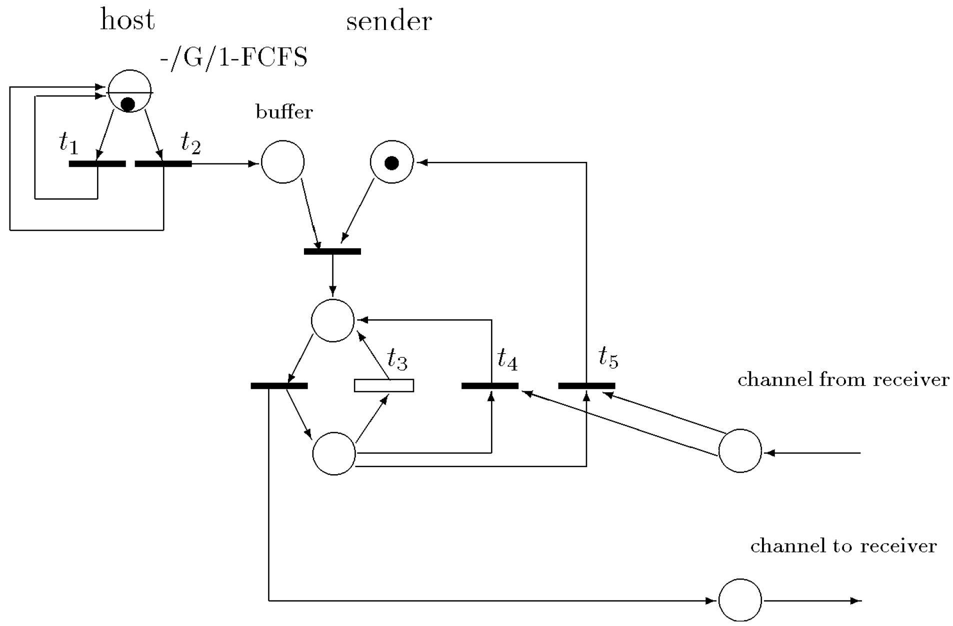

Figure 9: Time-out modelling with QPNs

- QPNs ⊃ CPNs
  It is obvious that every CPN=(P,T,C,I^-, I^+, M_0) can be represented by a QPN=(P,T,C,I^-, I^+, M_0,Q,W) where Q=(untimed, ... ,untimed) and all transitions are immediate.

- QPNs ⊃ queueing networks
  Each queueing network can be represented by a QPN using the following construction procedure:

  - For each station i of the queueing network create a uniquely determined timed place p_i and an entry q_i with the description of the queue for the array Q.
  - For each pair (i, j) of interconnected queues, i.e. ∃ classes s,t: $r_{i,s;j,t} > 0$, create a transition $t_{(i,j)}$ of the QPN.
  - Define C: $P \cup T \rightarrow$ *set_of_classes*, so that for each class c of customers of the queueing network
    $$C(x) := \begin{cases} \{t \mid \exists \text{ queue } i, \text{class } s : r_{i,s;j,t} > 0\} & \text{if } x = p_j \in P \\ \{(s,t) \mid r_{i,s;j,t} > 0\} & \text{if } x = t_{(i,j)} \in T \end{cases}$$
  - the incidence functions I^- and I^+ for a pair $(p_i, t_{(j,k)})$ are defined as
    $$I^-(p_i, t_{(j,k)})((s,t))(c) := \begin{cases} 1 & \text{if i = j and s = c} \\ 0 & \text{otherwise} \end{cases}$$
    $$I^+(p_i, t_{(j,k)})((s,t))(c) := \begin{cases} 1 & \text{if i = k and t = c} \\ 0 & \text{otherwise} \end{cases}$$
    $\forall\, c \in C(p_i), (s,t) \in C(t_{(j,k)})$.

– the initial marking is given by the number of customers in closed chains:

$$M_0(p_i)(c) := \begin{cases} k & \text{if } k \text{ is the number of customers in chain } h \ni c \text{ with} \\ & c \in C(p_i) \text{ and } \forall \text{ queues j}, c' \in C(p_j): \\ & c' \in h, M_0(p_j)(c') \neq 0 \Longrightarrow (i = j \text{ and } c = c') \\ 0 & \text{otherwise} \end{cases}$$ [12]

– Q := $(q_1, \ldots, q_{|P|})$

– all transitions are immediate and the array W is given by W := $(w_{(1,1)}, w_{(1,2)}, \ldots, w_{(1,|P|)}, \ldots, w_{(|P|,|P|)})$ where $w_{(i,j)}$ is defined as $w_{(i,j)}((s,t)) := r_{i,s;j,t}\ \forall (s,t) \in C(t_{(i,j)})$.

The source and sinks of an open chain can be modelled by the constructs given in figure 8.

- QPNs $\supset$ timed and stochastic Petri nets

There are a variety of timed and stochastic Petri nets which can be modelled by QPNs. First of all we will show that GSPNs $\subset$ QPNs by construction:

– The QPN's underlying CPN is directly defined by the underlying place/transition net of the GSPN with one colour for each place and transition (cf. definition 2).

– Q := (untimed, ... , untimed).

– Define the matrix W of the QPN using matrix W of the GSPN by $w_i^{QPN}(c) := w_i^{GSPN}$ for the uniquely determined colour $c \in C(t_i)$.

Here is a list of other timed and stochastic Petri nets representable by QPNs:

* SPNs [17], because SPNs $\subset$ GSPNs[13].

* Sifakis' timed Petri nets [21] belong to the TPPN class with deterministic dwelling times for tokens on places. Such a timed Petri net can be modelled by a QPN as follows.

  The QPN's underlying CPN is directly given by the place/transition net of the timed Petri net. All places of the QPN are timed with a IS scheduling strategy and deterministic service times given by the deterministic dwelling times of the timed Petri net. All transitions are immediate with weight 1.

* The stochastic Petri nets in [25] are similar to the timed Petri nets of Sifakis. The only difference is that deterministic dwelling times are exchanged by exponentially distributed times. If we use these service time specification for the queues of timed places, the same construction procedure is applicable.

* Extended stochastic Petri nets (ESPNs) and deterministic stochastic Petri nets (DSPNs) [17] can be represented in a similar way like GSPNs, if we assume that all additional Petri net elements integrated into these worlds, e.g. inhibitor arcs, are not taken into consideration. In both variants of stochastic Petri nets timed and immediate transitions are defined. In ESPNs firing delays of timed transitions may be described by general pdfs and in DSPNs all timed transitions have deterministic or exponentially distributed firing delays.

[12]Note that this recursive definition does not yield a uniquely determined initial marking for a given queueing network, but this has no effect on the overall behaviour of the net.

[13]In SPNs all transitions are timed with exponentially distributed firing delays.

* Also some timed and stochastic Petri nets of the preselection class can be represented by QPNs by modelling the preselection phase with immediate transitions (cf. [4]), e.g. Zuberek's stochastic Petri nets [26].

- QPNs and extended queueing networks
  Because of QPNs $\supset$ queueing networks, we have to prove that the additional elements of figure 4 can be represented by appropriate QPN elements. It is obvious that fork, join and split nodes can be directly represented by Petri net elements of the QPN. The only critical RESQ primitive is the passive resource. Release, creation and destruction of resource units are representable by appropriate transitions of the QPN. Unfortunately the allocation node is not directly representable, because it acts like a timeless (!) queue with a user defined scheduling strategy [20]. The FCFS strategy e.g. can't be modelled by standard Petri net elements [14]. So QPNs can only simulate allocation nodes with a random access strategy, which are also representable by standard Petri net places.

7 Conclusions

We have shown that the QPN world is a very powerful model world due to expressive power. The convenient notation is excellently suitable for Petri net as well as queueing network analysts. The modelling process is supported in the following way. First of all the designer can describe the logical structure of the system using standard Petri net elements. After verifying certain qualitative features, timing aspects can be integrated in a very convenient way for performance evaluation. The main advantage of the QPN world is its ease in describing scheduling strategies.

Furthermore the QPN world can be used as a starting point for the development of a reasonable proceeding for combining qualitative and quantitative analysis (cf. [8, 9]) exploiting efficient algorithms developed in the last decades for the other model worlds.

References

[1] M. Ajmone-Marsan. *Stochastic Petri nets: an elementary introduction.* Advances in Petri nets. G. Rozenberg (ed.), Lecture Notes in Computer Science 424, 1989, pp. 1-29.

[2] M. Ajmone-Marsan, G. Balbo, G. Conte. *A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems.* ACM Transactions on Computer Systems, Vol. 2, No. 2, Mai 1984, pp. 93-122.

[3] M. Ajmone-Marsan, G. Balbo, G. Conte. *Performance models of multiprocessor systems.* MIT Press, London (Great Britain), 1986.

[4] M. Ajmone-Marsan, G. Balbo, A. Bobbio, G. Chiola, A. Cumani. *The effect of execution policies in the semantics and analysis of stochastic Petri nets.* IEEE Transactions on Software Engineering, Vol. 15, No. 7, 1989, pp. 832-846.

[5] M. Ajmone-Marsan, G.Balbo, G. Chiola, G. Conte. *Applicability of stochastic Petri nets to performance modeling.* Second International Workshop on Applied Mathematics and Performance Reliability. Models of Computer Communication Systems, Italien, may 1987, pp. 517-534.

[6] *Petri Nets. Central Models and Their Properties. Advances in Petri Nets.* Proceedings of an Advanced Course, Bad Honnef (FRG), Sept. 1986, Part I, G. Rozenberg (ed.), Lecture Notes in Computer Science 254, 1986.

[7] G. Balbo, S.C. Bruell, S. Ghanta. *Combining queueing networks and generalized stochastic Petri nets for the solution of complex models of system behaviour.* IEEE Transactions on Computers, Vol. 37, No. 10, Oct. 1988, pp. 1251-1268.

[8] F. Bause. *Funktional und quantitativ analysierbare Rechensystemmodelle; über Verbindungen von Petrinetz- und Warteschlangen-Modellwelten.* Thesis for diploma, Fachbereich Informatik der Universität Dortmund (FRG), 1986.

[9] F. Bause, H. Beilner. *Eine Modellwelt zur Integration von Warteschlangen- und Petri-Netz-Modellen.* Proceedings of the 5. GI/ITG-Fachtagung "Messung, Modellierung und Bewertung von Rechensystemen und Netzen", Braunschweig (FRG), 1989, pp. 190-204.

[10] G. Berthelot. *Transformations and the decompositions of nets.* in [6], pp. 359-376.

[11] G. Chiola, C. Dutheillet, G. Franceschinis, S. Haddad. *On well-formed coloured nets and their symbolic reachability graph.* Proceedings of the International Conference on Application and Theory of Petri Nets, Paris (France), June 1990, pp. 387-410.

[12] E. Cinlair. *Introduction to stochastic processes.* Englewood Cliffs, New Jersey (USA), 1975.

[13] J.M. Couvreur. *The general computations of flows for coloured nets.* Proceedings of the International Conference on Application and Theory of Petri Nets, Paris (France), June 1990, pp. 204-223.

[14] A. Finkel, L. Rosier. *A survey on the decidability questions for classes of fifo nets.* Advances in Petri Nets, Lecture Notes in Computer Science 340, 1988, pp. 106-132.

[15] S. Haddad. *Generalization of reduction theory to coloured nets.* Proceedings of the 9th European Workshop on Application and Theory of Petri Nets, Venedig (Italy), June 1988.

[16] K. Jensen. *Coloured Petri nets.* in [6], pp. 248-299.

[17] A. Pagnoni. *Stochastic nets and performance evaluation.* in [6], pp. 460-478.

[18] J.L. Peterson. *Petri net theory and the modeling of systems.* Englewood Cliffs, 1981.

[19] C. Ramchandani. *Analysis of asynchronous concurrent systems by timed Petri nets.* Ph. D. thesis, MIT, 1973.

[20] C.H. Sauer, E.A. MacNair, J.F. Kurose. *The research queueing package: past, present, and future.* Proceedings of the National Computer Conference, AFIPS Vol. 51, 1982, pp. 273-280.

[21] J. Sifakis. *Use of Petri nets for performance evaluation.* Measuring, Modelling and Evaluating Computer Systems, H. Beilner, E. Gelenbe (ed.), 1977, pp. 75-93.

[22] E. de Souza e Silva, R.R. Muntz. *Queueing networks: solutions and applications.* Stochastic analysis of computer and communication systems, Hideaki Takagi (ed.), North-Holland, 1990, pp. 319-399.

[23] M. Silva, R. Valette. *Petri nets and flexible manufacturing.* Advances in Petri nets, Lecture Notes in Computer Science 424, 1989, pp. 374-417.

[24] M. Vernon, J. Zahorjan, E.D. Lazowska. *A comparison of performance Petri nets and queueing network models.* Proceedings on the International Workshop on Modelling Techniques and Performance Evaluation, Paris (France), 1987, pp. 191-202.

[25] C.Y. Wong, T.S. Dillon, K.E. Forward. *Timed places Petri nets with stochastic representation of place time.* Proceedings of the International Workshop on Timed Petri Nets, Turin (Italy), 1985, pp. 96-103.

[26] W.M. Zuberek. *Performance evaluation using extended Petri nets*, Proceedings of the International Workshop on Timed Petri Nets, Turin (Italy), 1985, pp. 272-278.