Computing First Passage Time Distributions in Stochastic Well-Formed Nets

Gianfranco Balbo, Marco Beccuti, Massimiliano De Pierro
Dipartimento di Informatica Università di Torino
{balbo,beccuti,depierro}@di.unito.it

Giuliana Franceschinis
Dipartimento di Informatica
Università del Piemonte Orientale
giuliana.franceschinis@di.unipmn.it

ABSTRACT

The increasing demand for customer centric evaluation of systems, mostly related with the assessment of the quality of service that they can deliver, requires the development of techniques properly designed to model and to study the movement of specific entities generically referred to as "customers". Stochastic Well-Formed Net (SWN) are naturally suited for the representation of systems in which "customers" of different categories compete for the use of common resources. Color classes of SWN are easily associated with these different categories, leaving to the peculiar features of the formalism the possibility of exploiting all the symmetries existing into the representation for the efficient and effective computation of the measures of interest. Within this application context, the computation of first passage time distribution measures in SWN is becoming of primary interest. Customers however are not primitive entities in the formalism and an approach similar to that previously developed for Generalized Stochastic Petri Nets (GSPN) is suggested to overcome this problem in which P-semiflows are used to identify the customers. In this paper we propose an original algorithm for computing some P-semiflows of colored PNs in parametric form by exploiting the peculiarities of the objective of this investigation, and extend the customer centric first passage time computation approach previously developed for GSPNs, to make it suitable for SWN models. Moreover, the paper proposes an enhancement of the SWN notation in order to provide a way to ease the modeler in the specification of customer scheduling policies that may affect the computation of first passage time distributions. This extension, inspired by Queueing Petri Nets, adds to SWN some "syntactic sugar" that allows to include in the model queueing places which are automatically replaced by appropriate submodels, before solving the model.

Categories and Subject Descriptors

I.6.4 [**Computing Methodologies**]: Simulation and modeling—*Model Validation and Analysis*; G.3 [**Mathematics of Computing**]: Probability and statistics—*Queueing theory; Markov processes; Stochastic processes*

General Terms

Performance, Theory

Keywords

High Level Petri Nets, Colored place invariants, First passage time, Symbolic Reachability Graph

ICPE'11, March 14–16, 2011, Karlsruhe, Germany.

1. INTRODUCTION

The SWN [9] high level Petri net formalism is an extension of GSPNs [1] with colors. Similarly to Colored Petri Nets [14], SWNs provide a compact and parametric model representation that, thanks to a well-structured color syntax, allows an efficient solution technique called Symbolic Reachability Graph (SRG). The SRG method automatically exploits the model symmetries leading to a reduced state space and a corresponding lumped Continuous Time Markov Chain (CTMC).

One type of performance index which is not straightforward to define on SWN models, is the distribution of the time required for a (specific) token to traverse a subnet. The difficulty is more related to the precise definition of the problem and of the choice of the appropriate abstraction level, rather than to the computation of this index, since first passage time analysis tools presented in the literature [7, 12] can be applied to the underlying CTMC.

This problem has been tackled in the context of GSPNs in [13] first, and then in [4]. The main differences between these two approaches are: (a) the specification of the tagged token and of its paths within the net, (b) the way to achieve the appropriate state description level and (c) the definition of the performance index (purely state based in the former work, event based in the latter).

In the SWN context the approaches devised for GSPNs in [4, 13] are not directly applicable, hence in this paper we propose a refinement of the technique developed for the specification of the tagged token in GSPNs in order to adapt it to the SWN, exploiting its efficient analysis techniques.

The method proposed in this paper is based on the assumption that "customers" are associated with sets of colors of the model and uniquely identified by their colors. Such unique identifier might be useful for several modeling features: e.g. to differentiate the conducts of different classes of customers in certain phases of their behaviors, to correctly model fork-join situations, to differentiate the delays

based on the color of the involved tokens. The presence of a unique color identifier for each customer makes it possible to directly follow the behavior of a specific *customer token* for first passage time computation purposes without the need to "tag" it, however it might also quickly lead to a dramatic growth of the state space size. The SRG approach overcomes this last problem, automatically exploiting symmetries in the behavior of SWN models, and mitigates the state space explosion problem while still providing the ability to track one specific token (equivalent to the *tagged token* in GSPNs).

This new approach still consists of three steps, as that in [4], but it has been redefined to work with the SWN formalism. The three steps can be summarized as follows: (1) identification of a set of colored tokens (i.e. customers) circulating in the net, satisfying certain conservation properties and behavior similarities; (2) identification of a subnet, where the customers may pass through, so that the measure of interest can be defined as the distribution of time spent by a token of a certain color (a customer) at each passage through the subnet; (3) translation of this high level specification into a CTMC that exploits the aggregation properties deriving from the construction of the SRG, to which classical MC first passage time computation techniques can be applied.

Furthermore, in our approach SWNs are extended with some "syntactic sugar" in order to provide a way to ease the modeler in the specification of customer queueing policies which, as pointed out in literature [3], may affect the computation of the first passage time distribution in a substantial manner. This aspect is illustrated through some experiments.

The paper is organized as follows: Sec. 2 introduces the SWN formalisms intuitively through a model of an Emergency Department. Sec. 3 defines how the customers can be identified on an SWN model with the help of structural property considerations. Sec. 4 introduces the method proposed to specify the measures of interest at the net level; Sec. 5 shows how to efficiently compute these measures using the SRG to derive the underlying CTMC. Sec. 6 introduces our extension of the SWN formalism to ease the modeler in the specification of customer queueing policies. Sec. 7 presents some numerical results. Sec. 8 draws some conclusions and presents directions for future work.

2. STOCHASTIC WELL-FORMED NETS

In this section the SWN formalism is described in an intuitive manner using the net in Fig. 1 and assuming that the reader has some basic knowledge of Petri Nets (PN) [1] and Colored Petri Nets (CPN) [14].

Running example: a hospital model.

We illustrate our approach through the analysis of patient waiting times in the model of a hospital Emergency Department (ED) shown in Fig 1, that has been developed using the description provided by [16]. In this model we classify the ED patients according to the following three categories: patients requiring resuscitation (high priority), patients with major illnesses or injuries (medium priority) and patients with minor illnesses or injuries (low priority).

Place *Healthy* contains all the healthy patients; while place *Ill* all the ill patients heading towards the hospital. An healthy patient that falls ill is represented by the firing of (an instance of) transition *FallIll*. When an ill patient reaches the hospital (place *Assessment*) its status is immediately evaluated (transitions *HighPrio*, *MediumPrio* and *LowPrio*). If its priority is high then s/he is moved to the resuscitation room (place *ResuscRoom*) and waits for being stabilized: the stabilization process can start (transition *BtoStabilize*) iff there is at least one trauma team available (place *TraumaTeam* marked). A stabilized patient is moved to the monitored room (place *MonitoredRoom*) by the firing of transition *EtoStabilize*. In the same room all patients with medium priority are also admitted (transition *MediumPrio*). A patient is constantly monitored until the results of his/her blood and X-ray exams become available (transitions *EBloodExam* and *EX-Ray*); then s/he is treated or operated by a doctor according to the outcomes of his/her tests (transitions *ToDoctorM* and *ToSurgery*). Obviously the medical examination can start if there is at least one doctor available (place *Doctors* marked), while the surgical operation can start if there is at least one doctor and one operating room (place *OperatingRoom* marked) available. Instead, patients with minor illnesses or injuries have to wait in the waiting room (place *WaitingRoom*) until a doctor is available and no more patients with higher priority need to be treated or operated (inhibitor arc from place *ReadyT* to transition *ToDoctorL*).

Finally, transitions *DischargeL*, *DischargeM* and *DischargeRec* model the discharge of a patient from the hospital.

For the moment the reader can ignore the new graphical notation for place *WaitingRoom* (a circle with a vertical bar) and consider it as a normal place. In Sec. 6 the meaning of this different representation will be discussed in details.

SWN informal definition.

A formal definition of the SWN formalism can be found in [9]; what follows is an informal presentation. In an SWN, as in any colored net formalism, a *color domain* ($cd()$) is associated with places and transitions. The color domain of a place defines the possible colors of the tokens that it contains, whereas the color domain of a transition defines its possible *firing instances*. The enabling conditions and the state change associated with each transition firing instance are specified through functions associated with arcs: given the color identifying an instance of the transition connected to the arc, the function provides the (multi)set of colored tokens that will be added to or removed from the place connected to the arc. The initial marking assigns a multi-set of colored tokens to each place.

Color domains in SWNs are expressed as Cartesian products of *color classes* (and $\mathcal{C} = \{C_1, \ldots, C_n\}$ is the set of all classes), which may be seen as primitive domains and may be partitioned into *static subclasses*[1]. The colors of a class represent entities of the same nature (e.g. patients in an hospital) whereas the colors inside a static subclass have also the same potential behavior (e.g. patients with major illness); the actual behavior depends also on the distribution of tokens of the same subclass in the initial marking.

In the net of Fig. 1 there is a single color class P, modeling patients, partitioned into three static subclasses $P = P_H \cup P_M \cup P_L$: each static subclass identifies a different level of urgency of treatment (and hence of priority) of the

[1]In the special case where it is not necessary to partition a color class into static subclasses we use the same name for the color class and its unique static subclass.

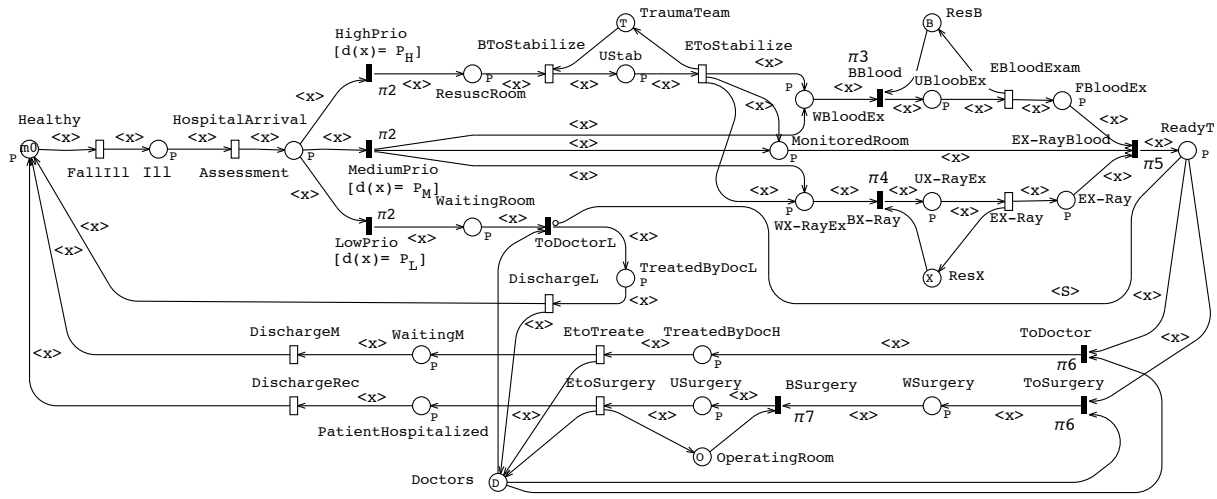


Figure 1: SWN model of patients flow in a hospital Emergency Department.

corresponding patients. The color domains of the places in this net are either P or ε the neutral domain consisting of a single color (i.e. the *black tokens* of ordinary nets). For instance, the color domain of place *TraumaTeam* is ε, while that of place *Healthy* is P.

If we are also interested in distinguishing between high specialized physicians and not specialized ones, then we have to introduce a new color class D partitioned into two static subclasses D_S (i.e. specialized doctors) and D_G (i.e. generic doctors) and to update consistently the color domain of the net places. In particular, the color domain of place *Doctors* becomes D, while that of *TreatedByDocL*, *TreatedByDocH*, *WSurgery* and *USurgery* becomes $P \times D$.

The transition color domains, are defined through a list of typed variables, whose types are selected within the color classes (for brevity, a transition color domain may be denoted with the Cartesian product of the variable types, in some conventional order). The variables of a transition appear in the functions annotating its arcs and can be interpreted as function parameters; a transition instance binds each parameter to a specific color of proper type (sometimes denoted with the tuple of the colors assigned to the variables in some conventional order, for brevity). Restrictions can be defined on the allowed instances of a transition by means of a *guard* which is a Boolean expression defined on the transition color domain expressed through a *standard predicate*. The terms of a standard predicate are *basic predicates*, which allow to compare colors assigned to variables of the same type (denoted $x = y$), or to test whether a color element belongs to a given static subclass (denoted $d(x) = C_{i,j}$), or to compare the static subclasses of the colors assigned to two variables of the same type (denoted $d(x) = d(y)$). For instance, the color domain of transitions *HospitalArrival* and *HighPrio* is defined as $x : P$; the guard $d(x) = P_H$ means that the transition instances to be considered are only those binding variable x to a color in P_H. If physician types were also included in the model then we could specify a guard $(d(y) = D_S)$ for transition $ToSurgery$ assuring that only specialized physician can operate a patient.

Arc functions are expressed by properly combining a set of predefined *basic functions*, whose domains and codomains are color classes. The allowed basic functions are the projection, the diffusion/synchronization, and the successor function[2] (only for ordered color classes); a linear combination of basic functions is also a basic function. The most recurrent type of basic function is the projection, which selects one item out of a tuple corresponding to a transition instance and is denoted with a variable name (e.g., x). In the net of Fig. 1, only two functions are used: the projection (x) and the synchronization (S_C, which is a constant function evaluating to the whole set of colors in class C). For instance, projection function $\langle x \rangle$ appearing on the arc connecting place *Ill* and transition *HospitalArrival* selects, from a given instance of the transition, the color element (of class P) bound to variable x meaning that a given instance of *HospitalArrival* characterized by the binding $x = c$ (denoted for brevity $\langle HospitalArrival : c \rangle$) is enabled iff a token of color c is presented in place *Ill* (the token will be consumed when firing that transition instance). A key property of the SWN syntax is that the color dependence of the possible behaviors represented in a model is limited by the types of arc functions and transition guards. This means that it is only possible to specify different behaviors for colors belonging to different static subclasses, or based on the equality or diversity of color elements involved in a given transition instance (e.g. synchronization of tokens with the same color - independently of the specific color). This is essentially the "symmetry" property that can be exploited in the analysis, and that will be used in the next sections.

The specification of the stochastic behavior is given by associating (integer) priorities and (real) weights with the transitions. Transitions with priority zero, called *timed transitions*, fire after a random delay, with a negative exponential distribution; transitions with priority π greater than zero are called *immediate transitions* and fire in zero time. The weight of a timed transition is interpreted as the rate of the corresponding distribution, while that of an immediate transition allows to compute a probability distribution, to be used when the transition firing involves some conflict res-

[2]The GreatSPN tool provides also the predecessor function which is not in the original SWN definition.

olution. Transition weights can depend on the color instance of a transition only in a limited (symmetric) manner.

Modeling complex systems with SWNs is more convenient than modeling them with GSPNs because of their compactness and readability as well as for their significantly higher degree of parametrization that can be exploited at the analysis level. Indeed, the constraints on the syntax of SWNs allow the automatic exploitation of the behavioral symmetries of the model and provide the possibility of performing the state-space based analysis on the more compact SRG. The SRG construction relies on the *symbolic marking* concept, namely a compact representation for a set of equivalent ordinary markings. A symbolic marking is a symbolic representation, where the actual color of tokens is abstracted away, but the ability to distinguish tokens with different colors and to establish their static subclass is retained. Tokens appearing in the same set of places with the same multiplicity and *belonging to the same static subclass* are grouped into so-called *dynamic subclasses*. An example of symbolic marking for the SWN Hospital model in Fig. 1 with $|P_H| = |P_M| = 2$ and $|P_L| = 5$ is $\hat{m} = Ill(1\langle Z_L^2\rangle)Healthy(1\langle Z_H^1\rangle 1\langle Z_M^1\rangle 1\langle Z_L^1\rangle)$ $TraumaTeam(1)ResB(2)ResX(2)Doctors(2)$, $|Z_H^1| = 2$, $|Z_M^1| = 2$, $|Z_L^1| = 4$, $|Z_L^2| = 1$ where all the patients of type P_H are grouped into the dynamic subclass Z_H^1, all the patients of type P_M into Z_M^1 and all the patients of type P_L into dynamic subclasses Z_L^1 and Z_L^2 with cardinalities 1 and 4, respectively. This symbolic marking represents 5 ordinary markings where only one patient of type P_L is ill, while all the other patients are healthy and one trauma team, two blood exam teams, two X-ray exam teams and two doctors are available.

Starting from an initial symbolic marking, the SRG can be constructed automatically using a symbolic firing rule. Observe that in this work we consider only initial symbolic markings where all the colors belonging to the same static subclasses are in the same places with the same multiplicity: hence it is possible to specify the initial symbolic marking as the Cartesian product of static subclasses. For instance, the symbolic marking $\hat{m}_0 = Healthy\ (1\langle Z_H^1\rangle 1\langle Z_M^1\rangle 1\langle Z_L^1\rangle)\ Trauma-Team\ (1)ResB(2)ResX(2)Doctors(2)$, $|Z_H^1| = 2$, $|Z_M^1| = 2$ $|Z_L^1| = 5$ satisfies this condition.

Most qualitative properties of the model can be analyzed on the SRG, moreover a lumped MC can be automatically obtained from the SRG, to compute the same class of performance indices that might be computed on the Reachability Graph.

Some analysis algorithms supporting the verification of qualitative as well as quantitative properties, may require to translate the SWN model into an equivalent GSPN. This is always possible by means of an *unfolding* procedure, which consists in replicating places and transitions as many times as the cardinalities of the corresponding color domains; the replicas (called instances) are denoted $\langle p : c\rangle$ and $\langle t : ..., x_i = c_i, ...\rangle$ (considering only transition instances that satisfy the transition guard); moreover, if an arc connecting t and p exists and is annotated with function f, then $\forall c' \in f(c)$ an arc connecting $\langle p : c'\rangle$ and $\langle t : c\rangle$ appears in the unfolding with weight $f(c)(c')$ (multiplicity of c' in multiset $f(c)$).

An example of structural analysis results which might be of interest in this context is the set of minimal p-semiflows, which are useful to establish marking invariance properties of a PN model, from which it is possible to deduce, e.g., the model boundedness or mutual exclusion between place markings or transition enabling. A p-semiflow is a function associating a non negative integer weight with each place of the model. The weighted sum of tokens in places is invariant in any marking reachable from the initial one, for this reason often the term place invariant (or p-invariant) is used to refer both to the p-semiflow, and to the marking invariant properties it implies. In the context of this paper we are interested in p-semiflows computed on the unfolding of a SWN model; the following formal sum notation is used to express such p-semiflows:

$$\sum_{i,d \in cd(p_i)} \lambda_{i,d}.\langle p_i : d\rangle \tag{1}$$

where $\lambda_{i,d}$ are the non negative integer weights.

Algorithms exist to compute a generating family of minimal p-semiflows from the structure of a PN. The computation of similar properties for colored PNs (in particular for SWN) in parametric and symbolic form instead is not feasible in general, except for certain subclasses of models [11].

In this paper we introduce a new method to compute a particular kind of place invariants. Such invariants may be deduced from the p-semiflows of the unfolded net expressed by formula (1), however the proposed method computes them directly without explicitly deriving the p-semiflows of the unfolded model.

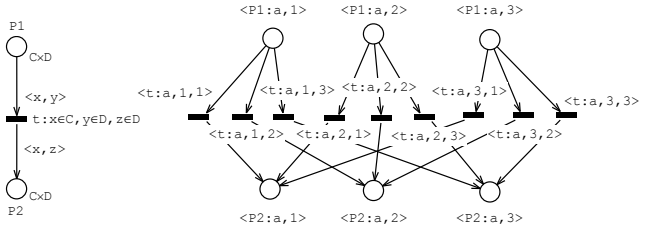


Figure 2: An example of unfolding.

Let us introduce an example of p-semiflow and place invariant on the simple net of Fig. 2. The SWN in Fig. 2.(a) has a single transition t with color domain $x : C, y : D, z : D$. Let us assume $C = \{a, b\}$ and $D = \{1, 2, 3\}$. The unfolding of this SWN model yields two disconnected subnets: one of them is depicted in Fig. 2.(b), and refers to place and transition instances characterized by C element a. The second is equal up to substitution of all occurrences of a with b. Two minimal (and similar) p-semiflows cover the unfolded net: the first one is $\langle P1 : a, 1\rangle + \langle P1 : a, 2\rangle + \langle P1 : a, 3\rangle + +\langle P2 : a, 1\rangle + \langle P2 : a, 2\rangle + \langle P2 : a, 3\rangle$ and covers all the place instances characterized by color a in the element corresponding to color class C; the second is equal up to a substitution of all occurrences of a with b.

The place (marking) invariant for the SWN that can be deduced from the first p-semiflow related with color a in C may be informally expressed as follows: the number of tokens (in $P1$ and $P2$) that have color component a is invariant in each marking reachable from the initial marking. The invariant implied by the second semiflow is similar, but refers to tokens that have color component b.

It would be convenient to define a compact and parametric expression for such "colored" invariants, which could be obtained by performing a "projection" of place color domains (and hence of the place markings) on their C component (assuming that there is only one occurrence of C in their

color domains). For instance, the invariance property on tokens with color component a and b stated above could be expressed as follows:

$$\forall c \in C, \forall m \in RS : \Psi_C(m(P1))(c) + \Psi_C(m(P2))(c) = K$$

where RS is the set of markings reachable from m_0, and $\Psi_C : Bag(cd(p)) \rightarrow Bag(C)$ denotes the projection of the marking of place p on color class C, appearing once in $cd(p)$.

3. MODELING CUSTOMERS IN SWN

The definition of passage time measures is often associated with the abstract notion of customers using resources or requiring service in shared service centers. While customers are a natural notion in Queueing Networks (QNs) they are not such in PN formalisms. One possibility is to interpret some set of tokens in a PN model as the set of customers circulating in the modeled system, however this makes sense only if some constraints are satisfied by the model. Firing a PN transition corresponds to consuming tokens from its input places and producing tokens in its output places: in general the consumed and produced tokens are not related; in other words a transition does not move tokens from input to output places. However, as already pointed out in [4], a relation between consumed and produced tokens may be derived by a proper interpretation of the model elements: this interpretation allows to model customers as tokens.

In the context of SWNs, in order to establish a relationship among tokens consumed and produced by transitions, we make the following two assumptions:

1. *Customers are mapped on the different colors of a given static color subclass.* If $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ is the set of customers that we want to represent, a color class C_i is specified in the SWN model so that a bijective assignment exists among the element of the set Γ and the colors in a static subclass $C_{i,j}$ of this C_i. Hereafter in the discussion this static subclass will be named $CLIENTS$, when we want to distinguish it from the other classes. if $CLIENTS = \{c_1, c_2, c_3\}$ then in the model there will be three customers respectively identified by colors c_1, c_2 and c_3. Some constraints are imposed on it: the model must behave homogeneously w.r.t. all elements of $CLIENTS$, which is guaranteed by the SWN syntax if it is defined as a static color class; moreover class C_i including $CLIENTS$, cannot occur more than once in the color domain of the places and transitions.

2. *Tokens of the same color consumed and produced by a transition are related*: if a transition firing consumes from place p_i a token with color component c_1 and produces in place p_j a token with color component c_1 then we assume that the firing of such a transition corresponds to moving customer c_1 from place p_i to place p_j. Moreover, we assume that a transition cannot generate tokens with color component c_i if it has not consumed tokens with this component: the interpretation being that the transitions may change the state of a customer, but it may not make a customer disappear, or create one customer out of the blue.

At this point of the discussion two remarks are due:
• the first concerns the fact that during the evolution of a SWN from its initial marking, colors are often combined yielding colored tokens with complex color schemes where several color classes appear in the domain: for example a model can represent with a token colored $\langle c_i, r_j \rangle$, with $c_i \in CLIENTS$ and $r_j \in RES$, a customer c_i that acquired a resource r_j. In this case we talk about the *color component* c_i of a colored token. Sometimes, for the sake of clarity and to simplify the discussion, referring to tokens related to customers we may omit this fact, focusing just on the component related to the $CLIENTS$ class and say that "a token has color c" meaning that "the token has a component of color c";
• the second concerns the number of tokens of the same color that can be present in a marking of a SWN. For instance, let us consider a model where the customers $\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ are processes that at some point in time fork into several threads which successively join. A SWN will describe such fork and join structure specifying the process identities with colors in $CLIENTS = \{c_1, c_2, \ldots, c_n\}$ and modeling, during the system evolution, the several threads of generic process γ_i with several tokens having color component c_i. Thus in some system state after the fork takes place there will be several tokens carrying the identity of customer c_i in several places, according to the progression of the threads. After the join operation all the tokens having color c_i will synchronize and only one such token will remain. This simple example shows that in general there can be several places containing tokens with color component c_i: their marking represents the state of γ_i.

As it happens in the fork and join example, it is possible in general to observe that at any instant (at any system state) a place invariant is satisfied by the colored tokens representing a customer γ_i. Such invariant states a conservation of the tokens representing the customers[3].

The property of conservation of the tokens related to the customers can be studied at structural level computing the p-semiflows of a particular net which can be derived from the original one. Moreover, the information on p-semiflows of the derived net also provides a mean to identify the portion of SWN visited by a customer during its evolution.

For the purpose of this paper we restrict our analysis to SWNs in which the place invariants derived for the customers satisfy the following law: if in the initial marking exactly one token in the places of the invariant has color component c_i (for any i), then in any reached marking there exists exactly one token with color component c_i in the same places. This law is satisfied if the weights of the p-semiflows of the derived net are either 0 or 1. Such requirement simplifies the specification of the passage time measure because the local state of the generic customer γ_i is uniquely identified by the position (the place) of the related colored token c_i in the places of the invariant.

Hereafter, we use the notation $\langle s : d \rangle$, where $s \in P$ or $s \in T$, to indicate color instance d of a transition or place s with $d \in cd(s)$.

3.1 P-semiflows involving the customers

As mentioned earlier, the computation of a generative family of p-semiflows of SWN models in parametric form is not feasible in general; on the other hand, the complexity of

[3]The conservation property is not strictly necessary in general for the computation of the first passage time, but the approach proposed in this paper only considers models where customers are *conserved*.

computing it on the unfolded model could be too high, and provide an unmanageable result. However, for our purposes we need not to compute a generative family of minimal p-semiflows, instead we are interested only in those semiflows involving place instances having a specific color and satisfying some additional restrictions: these places, together with the connected transitions, identify the path that a given customer may follow. Any p-semiflow covering place instances not including that specific color is not interesting in this context.

The computation of p-semiflows useful for our kind of analysis can be effectively performed by deriving a GSPN from the SWN, by means of an operation of projection and partial unfolding. Intuitively, the projection selects a portion of the SWN unfolding by focusing on an arbitrary customer, say c, belonging to $CLIENTS$. The resulting net, which is no longer colored, comprises the paths (i.e. the submodel) that a customer c may potentially follow in the SWN. Let us introduce such projection operation formally; the notation used hereafter is summarized in Table 1.

Let $\mathcal{N}$ be a SWN. Let $C_{i,j}$ be a static subclass satisfying the constraints made on $CLIENTS$ and $c \in C_{i,j}$. Then $\mathcal{N}' = \Pi(\mathcal{N}, c)$, the projection of $\mathcal{N}$ on color c, is a GSPN obtained as described next.

Let call Π the projection operator. The projection is defined in order to contain representatives of all the *instances* of those transitions and of those places of the SWN that are related with tokens having color component c. If a transition instance of $\mathcal{N}$ consumes or produces[4] a token with color component c, then it is *represented* in $\mathcal{N}'$ by a corresponding transition; moreover, the projection operation exploits the structural symmetries encoded in the definition of the functions that label the arcs of a SWN in order to reduce the size of the resulting net, in fact more transition instances can be represented by a single transition in the projection. The formal definition of the sets of places and transitions P' and T' and of the arc functions W'^* of $\mathcal{N}'$ is the following:

Definition of set P': $P' = \{\langle p : c\rangle : p \in P, e_i(p) \neq 0\}$. In words, the set P' contains one place, labelled $\langle p : c\rangle$, for each colored place p of the SWN that has C_i in the color domain.

Definition of set T' and arc function W': Let $t \in T$. In order to simplify the notation let us rearrange the Cartesian product expression for the color domain of t such that the class C_i appears in the first position, that is $cd(t) = C_i \times D$ (for simplicity we consider only one class D besides C_i in the Cartesian product, but it could be safely replaced with a Cartesian product of color classes, all different from C_i). Let c' be an arbitrarily fixed color in $C_i \backslash \{c\}$. Let us consider colors $\langle c, d\rangle$ and $\langle c', d'\rangle$ where $d \in D$, $d' \in D$. Let $a = |W^*(t,p)(\langle c, d\rangle)|_c$ and $b = |W^*(t,p)(\langle c', d'\rangle)|_c \ \forall d, d' \in D$ (see Tab. 1 for the definition of $|\mathbf{A}|_c$); due to the SWN definition and since we restrict to arc functions without guards, a and b respectively depend from c and c' only and not from d and d'.

If t is such that it is connected to a place p with C_i in the color domain, then T' and W'^* are defined as follows.
$\forall p \in P : e_i(p) \neq 0$, $\forall t \in T : W^*(t,p) \neq \emptyset$:

1. if $e_i(t) \neq 0$
   - if $a \neq 0$ and $\exists d \in D : \mathcal{G}(t)(\langle c, d\rangle) = true$ then
     $\langle t : c\rangle \in T'$ and $W'^*\big(\langle t : c\rangle, \langle p : c\rangle\big) = a$
     if $b \neq 0$ and $\exists d \in D : \mathcal{G}(t)(\langle c', d\rangle) = true$ then
     $\langle t : c'\rangle \in T'$ and $W'^*\big(\langle t : c'\rangle, \langle p : c\rangle\big) = b$
2. if $e_i(t) = 0$
   - $\langle t : \bullet\rangle \in T'$
     $W'^*\big(\langle t : \bullet\rangle, \langle p : c\rangle\big) = |W(t,p)(d)|_c$ where $d \in cd(t)$ and $\bullet$ represents a neutral color.

The net resulting from the projection is a GSPN, which represents the portion of the SWN unfolding where the tokens representing c may circulate. Any p-semiflow involving places of this net induces a place invariant for c in the SWN. Moreover, since c is an arbitrary customer in $CLIENTS$, a parametric colored p-semiflow could be derived directly.

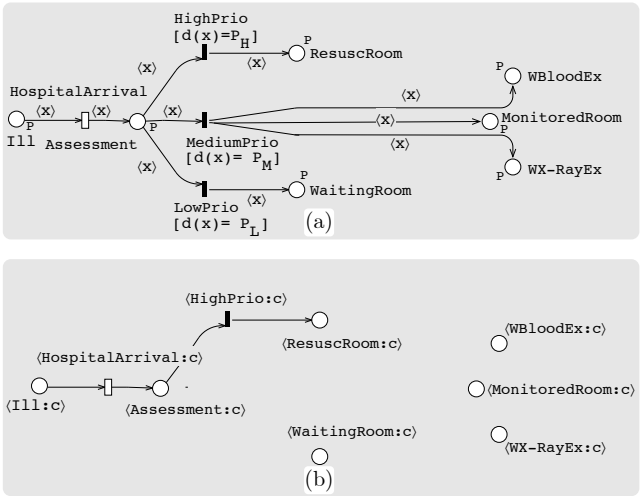


Figure 3: A portion of the hospital model projected on color $c \in P_H$.

Examples

The remainder of this section illustrates the application of the projection operator on four examples.

The first example shows the operator applied to a portion of the hospital model introduced in Sec. 2. Consider the subnet illustrated in Fig. 3.(a). The projection is done onto an arbitrary fixed color c belonging to static subclass P_H, which hence is the $CLIENTS$ class used in the discussion before. The result is depicted in Fig. 3.(b). Let us illustrate the derivation in detail. In the subnet that we are considering, all places have color domain P. Thus the result of the projection will be a model (Fig. 3.(b)) in which all the places contain an instance of color c. All the arc functions have the same form $\langle x\rangle$: when applied to an arbitrary color $d \in P$, they return the multiset $\langle d\rangle$ which is one token with a single color component equal to d. Considering the transition *HospitalArrival* only the instance $\langle \textit{HospitalArrival} : c\rangle$ is represented in the projection (rule 1 of the Π operator and case $a \neq 0$). Of the many colored transitions that *HighPrio* represents in the SWN, only the instance $\langle \textit{HighPrio} : c\rangle$ is considered in the projection; in fact, applying rule 1, case $b = 0$ of Π, we observe that for each $c' \in P\backslash\{c\}$, instance $\langle \textit{HighPrio} : c'\rangle$ consumes from place *Assessment* the multiset $\langle c'\rangle$ which is obviously irrelevant for our purposes since it

[4]Inhibitor conditions are not considered because they do not affect the p-semiflow computation.

$cd(s)$	is a function which assigns to each $s \in T \cup P$ a color domain such that $cd(s) = C_1^{e_1(s)} \times C_2^{e_2(s)} \times .... \times C_n^{e_n(s)}$, where the superscript $e_i(s)$ is defined below.
$e_i(s)$	returns a value $\in \mathbb{N}$ which denotes the number of occurrences of C_i in $cd(s)$.
$\mathcal{G}(t)$	it is the guard function of transition t. The guard function applied to a color instance of t: $\mathcal{G}(t)(d) \in \{true, false\}$, $\forall d \in cd(t)$.
$\langle \cdot, c, \cdot \rangle$	indicates any colored token containing the color c among its components.
$\lvert\mathbf{A}\rvert_c$, $c \in C_i$ where $\mathbf{A} \in Bag(C_1^{e_1}, \ldots, C_i, \ldots, C_n^{e_n})$,	is defined as the number of tuples in multiset $\mathbf{A}$ having color component $c \in C_i$; and $Bag(C)$ denotes the set of all multisets that may be built on set C (a multiset is a generalization of a set, that can contain several occurrences of the same element).
$W^{+/-}$	assigns to each pair $(t,p) \in T \times P$ a function $W^{+/-}(t,p) : cd(t) \rightarrow Bag[cd(p)]$ so that $W^{+/-}(t,p)$ maps each color of transition t into a multiset of tokens on place p and $W^{+/-}(t,p)(c)$ denotes the application of the function to color c of t. Notation W^* is used to indicate both W^+ and W^-

Table 1: A summary of the notation used in definition of projection operator.

does not involve the manipulation of tokens with color component c. For $c \in P_H$, (restriction due to the guard associated with transition *HighPrio*), the instance $\langle HighPrio : c \rangle$ consumes from place *Assessment* the multiset $\langle c \rangle$ that is composed exactly of one token with color component c. No instance of transition *MediumPrio* is represented in the projection because guard $[d(x) = P_M](\langle c \rangle)$ is false when evaluated with respect to color $c \in P_H$; moreover, for all c' satisfying the guard, and for all connected places $b = |\langle x \rangle(c')|_c = 0$ (rule 1, of the projection definition). Similar arguments hold for *LowPrio*.

As a second example we show the application of the operator Π focusing on a single transition t. Consider the nets depicted in Fig. 4.(a) and 4.(b). In the SWN of Fig. 4.(a), t has color domain $D \times C$; place $P1$ has color domain D while places $P2$ and $P3$ have color domain $D \times C$. The projection is performed on color $c \in C$ (in this example C is a class including a unique static subclass). Place $P1$ does not belong to the GSPN because it can not contain color c on which we are projecting the SWN. For the sake of clarity we depicted the deleted subnet with a dotted line in the projected net (GSPN of Fig. 4.(a)). The transition instance $\langle t : \cdot, c \rangle$ (the dot means "any" color) in the SWN consumes from place $P2$ 2 tokens whose second color component is c. This value is computed by analyzing the functions composing the arc function (when $x = c$). With regard to $P3$, $\langle t : \cdot, c \rangle$ does not produce any token whose second color component is c (when $x = c$, function $S - x$ evaluates to the set of all colors in C except c). When we consider any arbitrary instance $\langle t : \cdot, c' \rangle$ such that $c' \neq c$, the arc function $\langle y, S + x \rangle$ evaluated in c' is a multiset that contains a single token whose second component has color c, hence arc $P2 \rightarrow \langle t : c \rangle$ has multiplicity 1. Finally, $\langle 2y + S, S - x \rangle$ when evaluated for $\langle \cdot, c' \rangle$ has several tokens whose second component is c: $S - x$ maps to $C \backslash \{c'\}$ thus it contains one c; $2y + S$ maps to $2\langle \cdot \rangle + D$, hence $|2y + S| = |D| + 2$.

Fig. 4.(b) shows the case where a colored transition t does not contain C in its color domain, but the function on the arcs connecting places $p2$ and $p3$, includes a basic function which is a constant on class C (S in the example). In this case only a single instance denoted $\langle t : \bullet \rangle$ is included in the projection on c, and since $|S|_c = 1$ the weights of the arcs from $\langle p2, c \rangle$ and to $\langle p3, c \rangle$ are both 1.

In the third example we consider a simple SWN, depicted in Fig. 5.(a), that shows an interesting case where a transition embeds several instances that route the tokens with

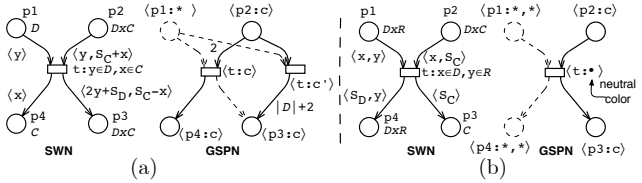


Figure 4: The projection on c: two examples.

color c in different manners. The color domain is C for all nodes. Let us verify if there exist some p-semiflows for colors in C and if they are valid for our analysis of first passage time. Fig. 5.(b) shows the projection on color c. We observe that similar nets can be obtained for any of the other colors: all these nets are structurally identical. Thus what we say for customer c is actually valid for all customers (so that we obtain a result which is parametric). The computation of the p-semiflows of the SWN in Fig. 5.(a) that involve client c can be done using standard algorithms for GSPN onto the projected net of Fig. 5.(b). The net of Fig. 5.(b) has one minimal p-semiflow covering all places. Finally we show the

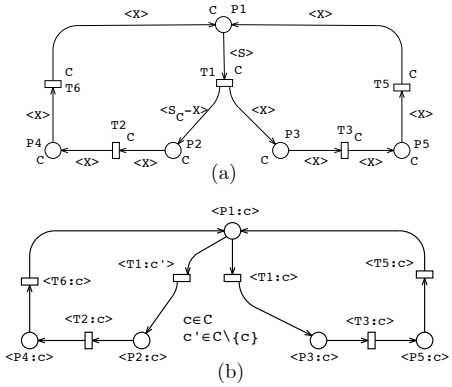


Figure 5: (a) An SWN; (b) its projection on c.

application of the method to the complete model of the hospital, analyzing customers modeled by static subclass P_H. First, we project the SWN onto an arbitrary color in P_H, obtaining the GSPN depicted in Fig. 6.(a). This GSPN has three minimal p-semiflows which have very similar support since they share a large set of places and are distinct only

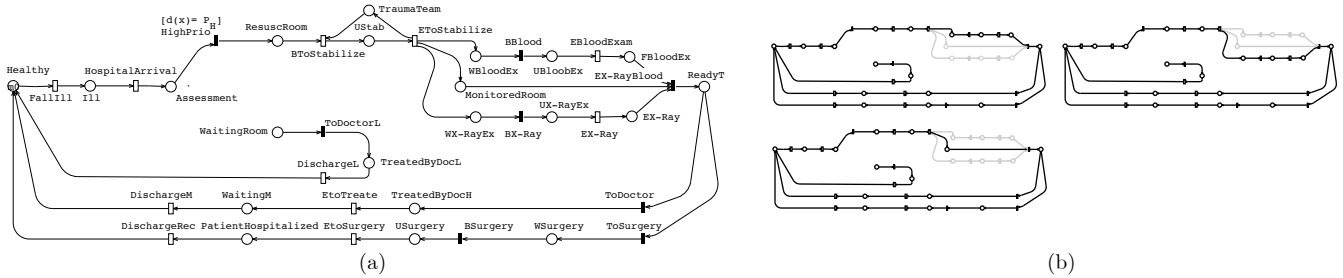


Figure 6: Hospital model: (a) Projection onto a color in static subclass P_H; (b) p-semiflows for a P_H customer.

with respect to places corresponding to the fork-and-join located at the top-right part of this model. Fig. 6.(b) highlights the three subnets induced by the three p-semiflows with the only purpose of providing a way of noticing these similarities and differences at a first glance.

4. SPECIFICATION OF FIRST PASSAGE TIME MEASURES

First passage time measures can be specified at the net-level (i.e., directly on the model) in a manner similar to that proposed for GSPNs in [4]; however since each transition in an SWN model usually embeds several instances, the specification must refer to such instances, taking into account the peculiar role of the colors used to identify customers.

The first passage time at the SWN level corresponds to the time required for a token with color component c (arbitrarily chosen in $CLIENTS$) representing a given customer, to traverse a portion of the net. Such subnet is identified specifying entry and exit points corresponding to specific *transition instances* which represent events involving customer c (e.g., admission and discharge of a patient). It is often useful to specify additional conditions on the system state at the instant of the firing of an entry transition instance (e.g., admission of a patient when all doctors are busy).

The p-semiflow analysis proposed in the previous section, allows to identify one or more subnets, denoted $\mathcal{N}'_{\mathcal{P}}$ (where $\mathcal{N}' = \Pi(\mathcal{N}, c)$) in which the customers can flow while keeping their distinct identity; hence the candidate entry and exit points can be more easily identified on these subnets rather than on the complete SWN model.
The formal definition of $\mathcal{N}'_{\mathcal{P}}$ is as follows: let $\mathcal{P} \subseteq P'$ be the set of places of $\mathcal{N}'$ covered by a p-semiflow; let $\mathcal{T} = \{t \in T' : \exists p \in \mathcal{P}$ such that $(t \in p^{\bullet} \vee t \in {}^{\bullet}p)\}$; then $\mathcal{N}'_{\mathcal{P}}$ is the subnet of $\mathcal{N}'$ whose nodes are the places in $\mathcal{P}$ and the transitions in $\mathcal{T}$. For instance, the SWN of the hospital has three possible different subnets $\mathcal{N}'_{\mathcal{P}}$ suited for first passage time analysis of patients belonging to P_H, as depicted in Fig. 6.(b).

Since the $\mathcal{N}'_{\mathcal{P}}$ subnet derives from a partially unfolded and projected net, in general it may include two representatives of the instances of the same transition in the original model: this helps the modeler to see more clearly the actual distinct paths in the net involving customer c thus making the task of selecting the appropriate entry and exit points easier. The SWN example in Fig. 5.(a) highlights such a situation: looking at the projection on c in Fig. 5.(b) one can clearly see that instance $\langle T1 : c\rangle$ routes customer c on the right part of the net while instance $\langle T1 : c'\rangle$ routes customer c on the left part of the net. In the SWN model these choices at $T1$ are encoded in the arc functions and are less directly visible.

On such net the modeler could specify as entry and exit points respectively $\langle T1 : c\rangle$ and $\langle T5 : c\rangle$: the measure obtained is the time a customer c takes to traverse the subnet $T1 \rightarrow P3 \rightarrow T3 \rightarrow P5 \rightarrow T5$, starting with the event corresponding to the firing of transition instance $\langle T1 : c\rangle$.

Summarizing, entry, exit and forbidden points may be conveniently selected by using the subnet $\mathcal{N}'_{\mathcal{P}}$ as a sort of *filter* for an easier interpretation of the original net. Once the transitions of interest have been identified, the set of instances to be selected may be automatically derived, assuming that any instance involving c must be selected, or explicitly specified using the guard's syntax: for instance one may be interested in measuring the time required for a patient to undergo a given treatment, assuming that upon entering the subnet representing the treatment (through entry transition t) the patient is assigned to a specialized physician, in this case a constraint on the static subclass of the variable (say y) of t representing the person who is in charge of the patient may be expressed as a guard (e.g. $d(y) = D_S$). Finally, the specification of conditions on the state before and/or after the firing of a given entry transition instance should be given in a form that does not break the symmetry properties of the model, and such that it can be verified on a symbolic marking. Examples of conditions that satisfy this constraint are those depending on the number of tokens in places independently of their color, or the number of tokens whose color components belong to certain static subclasses.

The specification of entry, exit, and forbidden points that must be provided by the modeler can be formalized as follows: let $\mathcal{T}_{in}, \mathcal{T}_{out}$ and $\mathcal{T}_{forbid}$ be three disjoint subsets of $\mathcal{T}$; for each transition t' in these sets the modeler must provide

- a guard $\mathcal{G}^m$ (possibly defined as the constant true) consistent with the color domain of transition $t \in T$ of the original SWN model $\mathcal{N}$ from which t' was generated; this guard must not contain any term referring to the variable of type C_i in $cd(t)$;
- two conditions $\xi_t pre$ and $\xi_t post$ that can be checked on symbolic markings.

The following section describes in details how this specification is used to define the *start*, *end*, and $forbidden$ states, required for passage time computation. The intuitive idea is that whenever a transition instance $\langle t : d\rangle$, which is represented by some $t' \in \mathcal{T}_{in}$ and satisfies $\mathcal{G}^m(t)(d)$, is fired within a sequence of one timed and zero or more immediate transition instances connecting two tangible (symbolic) markings $\hat{m}$ and $\hat{m}'$, satisfying $\xi_t pre$ and $\xi_t post$ respectively,

then $\hat{m}'$ is considered as a start state. Similarly, end states may be characterized by the fact that they are reachable through the firing of a sequence containing a transition instance $\langle t : d \rangle$, represented by some $t' \in \mathcal{T}_{out}$, which satisfies $\mathcal{G}^m(t)(d)$, and satisfies both the pre and post conditions as well. When choosing the transitions in the set $\mathcal{T}_{in}$ and $\mathcal{T}_{out}$, some structural conditions should be respected, guaranteeing the presence of a path in the net connecting each transition in $\mathcal{T}_{in}$ to some transitions in $\mathcal{T}_{out}$: this can be checked on the projected and partially unfolded net $\mathcal{N}'$ thus allowing to identify the subnet traversed by customer c subject to measure.

In the literature, different approaches are proposed to specify the passage time measure of interest. A purely state space based approach for "customer centric" measures in GSPN models is discussed in [13], our proposal instead is event oriented, but it also allows to define conditions on the state: the two methods were compared extensively in [4].

A different approach has been presented in [10] for stochastic process algebra models; it is based on *probes* and it has been implemented in the ipc (Imperial PEPA compiler) tool [8] and the ipclib library. The proposed technique exploits in a quite elegant way the compositional nature of process algebras. The basic idea consists in introducing special processes called *probes*, that monitor the system (or one specific component in the system) to witness specific events (actions) that trigger the start/stop of a time measure. The start and stop criteria specification for probes is expressed by means of regular expressions that are then translated into automata: this allows to specify quite complex start/stop conditions, based on the history of observed actions as well as the system state.

The firing of entry and exit transitions in our proposal correspond to the probes start/stop events and the state dependency allowed in the probes specification corresponds to our conditions on the state upon occurrence of an entry-exit transition firing, however probes are more powerful since they may base the start (and stop) of the observation on complex action sequences rather than single events. Of course this may produce an increase in the state space size, which is acceptable as far as the additional complexity is the minimal required to obtain the measure of interest. Another important difference is due to the fact that compositionality is not part of the PN formalism (although several proposals exist in the literature to introduce compositionality a-posteriori), nor the concept of "process" that can stem as an interpretation of the model, but is not a native concept in PNs. Actually the idea of identifying customers, of requiring their conservation, and the identification of subnets where the customers may flow and in which are conserved, can be interpreted as a mechanism to retrieve processes from the flat structure of the net. Finally, in [10] there is no particular emphasis on the notion of "customer" as proposed in this work as well as in [13]; nevertheless the possibility of specifying local probes seems to be a suitable mechanism to define "customer centric" measures (provided that the behavior of individual customers is represented by processes in the system model).

5. COMPUTING THE FIRST PASSAGE TIMES DISTRIBUTION

The computation of the first passage time distribution requires to isolate one of the customers and to follow its movements within the net. Once the static subclass C corresponding to the set of customers under observation has been chosen, (any) one among the colors in C must be isolated: this is performed by splitting C into two subclasses, the former containing the identity of the "tagged" customer, the latter containing the identities of all the other customers. The splitting may require some adjustments in the net structure. Let us denote with C^t and C^u the two new static subclasses obtained by splitting C (with $|C^t| = 1$) and $|C^u| = |C| - 1$, the following substitutions are required in the model guards and arc functions:

substitution of terms involving C in guards

- $(d(x) = C) \rightarrow (d(x) = C^t) \vee (d(x) = C^u)$
- $(d(x) = d(y)) \rightarrow$
  $\big((d(x) = d(y)\big) \vee \big(d(x) = C^t\big) \wedge \big(d(y) = C^u\big) \vee \big(d(x) = C^t \ \wedge \ d(y) = C^u\big)\big)$

substitution of S_C in arc functions

- $S_C \rightarrow S_{C^t} + S_{C^u}$

After this transformation the SRG can be computed: thanks to the splitting of the static subclass of interest, the tagged customer can be easily followed because it is the only color contained into static subclass C^t. The identity of the elements in C^u is automatically handled in the most efficient way by the SRG algorithm. A lumped CTMC is then generated from the SRG.

The computation of the index of interest also requires to identify the start and end states on the CTMC underlying the SWN model (and isomorphic to its SRG). When several start states exist an initial distribution on them must also be specified: in case the CTMC has a steady state solution, then the steady state probabilities of the start states can be used as initial distribution. Start states are selected by inspecting the SRG looking for any tangible state m' with the following properties: in marking m', the token with color in C^t is in a place belonging to the monitored subnet; m' is reachable from a tangible state m, where the token with color C^t was outside the monitored subnet; moreover m must reach m' through the firing of a transition sequence containing one timed and possibly some immediate transition firing instances, embedding an *entry point* transition instance. Similarly for the end states (markings with C^t outside the subnet, reached from a marking where C^t was inside the subnet, through a path including an exit point).

The SRG properties can be used to optimize the steady state solution phase: in fact, under certain conditions, it is possible to compute the probability distribution of the states of the SWN *before* refining it: since it has been proved that the ordinary states in a given symbolic marking are equiprobable, and since (using combinatorial arguments) it is possible to compute the number of ordinary states contained in each symbolic marking, then it is possible to obtain the steady state distribution of the refined SRG from that of the SRG computed using the more abstract model. The correspondence between the abstract and refined states is straightforward. A further optimization, concerning the actual first passage time computation phase, could be obtained by avoiding to build the whole refined SRG, but rather deriving the refined version of the states on the paths from the start to the end states from the most abstract states containing them: the technique used to build the Extended SRG [6] could be applied to this purpose.

With respect to the method proposed in [4], the approach

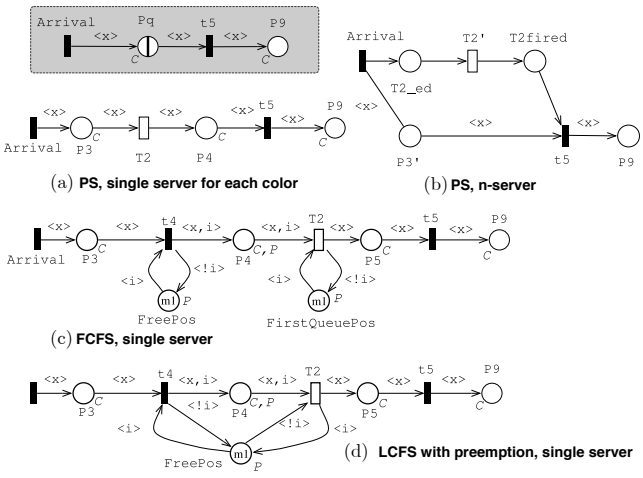


Figure 7: Some examples of SWN templates for a "queue place"

presented in this paper does not require any unfolding of the model: the right level of detail of the state is automatically generated by the SRG algorithm. With respect to the method proposed in [13], the colors here are used in a different manner: in our proposal each customer has a unique identity (color), while in [13] only two colors are used, one for the tagged customer and the other for the remaining customers; despite this increased detail level, thanks to the SRG technique, there is no loss of efficiency.

6. SOME SUBTLE MODELING ISSUES

The interpretation of some tokens of an SWN as customers leads naturally to identifying the timed transitions that involve customers as service centers: if the resources available to perform the service are limited then the model should specify both the queueing policy used to define the service order of the customers requiring service, the number of available servers and their service rate (in this context the service time distribution is negative exponential).

In [4] the authors show that in GSPN models the customers queueing policy may influence the passage time distribution, while it is irrelevant for the computation of average performance indices (due to the memoryless property of the exponential distribution and to the fact that tokens in GSPNs cannot be distinguished). In the context of the SWN formalism, since tokens can be distinguished through colors, the queueing policy may influence also the average performance indices[5] [3]. Although the SWN is powerful enough to directly model any queueing policy, the addition of some "syntactic sugar" may help the modeler in correctly specifying the order of extraction of customers from places as well as various service specifications.

Here, we propose an extension of the SWN formalism introducing a special place, called "queue place" (drawn as a circle with a bar inside) which embeds into a single place the queue, the service and an output buffer (called depository) as already proposed in the QPN formalism [5]. Moreover, the "queue places", as in [15] where an extension of QPN is proposed, can also model specific orders of extraction of tokens from a place without any service.

During the solution process these special places are automatically replaced by a corresponding subnet that is derived instantiating an appropriate version of the several templates that have been developed for representing the different queueing policies: Processor Sharing (PS), First Come First Served (FCFS), Last Come First Served (LCFS), etc. The obtained model corresponds to a standard SWN model on which the SRG approach can be applied.

Fig. 7 shows some examples of SWN templates for a generic queue place: they are described in details hereafter.

In Fig. 7.(a) transition `T2` with single server semantics may be interpreted as a service center and place `P3` as the queue in front of it. Since the color domain of `T2` is $x : C$, there are actually $|C|$ instances of `T2`, corresponding to the possible values that can be assigned to variable x. If the tokens in `P3` have all different colors (which is true if the tokens represent customers) then there will be as many *enabled concurrent instances* of `T2` as the number of different color tokens in `P3`: this correctly models a single server *for each color* (i.e. a delay), while it is not suitable to represent a service center with n servers shared by all the customers (colored tokens queueing for service in the input place). If instead we want to model a PS queueing discipline and n servers [6], the model of Fig. 7.(b) must be used, where transition `T2`$'$ has a unique instance and has a n servers semantics. Note that in GSPN models it is sufficient to assign a single server semantics to a timed transition to obtain the desired behavior, while in the context of SWNs a model transformation is needed.

In Fig. 7.(c) it is shown how a single server FCFS queue could be explicitly modeled in an SWN. In the explicit representation an ordered color domain, `P`, is introduced modeling the available positions in the queue: the ordering is circular, so that the SWN model may be seen as a circular array implementation of the queue: places `FirstQueuePos` and `FreePos` correspond to two pointers to the head and to first free position in the queue respectively, when the queue is empty these places are marked with tokens of the same color. Upon a new arrival the `FreePos` is updated to the successor ($\langle !i \rangle$), while `FirstQueuePos` remains unchanged. The transition `T2` representing the service is enabled only for the instance involving the customer in the first position of the queue; when it fires, `FirstQueuePos` is updated to point to the next position in the queue.

Observe that the same submodel could be used also to model a specific order of extraction of token for a queue place without service, by simply changing timed transition `T2` with an immediate one. An example of this type of queue place for the hospital Emergency Department in Fig. 1 is the queue place *WaitingRoom*.

Finally, Fig. 7.(d) shows an SWN template for a single server LCFS-PR queue. This model is derived by the previous SWN model removing the place `FirstQueuePos` and adding two arcs, with labels $\langle !i \rangle$ and $\langle i \rangle$, connecting place `FreePos` to transition `T2`, and vice-versa. In this case the transition `T2` is enabled only for the instance involving the customer for which the next queue position is free; when

[5]In completely symmetric models where the color does not influence choices or the possibility to perform a synchronization, insensitivity of the average performance indices to the order in which tokens are extracted can be proven.

[6]Exploiting the property of the exponential distribution it is modeled as an exponentially distributed service time followed by a random choice among the customers present in the queue at the end of service.

it fires, `FreePos` is updated to point to the queue position previously occupied by the customer.

7. EXPERIMENTAL RESULTS

In this section some experimental results are presented, which have been computed with a prototype framework based on GreatSPN [2] for the generation of the CTMC, and on Hydra [12] for the computation of the first passage time distribution. GreatSPN is also used to construct the sets of start, end, and forbidden states, while the computation of the (steady state) initial distribution of the start states can be performed either by using GreatSPN or Hydra.

Two measures have been computed for the running example of Fig 1: the former measures the time from the entrance of a P_H patient through entry point *HospitalArrival* up to exit points *DischargeM* and *DischargeRec* [7]; the latter measures the time from the entrance of a P_L patient through entry point *HospitalArrival* up to exit point *DischargeL*. Two scheduling policies are considered for the "queue place" *WaitingRoom*: FIFO and RO.

Tagging a "customer" causes an increase in the state space size: in Tab. 2 the number of symbolic states of the plain ED model is compared with that where a P_H or P_L patient is tagged. These experiments are performed for different number of patients ($|P_H|, |P_M|, |P_L|$) fixing the number of trauma teams T=1, doctors D=2, blood analysis teams B=2, X-ray analysis teams X=2 and operating rooms O=1. The first column shows the values of the parameters used for the experiments. The second and third columns report the RG size[8] and the untagged model SRG size. The last two columns show the SRG size of the tagged models. From the second column it is clear that the RG approach becomes quickly intractable due to the state space explosion. The SRG approach mitigates this effect exhibiting a high level of aggregation (e.g. in case 2,3,6 the SRG size is ≈ 590 times smaller than that of the RG). The SRG size of the tagged model increases by a factor that depends on both the cardinality of the $CLIENTS$ static subclass and the complexity of the subnet where these tokens can circulate. An upper bound on the SRG size reduction w.r.t. the RG size is given by $\prod_{C_s} |C_s|!$ on all static subclasses C_s used in the model. When separating the tagged customer, the $CLIENTS$ subclass is partitioned into two subclasses, of cardinality 1 and $|CLIENTS|-1$, hence the reduction upper bound decreases of a factor $|CLIENTS|$ (in other words the SRG of the tagged model may increase of a factor $|CLIENTS|$). In practice, the degree of reduction depends on how much the colored tokens are dispersed over the places of the model (more reduction is achieved when the tokens tend to be distributed in different places); this effect can be observed on Tab. 2: when the tagged customer is in $|P_H|$ the SRG size of the tagged model increases of $1.91, 2.76$ and 3.55 when P_H is $2, 3$ and 4 respectively, while when tagging P_L the size increases of $2.10, 2.42, 2.65$ and 2.83 when $|P_L|$ is $3, 4, 5$ and 6, respectively (indeed, there are several places where the patients in P_H may be located, while there are only few places where the patients in P_L may stay).

Modeling the FIFO policy (for customers in subclass P_L) instead of RO for place $WaitingRoom$ causes the state space size to increase; for instance, in case 2,2,6 |SRG| increases by a factor of ≈ 1.24, and |RG| by a factor of ≈ 5.53.

In Fig. 8.(a) the probability density function (pdf) for the first measure is shown where the transition weights are defined as in Tab. 3 and $|P_H| = 2$, $|P_M| = 2$, $|P_L| = 6$. The pdf allows to state properties as e.g. "90% of P_H patients leave the hospital within 450 time units from admission".

Finally, Fig. 8.(b) plots the pdf of the time taken for a P_L patient to move from admission (transition *HospitalArrival*) to discharge (transition *DischargeL*) assuming different scheduling policies on the queue place *WaitingRoom*. As expected, the two probability density functions are different (and the difference increases as the number of patients increases), while they have the same first moment (170.6785 time units), that coincides with the average time required for a P_L patient to traverse the same subnet, computed from the steady state distribution of the states in the untagged model (applying Little's formula).

8. CONCLUSION

In this paper we have proposed a way for specifying on SWN models a class of performance measures that involve the identification of one specific colored token representing one among a set of customers uniquely identified and constantly present in the model. The measure to be computed is the time required for the "tagged" token to traverse a given subnet: in particular we are interested in computing the pdf of such time, which can be obtained by applying first passage time computation techniques on the underlying (lumped) CTMC.

This work improves the methods developed in the context of GSPNs [4, 13] to SWNs, thus exploiting the efficient analysis techniques based on SRG, which mitigate the problem that affects all the approaches based on state space generation and thus characterized by an exponential growth of their complexity. The new approach is described in details and thanks to a prototype implementation in the GreatSPN framework interfaced with Hydra, some experimental results have been computed.

A proposal of adding some "syntactic sugar" to SWNs to ease the modeler task when representing "customer centric" SWN models is also discussed: the extension is inspired by the QPN formalism [5, 15].

The proposed technique could be exploited also in the computation of similar measures for GSPNs by replacing the *unfolding procedure* of [4] with a "coloring procedure" to be used in conjunction with the SRG: the correct handling of the transitions service semantics and of queue-places would benefit from the SWN extensions discussed in Sec.6.

9. REFERENCES

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. J. Wiley, New York, NY, USA, 1995.

[2] S. Baarir, M. Beccuti, D. Cerotti, M. De Pierro, S. Donatelli, and G. Franceschinis. The GreatSPN tool: recent enhancements. *SIGMETRICS Performance Evaluation Review, Special Issue on Tools for Performance Evaluation*, 36(4):4–9, 2009.

[3] G. Balbo, M. Beccuti, M. De Pierro, and G. Franceschinis. Stochastic Petri Nets sensitivity to

[7] Transition *DischargeL* is not an exit point since in the subnet derived by projection on P_H, the tagged token entering the subnet through *HospitalArrival* can never reach it.

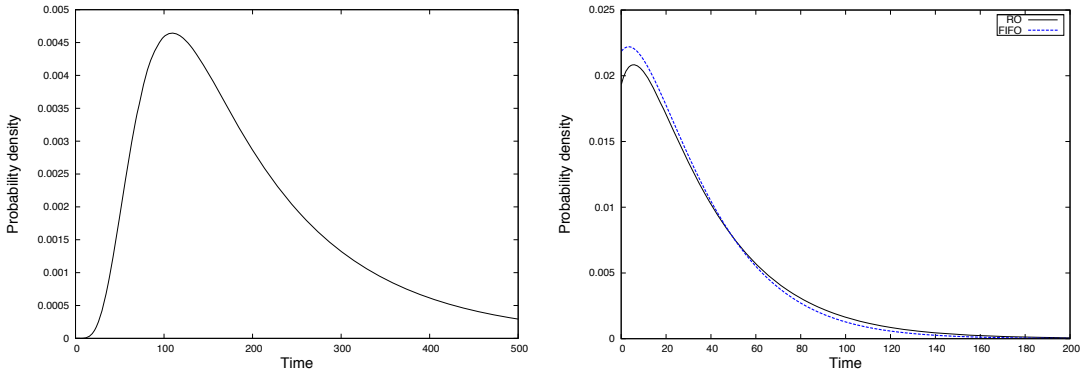
[8] The RG size is computed from the SRG.

Table 2: RG and SRG size for the tagged and untagged ED model varying the numbers of patients.

Configs	Untagged model		Tagged (Ratio) P_H	Tagged (Ratio) P_L (RO)
$\|P_H\|,\|P_M\|,\|P_L\|$	|RG|	|SRG|	|SRG|	|SRG|
2,2,3	2,034,456	162,991	311,802	341,791
2,2,4	7,688,795	255,029	487,503	616,652
2,2,5	29,467,420	368,384	703,800	977,337
2,2,6	113,230,430	503,056	960,693	1,423,828
2,3,6	1,430,943,303	2,436,764	4,662,196	6,817,738
3,2,2	8,388,081	507,870	1,402,926	832,877
4,2,2	130,303,957	2,218,533	7,877,628	3,626,260
3,2,3	30,483,759	892,954	2,480,310	1,858,610
4,2,3	469,988,587	3,893,238	13,796,762	8,064,458

Table 3: ED model transition weights.

Transition	Weight
FallIll,EBloodExam	0.100
EToStabilize	0.050
EToTreate, DischargeL	0.050
BToStabilize	1.000
EX-Ray	0.200
EToSurgery	0.016
DischargeM	0.008
DischargeRec	0.001
ToDoctor	0.70
ToSurgery	0.30



(a) (b)

Figure 8: Pdf of: (a) the time from admission to discharge of P_H patients; (b) the time from admission to discharge of P_L patients with RO or FIFO scheduling policies on place *WaitingRoom*.

token scheduling policies. In *OR2010: Proc. of Int. Conf. Operations Research "Mastering Complexity"*. LNCS Springer, 2010. To appear.

[4] G. Balbo, M. Beccuti, M. De Pierro, and G. Franceschinis. First passage time computation in tagged GSPNs with queue places. *The Computer Journal*, First published online: July 22, 2010.

[5] F. Bause and P. Kritzinger. *Stochastic Petri Nets - An Introduction to the Theory,2nd ed.* F.Vieweg & Sohn Verlag, Braunschweig/Wiesbaden, Germany, 2002. `ls4-www.informatik.uni-dortmund.de/QM/MA/fb/spnbook2.html`.

[6] M. Beccuti, S. Baarir, G. Franceschinis, and J.-M. Ilié. Efficient lumpability check in partially symmetric systems. In *3rd Int. Conf. on Quantitative Evaluation of Systems*, pages 211–221, Riverside, CA, USA, September 2006. IEEE CS Press.

[7] L. Bodrog, G. Horvath, S. Racz, and M. Telek. A tool support for automatic analysis based on the tagged customer approach. In *Proc. of the 3rd int. conf. on the QEST'06*, pages 323–332, Washington, DC, USA, 2006. IEEE CS Press.

[8] J. T. Bradley. Derivation of passage-time densities in pepa models using ipc: The imperial pepa compiler. In *Proc. of the 11th IEEE/ACM Int. Symposium MASCOTS03*, pages 344–351. IEEE CS Press, 2003.

[9] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets for symmetric modelling applications. *IEEE Trans. on Computers*, 42(11):1343–1360, nov 1993.

[10] A. Clark and S. Gilmore. State-aware performance analysis with extended stochastic probes. In *EPEW08: Proc. of the 5th European Performance Engineering Workshop, LNCS 5261*, pages 125–140. Springer, 2008.

[11] J.-M. Couvreur, S. Haddad, and J. F. Peyre. Generative Families of Positive Invariants in Coloured Nets Sub-Classes. In *Proc. of the 12th International Conference on ATPN*, pages 51–70, Chicago, Illinois, USA, 1993. LNCS Springer.

[12] N. Dingle, P. Harrison, and W. Knottenbelt. Uniformisation and Hypergraph Partitioning for the Distributed Computation of Response Time Densities in Very Large Markov Models. *Journal of Parallel and Distributed Computing*, 64(8):309–920, 2004.

[13] N. J. Dingle and W. J. Knottenbelt. Automated Customer-Centric Performance Analysis of Generalised Stochastic Petri Nets Using Tagged Tokens. *Electron. Notes TCS*, 232:75–88, 2009.

[14] K. Jensen. *Coloured Petri nets. Basic Concepts, Analysis Methods and Practical Use (vol.1,2,3)*. Springer Inc., New York, NY, USA, 1997.

[15] S. Kounev. Performance Modeling and Evaluation of Distributed Component-Based Systems Using Queueing Petri Nets. *IEEE Trans. Softw. Eng.*, 32(7):486–502, 2006.

[16] S. Wau Men Au-Yeung. *Response Times in Healthcare Systems*. PhD thesis, Imperial College, London, 2008. `pubs.doc.ic.ac.uk/response-times-in-healthcare`.