Efficient analysis/simulation of complex SWN models: a structural approach

Lorenzo Capra
Dip. di Informatica e Comunicazione,
Milano, Italia
capra@dico.unimi.it

Abstract— **Petri nets structural techniques may be used not only to study interesting system properties, but also to manage state space explosion and discrete-event simulation complexity. In order to avoid net unfolding, a symbolic calculus for high level PN structural analysis is needed, directly working on net annotations. Colored Petri Nets structural analysis relies upon a common subset of functional operators. A language to denote structural relations in WN (a CPN flavor retaining expressive power) was recently introduced and proved to be closed under these operators. The paper shows an application of structural techniques for efficient performance-oriented analysis of a complex Stochastic WN model representing a P2P configuration management tool. Structural techniques are used in combination with symmetries specified through the SWN formalism.**

I. Introduction

High Level Petri Nets (HLPN) [1] have been introduced to allow parametric and compact representations of complex discrete-event systems. Although HLPN models can be translated into PN ones by applying an unfolding procedure, ad-hoc analysis techniques may be applied directly to the high level representation: this approach leads to improved efficiency and allows also parametric results.

Several types of analysis techniques can be applied to HLPN models, for example a reachability graph (RG) can be computed, or structural analysis methods can be applied which allow to derive marking independent properties. Structural analysis results are often used to check some properties before RG construction, and they can be exploited for early detection of modeling errors, or for improving the efficiency of the RG construction process.

Well-formed Nets (WN) [2] are a Colored PNs (CPN) [3] flavor retaining expressive power, characterized by a structured syntax that has been exploited for developing efficient RG generation algorithms [4]. Some results on HLPN models structural relation computation have been presented in [5], however they apply to a restricted subclass of WN. Such results are extended to the WN class in [6], [7]: the core result is the formalization of a calculus for the computation of *symbolic structural properties* and of *symbolic structural relations* between WN nodes (e.g., the structural conflict and causal connection between transitions). A language for representing such properties and relations in symbolic form is proposed, which is a simple extension of WN arc function syntax. The symbolic calculus we developed in [6], [7] can be used for a number of different applications: structural relations can improve the RG construction process efficiency [8]; they are also useful in the construction of Stochastic WN (SWN) models to help the modeler in correctly specifying the underlying stochastic process [9];finally structural considerations can speed up the computation of enabled transition instances in HLPN discrete-event simulation or RG construction [10].

In this paper the calculus defined in [6], [7] is applied to a complex SWN model representing a peer-to-peer configuration management tool [11]. A first version of the model has been presented and analyzed in [12], where a simplified scenario has been considered. The model we study in this paper inherits the basic structure from [12], but introduces a multi-document cooperation setting typical of modern software development, that further complicates color annotations, and consequently compromises the analysis process. The focus of the paper is on a novel technique (here semi-formally described through the application example) that combines computation of SWN symbolic structural relations and verification of (conjectured) colored invariants, to approach the otherwise untreatable complexity of analysis. The symbolic calculus that the technique relies upon is currently semi-automated, but we are confident of a forthcoming full implementation. What is interesting, structural considerations like those used in this setting are orthogonal to exploitation of symmetries encoded in SWNs.

The paper is organized as follows: Section II introduces the basic WN's definitions and the notation used in the paper. Section III motivates the combined use of SWN structural analysis with other analysis techniques and recalls the WN structural properties and the language for expressing them. Section IV shortly presents the SWN model of the P2P architecture. Section V semi-formally describes the technique used to manage the model complexity. Section VI comments on a selection of performance results. Section VII discusses the paper outcomes and the planned future work .

II. Stochastic Well-formed Nets

This section highlights the aspects of the SWN formalism which are useful to describe the topic of this paper. We refer to [2] for the complete SWN definition.

Well-formed Nets (WN) are a Colored Petri Nets (CPN) flavor [3] retaining expressive power. Unlike

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

CPNs, the WN formalism includes transition priorities and inhibitor arcs. Stochastic WNs (SWN) represent the high-level version of Generalized Stochastic Petri Nets (GSPN) [9], from which they inherit time representation.

As in CPN, each place and transition has an associated *color domain* which is defined as the Cartesian product of finite *basic color classes* denoted by symbols $C_1, \ldots, C_n$. The mapping $\mathcal{C}(s)$ is used to denote the color domain of a SWN node s. Thus $\mathcal{C}(s) = C_1^{m_{1,s}} \times C_2^{m_{2,s}} \times \ldots \times C_n^{m_{n,s}}$, the superscript $m_{i,s} \in \mathbb{N}$ being the multiplicity of C_i in $\mathcal{C}(s)$. Basic color classes may be partitioned into *static subclasses* (the j-th static subclass of C_i is denoted $C_{i,j}$) and may be cyclically ordered. A marking M maps each place p to a multiset on $\mathcal{C}(p)$ (the set of multisets on A is denoted $Bag[A]$, whereas the formal sum $\sum_{a_j \in A} \lambda_j.a_j$, $\lambda_j \in \mathbb{N}$, denotes a multiset).

As in CPN, functions on color domains label SWN arcs. Symbols W^+, W^-, H are used to denote the (family of) functions labeling output, input and inhibitor arcs, respectively. A function F labeling an arc between transition t and place p is a mapping $F : \mathcal{C}(t) \rightarrow Bag[\mathcal{C}(p)]$. The SWN formalism prescribes a syntax to express such functions, built on a limited set of primitive symbols:

$$F = \sum_i \lambda_i T_i[guard_i], \quad \lambda_i \in \mathbb{N}$$

The innermost term T_i is a function *tuple* $\langle f_1, \ldots, f_k \rangle$ and represents a mapping $\mathcal{C}(t) \rightarrow Bag[\mathcal{C}(p)]$. Each tuple component f_i, called *class-functions*, is in turn a mapping $\mathcal{C}(t) \rightarrow Bag[C_j]$. The application of tuple T to color $c \in \mathcal{C}(T)$ is defined as $T(c) = \otimes_{i=1}^{k} f_i(c)$, $\otimes$ being the multiset Cartesian product operator.

Each f_i is formally expressed as linear combination of *elementary* functions: $X_k, !X_k, S, S_q$. The projection X_k maps a color $c \in \mathcal{C}(t)$, $c = \langle c_1, \ldots, c_m \rangle$, into its k-th component $c_k \in C_j$, the successor $!X_k$ maps a color $c \in \mathcal{C}(t)$ into the successor of its k-th component and may be used only when basic class C_j is ordered, the constant function S maps a color $c \in \mathcal{C}(t)$ into the constant multiset $\sum_{c \in C_j} c$, the constant S_q on the q-th static subclass of C_j maps its argument into the constant multiset $\sum_{c \in C_{j,q}} c$.

$[guard_i] : \mathcal{C}(t) \rightarrow Bag[\mathcal{C}(t)]$ in the expression of F represents a guard. Guards are functions which are right composed to function tuples, and act as the identity function for those elements of the domain which satisfy a given condition, otherwise map into the empty multiset. Guards can also be associated with transitions to restrict the transition color domain. SWN guards are Boolean expressions defined in terms of basic predicates ($X_i = (\neq) X_j$, $d(X_i) = (\neq) C_{j,q}$) testing equality between components of the guard argument, and membership of a component to a given static subclass, respectively.

The linear extension of WN functions often is considered: the linear extension F* of function F is assumed by definition: $F^*(a+b) = F(a)+F(b)$. In the rest of the paper F will be used instead of F^*, abusing notation.

A priority level is associated to each SWN transition; level 0 is reserved for *timed* transitions (graphically represented as white boxes), while priority levels > 0 are for *immediate* transitions (graphically represented as black bars), which fire in zero time. Each timed transition is characterized by a random firing delay with exponential pdf. *Weights* associated to each immediate transition are used to probabilistically characterize the conflict resolution policy between immediate transitions with equal priority.

A color instance $c_t \in \mathcal{C}(t)$ *has concession* in M iff for each input place p of t $W^-(t,p)(c_t) \leq M(p)$, for each inhibitor place p of t $H(t,p)(c_t) >' M(p)$, and $guard_t(c_t) =$ true ($>'$ restricts the comparison to non-zero elements of the multi-set $H(t,p)(c_t)$). c_t is *enabled* in M if has concession, and no higher priority color instances have concession in M (those markings in which no immediate transitions are enabled are called *tangible*, the other are called *vanishing*). The firing of a color instance occurs as on CPN.

As result of the adopted time representation, the reduced Reachability Graph of a SWN (obtained by removing vanishing markings) is isomorphic to a Continuous Time Markov Chain (CTMC) (see [2]).

III. SWN structural analysis application

The structured syntax of WN's arc functions allows the symmetries of a system to be encoded into its model; in this way efficient methods can be applied that exploit symmetries to build a compact state space (the Symbolic Reachability Graph, or SRG, from which a lumped CTMC is directly derived) or to execute symbolic discrete-event simulation runs.

Many analysis tools base their algorithms on the ability to derive structural relationships on the net, and sometimes even state-space based methods take benefit from structural considerations; an example is the Extended Conflict Sets (ECS) technique. ECS were originally defined for GSPN [9] as a partition of immediate transitions into independent conflict sets. The firing of a transition in a given ECS only affects the enabling status, or the firing probability, of the other transitions in the same ECS. This step on one side is essential to the correct net-level definition of the stochastic process underlying GSPN. On the other side, it allows a drastic reduction of the number of vanishing markings generated during the reachability graph construction, lowering the immediate transition interleaving: transitions belonging to different ECS may indeed be fired according to an arbitrary order, without altering the quantitative/qualitative semantics of the model. The ECS definition relies on the computation of simple structural relations between GSPN immediate transitions (basically structural conflict and structural causal connection).

When considering high-level Petri nets, structural dependency relations should be stated in a symbolic form, to characterize in a compact and parametric way the *color instances* of a given transition that may influence the enabling/firing of other transition color instances. A first question is how to express the known

PN structural relations in WNs. The second question is how to efficiently manipulate them to extend the PN algorithms to work on (S)WN models.

In [5] an elegant way to express colored structural relations in CPN is introduced. A relation $\mathcal{R}$ between two colored nodes s, s' is defined by means of a function $F(s,s') : \mathcal{C}(s') \to Pow[\mathcal{C}(s)]$ ($Pow[A]$ denotes the power-set on A) mapping any color instance c' of s' into the instances c of s that are in relation $\mathcal{R}$ with instance c'. A symbolic relation thus represents in a compact way a binary relation between several pairs of nodes in the unfolded net. In [5] it is shown that structural relations may be computed by means of formulas expressed in terms of arc functions and appropriate operators. The key point is the ability to determine algorithmically, and without unfolding the high-level net, the expressions above. An algorithm to compute such expressions was proposed in [5] for a restricted WN subclass, Unary Regular colored nets.

Structural analysis has been recently extended to the (S)WN class of nets [6], [7]. The core result is the formalization of an algebraic calculus for computing symbolic structural properties and relations between WN nodes. An intuitive high-level language for representing such properties and relations has been proposed, based on the WN arc function syntax. The derivation of symbolic structural properties and relations relies upon the application of a basic set of functional operators (support, difference, transpose and composition) on language expressions.

The language introduces two main novelties: (1) class-functions (i.e., tuple components) may be intersections of elementary functions, (2) guards can appear on the left of a tuple, i.e., they can be left composed to a tuple: in this case they are called *filters*. For a given domain D and codomain D', the language $\mathcal{R}[\mathcal{D}; \mathcal{D}']$ is the collection of the expressions which are weighted sums of tuples (having D and D' as domain and codomain, respectively) possibly preceded by a filter and followed by a guard. As an example the following element belongs to $\mathcal{R}[\mathcal{C} \times \mathcal{C} \times \mathcal{C}; \mathcal{C} \times \mathcal{C}]$: $\langle X_1 \cap (S - X_2), S\rangle + \langle X_1, X_3\rangle$. The WN structural relation language had been shown to be closed under the basic set of operators.

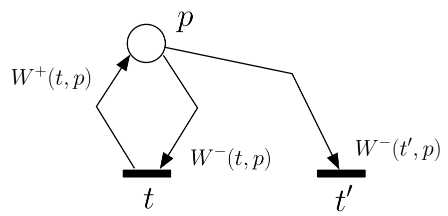

Fig. 1. Structural conditions for conflict

A. Example of Symbolic Structural Formula

Let us consider the structural conflict between t and t' with respect to place p:

$$\mathrm{SC}_p(t,t') = \overline{W^-(t,p) \ominus W^+(t,p)}^t \circ \overline{W^-(t',p)}$$

$W^-(t,p) \ominus W^+(t,p)$ is the the multiset of tokens actually removed from place p by the firing of an instance of t. ($\ominus$ is the multi-set difference, in which resulting negative coefficients are set to 0). The *support* of such expression, The support $\overline{W^-(t,p) \ominus W^+(t,p)}$, takes those colors whose multiplicity is strictly greater than zero. Its *transpose*, $\overline{W^-(t,p) \ominus W^+(t,p)}^t : \mathcal{C}(p) \to Pow[\mathcal{C}(t)]$, results in the colors instances of t that actually remove tokens from p. Concluding, applying $\overline{W^-(t,p) \ominus W^+(t,p)}^t$ to the set $\overline{W^-(t',p)}$ (by composition), then the color instances of t removing from p some token for which t' has need are obtained. $SC(t,t')$ is thereby defined as $\bigcup_{p \in P} SC_p(t,t')$.

IV. The SWN model of a P2P version system

In this section we shortly present the model of the peer-to-peer (P2P) architecture for configuration management systems implemented by the PeerVerSy tool [11]. The model we are considering (Figure 2) extends that one presented in [12], introducing the representation of a multi-document cooperation setting, and has been developed using the GreatSPN package [13]. The model's building-block, the peer life-cycle, was built by specializing the client-server architecture's model (not depicted due to lack of space), according to the OO parlance. The interested reader can refer to [12] for a thorough discussion on modeling methodological issues.

The model's basic color classes are: UID, DID, ST, UPD. Class UID denotes the group of cooperating workers, DID denotes the set of shared artifacts. If $|DID| = 1$ we fall in the scenario considered in [12]. Classes ST and UPD denote the on-line status (*on,off*) of a peer, and the up-to-date status (*upd,noupd*) of the local copy of an artifact owned by a peer, respectively. Either class is accordingly partitioned.

The whole model has been obtained by gluing three submodels via superposition of boundary places (*Repository*, *WCopyUpdated*, *UpdEvent*). Submodels, enclosed within dashed boxes in Figure 2, represent: i) the peer life-cycle ii) the infrastructure reaction to a peer going online, iii) the infrastructure reaction to a successful check-in.

Let us briefly comment on the interpretation of color domains of submodel i). Each place has domain $UID \times DID$, and represents (with the exception of *Authority*) the status of a peer while working on a given document. The association authority peers-artifacts is fixed by the marking of place *Authority*. Submodel's transitions have color domain $UID \times DID \times \ldots$: every transition instance represents an operation done by a peer on a given artifact. Timed transitions represent the main time-consuming activities done by peers (e.g. merging) while immediate transitions represent logical or time negligible actions.

We had to differentiate the authority from normal peers because of their different prerequisites: e.g., the authority can accomplish a check-out, or a succeeding check-in, also if he/she is off-line. A peer can issue a check-in request for a given artifact also if he/she and/or the corresponding authority are not currently on-line: the check-in status is thus set *Pending*, wait-

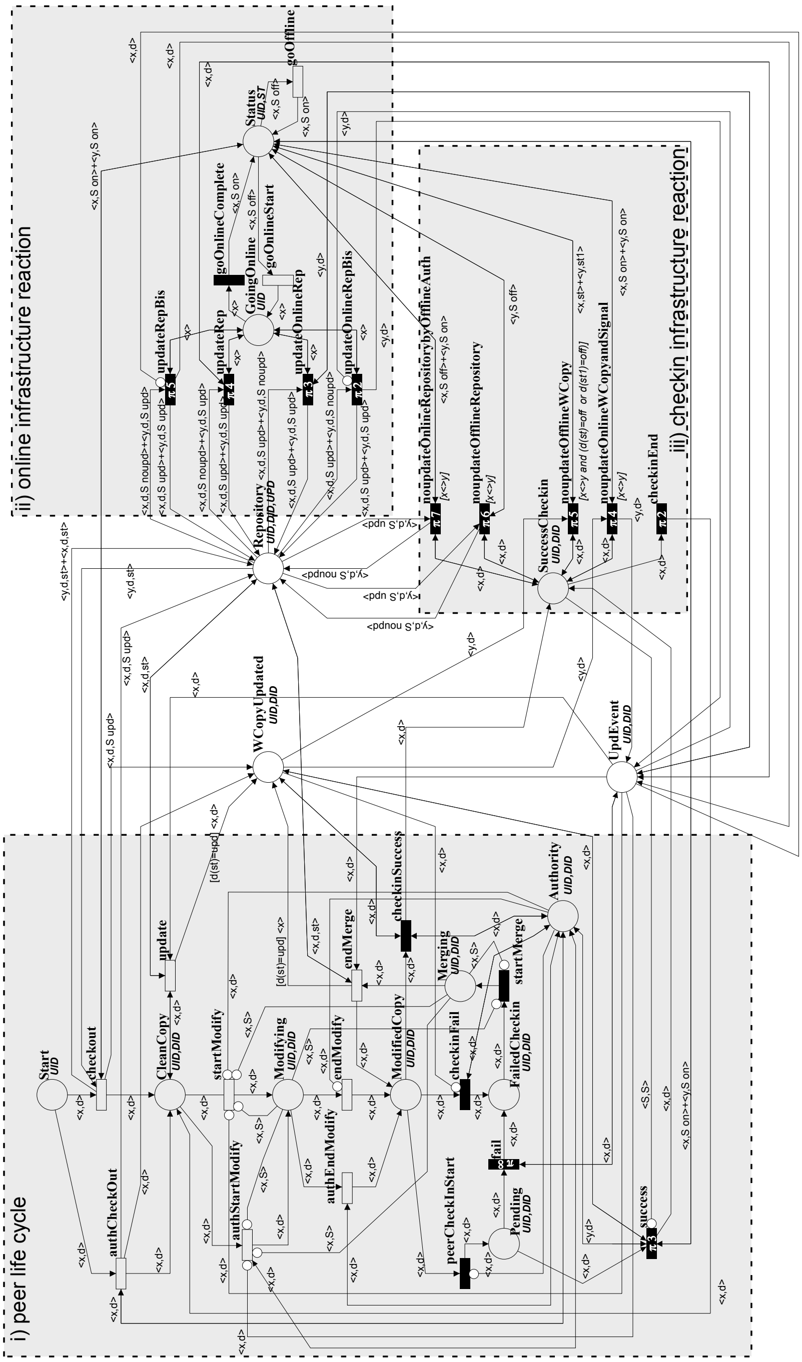

Fig. 2. The SWN model of the P2P architecture

ing for a condition in which it is possible to evaluate whether the check-in succeeded or not.Conflicting check-ins might be simultaneously pending.

In order to exploit symmetries, we chose not to use version numbers for checked-out artifact working copies (as happens in the real world), but to directly maintain the information about the up-to-date status of artifacts in place *WCopyUpdated*. Place *Repository* permits us to model not only the status of the explicitly checked-out copy of an artifact, but also the status of the repository replica automatically managed by the PeerVerSy infrastructure. Place *UpdEvent* represents the sending of a message to the peers by the system signaling that a new version of an artifact is available.

The infrastructure sub-models are in charge of coherently updating these information. Sub-model ii) represents what happens when a peer goes online. In case the version owned by the peer is newer than those owned by the already on-line peers, a signal is automatically sent to such peers, that then update their repository replica. Conversely, the repository of the peer is updated after receiving the signal. Sub-model iii) represents what happens when a check-in is accomplished by a peer: the working copy of any other peer is set to non-up to date, as well as the repository replica of all the off-line peers. A signal standing for new version available is sent to on-line peers, then all the conflicting pending check-ins are aborted ($fail$ transition of i). The check-in can be accomplished by the authority when he/she is off-line: in that case no signal is sent and all working copies and repository replica are set as non-up-to-date.

V. Dealing with model complexity

In this section we show how (symbolic) structural analysis may be used to significantly improve both the effectiveness and the efficiency of SWN techniques that exploit symmetries. Structural considerations not only impact on exact analysis, reducing the SRG size, they also permit simulation times to be kept under acceptable bounds, by drastically simplifying the (usually expensive) transition enabling test.

In order to exploit behavioral symmetries, a suitable initial *symbolic marking* (SM) must be set. The SM defines an equivalence relation between markings in a syntactical fashion: two ordinary markings are equivalent if they can be obtained from each other by applying a permutation on split color classes that preserves static subclasses. A *symbolic* firing rule allows the building of a compact Symbolic Reachability Graph (SRG), working directly at SM level (an associated lumped CTMC is automatically computed in SWN). Alternatively, the symbolic discrete-event simulation engine of GreatSPN can be used. The initial SM ($\widehat{M_0}$) of the SWN in Fig.2 is:

$\widehat{M_0}(cleanCopy) : 1 \cdot \langle \mathsf{S}, \mathsf{S} \rangle$; $\widehat{M_0}(status) = 1 \cdot \langle \mathsf{S}, \mathsf{S}_{on} \rangle$;
$\widehat{M_0}(auth) : \sum_{k:1}^{n_d} 1 \cdot \langle Z^k_{UID}, Z^k_{DID} \rangle$, $|Z^k_{UID}| = |Z^k_{DID}| = 1$;
$\widehat{M_0}(Repository) = \widehat{M_0}(wcopy) : 1 \cdot \langle \mathsf{S}, \mathsf{S}, \mathsf{S}_{upd} \rangle$

A SM is formally expressed in terms of dynamic subclasses, denoted by symbols $Z^k_{C_i}$ ($Z^k_{C_{i,j}}$), that define a *parametric partition* of color (sub)classes. A constant function symbol occurring on a SM denotes a whole color (sub)class.

$\widehat{M_0}$ represents a situation in which every developer is ready to start his activity, is working on all documents, and is currently on line. $\widehat{M_0}$ is *generic*: the set of (n_d) shared artifacts are associated to an arbitrary subset of (n_d) authoritative peers (we are assuming $n_p \geq n_d$). This configuration is of particular interest, being the most decentralized one. Note that we are skipping the transient system behavior (*checkout* actions): in such a way we achieve some modest reduction, without affecting steady-state solution.

The model symmetry level is measured by the color class splitting level. That means a *potential* reduction factor (with respect to the nodes of the ordinary RG) of $n_p! \cdot n_d!$, $n_p = |UID|$, $n_d = |DID|$. The set of authoritative peers, as well as the association authority-artifact, are not fixed in $\widehat{M_0}$. Based on simple combinatorics, we argue that the potential reduction factor for the P2P SWN model is considerably lower than the theoretical one, that must be divided by factor $r = \binom{n_p}{n_d} \cdot n_d!$. Factor r corresponds to the number of possible choices of the authorities among the n_p peers, multiplied by the number of possible associations authority-artifact. The (potential) reduction factor due to symmetries for this SWN model becomes: $(n_p - n_d)! \cdot n_d!$. In other words, the lower the difference between number of peers and documents, the lower the symmetry effectiveness.

A. Reduction via structural analysis

The main point of criticism during the construction of the SRG of the P2P SWN model lies in the enormous number of generated vanishing SMs, because of the frequent interleaving among immediate transitions. This aspect affects not only analysis but also discrete-event simulation. With some analogy with the ECS approach, we try to reduce such interleaving by exploiting the information provided by SWN symbolic structural relations and colored invariants. The model immediate transitions are partitioned into independent conflict sets, each composed by transitions whose color instances do not depend by color instances of transitions belonging to other conflict sets. Even if rougher than a ECS-like partitioning (in which color instances of a transition would be in turn partitioned), this technique allows us to considerably reduce the number of vanishing SM, by assigning different conflict sets different priority levels. Intuitively speaking, the core idea of the algorithm is to verify, for each pair of immediate transitions t, t' to whom the modeler assigned the same priority, whether color instances of t' are in structural mutual exclusion with conflicting color instances of t (formally: $SC(t,t') \ominus SME(t,t') = \emptyset$), and vice versa. In such a case t and t' are set as independent.

For example, let us compute $SC_p(t,t')$ for p: *Repository*, t': *noupdateOfflineRepository*, t: *noupdateOnlineRepositoryOfflineAuth*, (submodel iii), Fig.2), where $\mathcal{C}(t) = \mathcal{C}(t') : UID^2 \times DID$, and $\mathcal{C}(p) : UID \times DID \times UPD$. After renaming symbols x, y, d with projection symbols X_1, X_2, X_3, respectively, we obtain by applying the algorithms defined in [6], [7] that $\overline{W^-(t,p) \ominus W^+(t,p)}$ is equal to $\langle X_2, X_3, \mathsf{S}_{upd} \rangle [X_1 \neq X_2]$. Its transpose is a function $UID \times DID \times UPD \rightarrow Pow[UID^2 \times DID]$:

$$[X_1 \neq X_2] \langle \mathsf{S}, X_1, X_2 \rangle [d(X_3) = upd]$$

By applying the composition rules we finally obtain:

$$\begin{gathered}[X_1 \neq X_2] \langle \mathsf{S}, X_1, X_2 \rangle [d(X_3) = upd] \circ \\ \langle X_2, X_3, \mathsf{S}_{upd} \rangle [X_1 \neq X_2] \\ \equiv \langle \mathsf{S} - X_2, X_2, X_3 \rangle [X_1 \neq X_2]\end{gathered}$$

$\langle \mathsf{S} - X_2, X_2, X_3 \rangle [X_1 \neq X_2]$ is the symbolic representation of $SC_p(t,t')$ (we can easily argue that it corresponds to $SC(t,t')$). We can give this result an intuitive interpretation: the set of color instances of t conflicting with a color instance $\langle c_1, c_2, c_3 \rangle$ of t' is empty if $c_1 \neq c_2$, otherwise is: $\{\langle c_x, c_2, c_3 \rangle : c_x \neq c_2\}$.

Since t and t' have a non-empty set of potential conflicting color instances, we have to go one step beyond, considering whether color instances in structural conflict are in truth mutually exclusive due to some *symbolic* P-invariant [3]. A symbolic P-invariant (or flow) in a CPN is a P-vector $\vec{I}$ whose components are functions $I_p : \mathcal{C}(p) \to Bag[A]$, A being a domain built from colors common to all the places corresponding to non-null entries of $\vec{I}$. Each I_p maps a place marking into a new multiset, since all these multisets have the same support they can be added together to give a weighted sum, $\sum_{p\in P} I_p(M(p))$ (the function linear extension on multisets with integer coefficients is implicitly considered here), that must be the same for each marking.

Letting $\vec{\mathcal{H}}$ be the $|P|\cdot|T|$ incidence matrix of a CPN ($\mathcal{H}_{[p,t]} = W^+(t,p) - W^-(t,p)$) a P-invariant $\vec{I}$ is a solution of the homogeneous system $\vec{I}\cdot\vec{\mathcal{H}} = \vec{\emptyset}$, where $\circ$ is used as scalar multiplication operator. To our knowledge, there is no way to automatically compute invariant basis for WN nets, but only for a restricted subclass. The symbolic manipulation of the composition operator treated in [7], however, allows us to verify any *conjectured* invariant expressed using the language for WN structural relations.

As concerns our SWN model, we can easily infer a simple P-invariant $\vec{I}$ whose only entries different from the constant function $\emptyset$ are (p_1: *GoingOnline*, p_2: *Status*)

$$I_{p_1} = 1\cdot\langle X_1\rangle : UID \to Bag[UID]$$
$$I_{p_2} = 1\cdot\langle X_1\rangle : UID \times ST \to Bag[UID]$$

from which we obtain the invariant expression:

$$1\cdot\langle X_1\rangle(\widehat{M_0}(p_1)) + 1\cdot\langle X_1\rangle(\widehat{M_0}(p_2)) = 1\cdot\langle \mathsf{S}\rangle$$

whose meaning is clear: $\forall c_1 \in UID$ either there is one token $\langle c_1\rangle$ in p_1, or one token $\langle c_1, \_\rangle$ in place p_2. From the structural conflict between t and t', we can prove (in a fully symbolic way) that the *simultaneous* enabling of an instance $\langle c_1,c_2,c_3\rangle \in \mathcal{C}(t')$ and a conflicting color instance $\langle c_x,c_2,c_3\rangle \in \mathcal{C}(t)$, ($c_x \neq c_2$) is impossible: the resulting expression for simultaneous enabling of t,t' in p_2 turns out to be (letting $m,m' \in Bag[A]$, then $\max(m,m') \in Bag[A]$ is defined as: $\forall c, \max(m,m')(c) = \max(m(c),m'(c))$)

$$\max\left(W^-(p_2,t')(\langle c_1,c_2,c_3\rangle), W^-(p_2,t)(\langle c_x,c_2,c_3\rangle)\right) =$$
$$1\cdot\langle c_2,\mathsf{S}_{off}\rangle + 1\cdot\langle c_2,\mathsf{S}_{on}\rangle + 1\cdot\langle c_x,\mathsf{S}_{off}\rangle$$

which is incompatible with the invariant expression above. Computing $SC(t',t)$ we can analogously prove that color instances of t and conflicting color instances of t' are in structural mutual exclusion. We can thus assign t and t' different priorities, reducing their interleaving.

Table I shows the size of the model's SRG (in terms of tangible and vanishing markings) faced to the ordinary RG (RG data have been divided by factor r). To appreciate the impact of structural analysis, the 4^{rd} column reports original SRG data (no structural analysis accomplished).

VI. Peer-to-peer against client-server

We are interested in steady-state evaluation of the SWN model previously described. Configurations too complex to be analytically solved have been simulated. Having at our disposal a significant set of exact results we preselected the performance metrics whit respect to whom simulation turned out to be closest to exact analysis. The data files generated by GreatSPN with $n_p\cdot n_d > 7$ exceeded the maximum file size (4GB) set on a Intel Xeon 2.4GHz. Configurations up to $n_p\cdot n_d = 16$ have been studied by means of the symbolic simulation engine of GreatSPN. Simulation times range between a few seconds and about six hours (for $n_p = 4, n_d = 4$). Simulation of larger configurations are currently unfeasible. Without structural considerations, configurations up to $n_p\cdot n_d = 6$ can be simulated.

We can estimate how the PeerVerSy approach performs under a number of different conditions. Model inputs are: the number of workers (n_p), the number of documents (n_d), the Working profile (W) (we assume that workers repeat cycles in which on average they work 2 units of time and they are idle for a unit of time), and the On-line profile (O). Every member may be on-line or off-line. We characterized the on-line profile with two parameters: On-line ratio (OR) (how much time a worker is on-line against total time) and Connection periodicity (OP) (how long is an on-line/off-line cycle).

We focused on three metrics: i) Average successful check-in throughput per worker per document (sci) (we considered separately normal peers and the authority);ii) Average failed check-in throughput per worker per document (fci); iii) probability that a random document copy is consistent with the reference copy (P(Uptd)) (this metrics may be computed using GreatSPN only when analytical solution is feasible). Table II reports a summary of results of the evaluation of the P2P architecture. Results in bold refer to analysis, while those in normal font refer to simulation. For the sake of comparison, also the results of the analysis of the server-based architecture are reported (in italic). The number of authorities was always kept identical to the number of documents. Results in Table II refer to a working profile in which authorities and peers works equally on a document. In a peer-to-peer setting, when the number of peers and documents increases, the number of check-ins needed to work increases as well. Successful and failed check-ins increase proportionally. Quite surprisingly, the server setting shows an opposite behavior at least as concerns the average success: this trend is only partially balanced by data on failed check-ins, where the server setting turns out to work better if we consider small number of clients, quickly degradating as the number of clients/documents is growing. Moreover, since in a peer-to-peer setting authorities have very frequently the opportunity of checking in their modifications without any conflict, the compound probability that a random working copy is up to date, is in general higher than in the server based architecture. Summarizing the P2P architecture seems

TABLE I: State space reduction

(n_p,n_d)	$\lvert SRG \rvert$ (tang. + van.)	$\lvert RG \rvert \div r$ (tang. + van.)	$\lvert SRG \rvert$ (tang. + van.) (no struct. analysis)
(2,1)	115+331	115+331	115+52544
(3,1)	1392+6136	2669+12082	1392+1537325
(4,1)	10696+63333	55939+354022	10696+10066583
(5,1)	61365+454344	1107077+9167886	unfeasible
(6,1)	286014+2536334	21156835+220184830	unfeasible
(7,1)	1139128+11765597	395386709+5038278122	unfeasible
(2,2)	1037+4108	2039+8212	1037+1557341
(3,2)	235491+1535528	470598+3070648	unfeasible

TABLE II: Summary data of peer-to-peer setting (Analysis and Simulation)

Peers	Docs	sci	fci	sci (auth.)	fci (auth.)	P(Uptd)	sci (server)	fci (server)	P(Uptd) (server)
2	**1**	**0.088**	**0.077**	**0.148**	**0.045**	**0.6**	*0.108*	*0.035*	*0.552*
2	**2**	**0.056**	**0.041**	**0.107**	**0.022**	**0.61**	*0.09*	*0.028*	*0.574*
3	**1**	**0.109**	**0.129**	**0.093**	**0.04**	**0.468**	*0.127*	*0.091*	*0.447*
3	**2**	**0.075**	**0.076**	**0.066**	**0.023**	**0.492**	*0.1*	*0.069*	*0.466*
3	3	0.058	0.049	0.045	0.014		*0.065*	*0.037*	*0.44*
4	**1**	**0.116**	**0.162**	**0.068**	**0.034**	**0.4**	*0.082*	*0.059*	*0.33*
4	2	0.086	0.097	0.048	0.021		*0.07*	*0.05*	*0.345*
4	3	0.066	0.069	0.034	0.013		*0.053*	*0.041*	*0.328*
4	4	0.054	0.047	0.027	0.009		*0.038*	*0.034*	*0.29*
5	**1**	**0.119**	**0.184**	**0.053**	**0.029**	**0.356**	*0.077*	*0.066*	*0.28*
5	2	0.085	0.118	0.036	0.017		*0.052*	*0.057*	*0.3*
5	3	0.066	0.082	0.028	0.012		*0.038*	*0.039*	*0.32*
6	**1**	**0.12**	**0.2**	**0.043**	**0.025**	**0.316**	*0.073*	*0.073*	*0.246*
7	**1**	**0.12**	**0.212**	**0.036**	**0.022**	**0.303**	*0.069*	*0.078*	*0.222*
8	1	0.152	0.259	0.029	0.02		*0.066*	*0.082*	*0.203*
9	1	0.149	0.269	0.027	0.018		*0.063*	*0.086*	*0.188*
10	1	0.155	0.294	0.025	0.016		*0.059*	*0.09*	*0.165*
11	1	0.166	0.311	0.020	0.014		*0.056*	*0.094*	*0.137*
12	1	0.169	0.328	0.020	0.013		*0.052*	*0.097*	*0.1*

to perform better than the server-based one in case of medium-size teams working on a number of shared documents. In case of small teams working on a single document the server-based architecture retains its claimed supremacy.

VII. Conclusion and future work

We have reported an application of a calculus for *symbolic* structural analysis of Stochastic WN (a CPN flavor preserving expressive power) oriented to performance-evaluation of complex models. The SWN analyzed in the paper specifies a P2P configuration management algorithm, and extends an existing model to a multi-document cooperation scenario. Steady state analysis/simulation has been carried out, and a selection of performance metrics has been presented. The original contribution of the paper lies in a structural technique (semi-formally described) for tackling the model's analysis/simulation complexity, based on the combined use of SWN structural relations and colored invariants. Orthogonality of structural approach to symmetries typical of the WN formalism has been exploited. The symbolic calculus that the technique is based upon is currently semi-automated: its full implementation and the generalization of the structural approach are forthcoming goals.

References

[1] Jensen, K., Rozenberg, G., eds.: High-Level Petri Nets. Theory and Application. Springer Verlag (1991)

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

[3] Jensen, K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag. ISBN: 3-540-60943-1. (1997)

[4] Chiola, G., Dutheillet, C., Franceschinis, G., Haddad, S.: A symbolic reachability graph for coloured Petri nets. Theoretical Computer Science B (Logic, semantics and theory of programming) **176** (1997) 39–65

[5] Dutheillet, C., Haddad, S.: Conflict sets in colored Petri nets. In: Proceedings of the 5th International Workshop on Petri Nets and Performance Models, Toulouse, France (1993) 76–85

[6] Capra, L., De Pierro, M., Franceschinis, G.: A High Level Language for Structural Relations in Well-Formed Nets. In Ciardo, G., Darondeau, P., eds.: Proceeding of the 26th International Conference on Application and Theory of Petri Nets. LNCS 3536, Miami, USA (2005) 168–187

[7] Capra, L., De Pierro, M., Franceschinis, G.: Symbolic calculus for structural relations in well-formed nets. In: Technical report RT 03-04; http://www.di.unito.it/∼depierro/public/, D.I.Co, Università di Milano - Italia (2004)

[8] Kristensen, L.M., Valmari, A.: Finding stubborn sets of coloured Petri nets without unfolding. Lecture Notes in Computer Science: 19th Int. Conf. on Application and Theory of Petri Nets, ICATPN'98, Lisbon, Portugal, June 1998 **1420** (1998) 104–123

[9] Chiola, G., Ajmone, M., Balbo, G., Conte, G.: Generalized stochastic Petri nets: A definition at the net level and its implications. IEEE Transactions on Software Engineering **19** (1993)

[10] Gaeta, R.: Efficient discrete-event simulation of colored Petri nets. IEEE Transactions on Software Engineering **22** (1996)

[11] Balzarotti, D., Ghezzi, C., Monga, M.: Freeing cooperation from servers tyranny. In Gregori, E., Cherkasova, L., Cugola, G., Panzieri, F., Picco, G.P., eds.: Web Engineering and Peer-to-Peer Computing. Volume 2376 of Lecture Notes in Computer Science. Springer-Verlag (2002) 235–246

[12] Bellettini, C., Capra, L., Monga, M.: Quantitative assessment of a peer-to-peer cooperative. infrastructure using swns. In: Proc. of First International Workshop on Practical Applications of Stochastic Modelling, (PASM04). ENTCS, ISSN: 1571-0661, London, UK, ELSEVIER (2004)

[13] Chiola, G., Franceschinis, G., Gaeta, R., Ribaudo, M.: GreatSPN 1.7: Graphical editor and analyzer for timed and stochastic Petri nets. Performance Evaluation (1995)