Parameterised CSA-nets

Mohammed Alahmadi[1]

[1] *School of Computing, Newcastle University*
Science Square, Newcastle upon Tyne, NE4 5TG, United Kingdom

Abstract

Communication Structure Acyclic Nets (CSA-nets) are a Petri net-based model used to represent the execution behaviour of Complex Evolving Systems (CESs). CSA-nets consist of sets of acyclic nets that communicate with each other through a set of buffer places. However, this can generate an excessive number of buffer places, making the model hard to visualize and analyse. Additionally, having several components that perform similarly can result in the development of a complicated model. This paper introduces two extensions for CSA-nets, which are used to depict the relationships between interacting systems' components represented by sets of acyclic nets. Specifically, it introduces a way of folding buffer places to address the issue of a large number of buffer places. In addition, it introduces parameterisations for CSA-nets using multi-coloured tokens to extend places to accept multiple tokens distinguished by parameters. The combination of these techniques should lead to improved visualization and analysis of large and complex CSA-nets. We apply mechanisms found in the domain of coloured Petri nets.

Keywords

Petri net, acyclic net, communication structured acyclic net, parameterisation, folding, visualisation

1. Introduction

Today's world is associated with large amounts of data stored in servers, hard drives, and the cloud, among other places. Regardless of the discipline, modern technologies need to be able to process this data, whether it relates to education, health, crime, or anything else. In particular, there are enormous amounts of data available, and this quantity is only going to continue to grow due to advances in digital sensors, communications, and storage that generate vast amounts of data. This needs the integration and interaction of several subsystems, including data storage, processing, analysis, and visualization, among others.

A complex evolving system (CES) is composed of a large number of subsystems that interact with each other concurrently. Such a system is often characterized by a dynamically changing structure and features, as well as having intricate and large behavioral patterns. In other words, when new features are added, the behavior of the entire system may be affected. That is, these systems are highly complex in terms of interpreting their behavior and visualizing it [1][2].

Structured occurrence nets (SONs)[3][4][5] are a Petri net-based model for representing the execution behavior of CESs. They are an extension of occurrence nets, which represent the causality and concurrency information relating to a single system execution [4]. Communication Structured Occurrence Net (CSO-net) consists of several acyclic nets linked through different

CEUR-WS.org/Vol-3430/paper10.pdf

PNSE'23, International Workshop on Petri Nets and Software Engineering, June 26-27, 2023, Lisbon, Portugal
m.s.h.alahmadi2@ncl.ac.uk (M. Alahmadi)
© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

types of formal relationships [3]. It provides full and clear record of all causal dependencies between the events involved in a single ‘causal history’. That is, only cycles involving buffer places are allowed. CSO-nets exhibit backward determinism and forward determinism [3, 5]. However, we also consider CSA-nets with only forward nondeterminism.These nets are intended to capture information about (i) the interaction between actual/expected behaviors and (ii) the collected evidence to be analyzed. Moreover, they can represent the different actions of dynamic evolving systems [3]. Each component acyclic net represents a local view of the system’s behavior, which is generally easy to interpret. However, an analogous representation at the global system level leads to complex visualizations, especially in evolving systems. This decreases the system’s interpretability.

The particular formal model used in this paper is communication structured acyclic nets (CSA-nets) and is used in [6][7], which are generalizations of CSO-nets where the individual acyclic nets are linked by buffer places capable of modeling both asynchronous and synchronous communication between different subsystems. In other words, events are used to link the acyclic nets with each other through buffer places. Figure 1 shows an example of a CSA-net with two ‘horizontal’ occurrence nets (the upper one exhibiting concurrency) connected by two buffer places, q_1 and q_2. The idea is that these places establish communication links. When the buffer places are used asynchronously, the CSA-net can generate, for example, the step sequence d, ab, ec. Additionally, when they are used synchronously, the step sequence da, b, ce can be executed. In other words, there is an instantaneous transfer of tokens generated by a and b to transition c thanks to the connections through buffer places. This feature can be used to model synchronous communication between the component occurrence nets in particular. Nodes linked by arcs are one of the common graphical representations in the field of visualization. This technique is widely applied in different areas, such as bioinformatics, software engineering, VLSI circuits, and networking. Improving the representation of CSA-nets would lead to better comprehension of the problem under investigation. In particular, it could help investigators detect causality and dynamic behaviors between events, which would, in turn, aid in analyzing such systems. However, as the size of data being represented grows, the number of nodes increases even more rapidly. Consequently, such a large number of nodes often make the model under investigation difficult to follow and understand, resulting in comprehension difficulties.

One example of CES systems where there is a large amount of data is cybercrime investigation. In such cybercrime applications, the generated data is of high importance, and high-quality analysis is needed. Proper visualization of CSA-nets with a large amount of data would result in a much better understanding for the investigators and the possibility of linking crime events and locations with data representations that help detect crime. This is done through extracting valuable knowledge from large and complex data and identifying useful information from a variety of data sources. A major specific benefit to cybercrime investigation is that such analysis allows early detection of crimes, hence crime prevention. The current work is an attempt in this direction, i.e., visualization improvement [1, 2].

CSA-nets visualization framework allows for the analysis of complex systems. The hurdle is that managing large amounts of data is not an easy task. In other words, with the high number of components (acyclic nets), there is a large number of buffer places leading to difficult analysis and visualization. What also adds to this complexity is that CSA-nets are limited to presenting only one token in a transition at any given time, which, in turn, limits its ability to represent, e.g.,

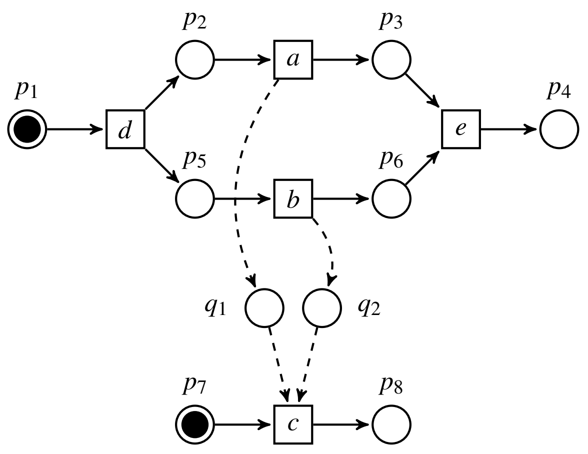


Figure 1: A CSA-net model.

several similar processes at the same time. In general, system parameterisation is used to examine how changing input parameters affects the output of a system. By defining a set of different parameters, one can obtain a corresponding set of different outputs without the need to rebuild the system model every time. This advantage allows for the preparation of the system model once and its execution several times, providing insights into the relationship between inputs and outputs. It also simplifies the system tracing process by automatically changing the results of one parameter when other parameters are changed.

This paper aims to improve the visualization of CSA-nets by addressing two issues. First, it extends the work done in [8] by providing the necessary formalization of the concept of *master buffer places*. This concept adds a folding mechanism to CSA-nets and addresses the issue of a large number of buffer places. Second, we introduce a new class of CSA-nets called *parameterised* CSA-*nets* that extends the single-token concept. This idea aims to allow places to accept multiple tokens distinguished by parameters.

2. Preliminaries

This section presents notions related to acyclic nets and communication structured acyclic nets.

2.1. Acyclic nets

An *acyclic net* is a triple $acnet = (P,T,F)$, where $P(= P_{acnet})$ and $T(= T_{acnet})$ are disjoint finite sets of *places* and *transitions*, respectively, and $F(= F_{acnet})$ is the *flow relation* included in $(P \times T) \cup (T \times P)$ such that: (i) P is nonempty; (ii) F is acyclic; and (iii) for every $t \in T$, there are $p,q \in P$ such that pFt and tFq. For every node $x \in P \cup T$,

$$\text{pre}_{acnet}(x) = \{z \mid zFx\} \ \ \textit{and} \ \ \text{post}_{acnet}(x) = \{z \mid xFz\}$$

and similarly for sets of nodes.

The *markings* of *acnet*, denoted by markings($acnet$), are the subsets of P. Hence, the nets we are dealing with are safe. [1] The default *initial* marking is $M^{init}_{acnet} = \{p \in P \mid \text{pre}_{acnet}(p) = \varnothing\}$, and the steps are

$$\text{steps}(acnet) = \{U \in 2^T \mid \forall t \neq u \in U : \text{pre}_{acnet}(t) \cap \text{pre}_{acnet}(u) = \varnothing\} \ .$$

Graphically, places are represented by circles, transitions by boxes, arcs between the nodes represent the flow relation, and markings are indicated by black tokens placed inside the corresponding circles.

A step U of *acnet* is *enabled* at marking M if $\text{pre}_{acnet}(U) \subseteq M$. It can then be *executed* and yield

$$M' = (M \cup \text{post}_{acnet}(U)) \setminus \text{pre}_{acnet}(U).$$

This is denoted by $M[U\rangle_{acnet} M'$.

To capture the behaviour of acyclic nets (and other nets later on), we use step sequences, involving reachable markings and executed steps. Let $(M^{init}_{acnet} =)M_0, M_1 \dots, M_k$ $(k \geq 0)$ be markings and $U_1, \dots, U_k$ be steps of *acnet* such that we have $M_{i-1}[U_i\rangle_{acnet} M_i$, for every $1 \leq i \leq k$. Then

$$\mu = M_0 U_1 M_1 \dots M_{k-1} U_k M_k$$

is a *mixed step sequence* of *acnet*. This is denoted by $\mu \in \text{mixsseq}(acnet)$.

A basic consistency criterion applied to acyclic nets is well-formedness, and *acnet* is *well-formed* if the following hold, for every $M_0 U_1 M_1 \dots M_{k-1} U_k M_k \in \text{mixsseq}(acnet)$:

- $\text{post}_{acnet}(t) \cap \text{post}_{acnet}(u) = \varnothing$, for every $1 \leq i \leq k$ and all $t \neq u \in U_i$.
- $\text{post}_{acnet}(U_i) \cap \text{post}_{acnet}(U_j) = \varnothing$, for all $1 \leq i < j \leq k$.

Intuitively, in a step sequence of a well-formed acyclic net, no place receives a token more than once. This ensures an unambiguous representation of causality in the behaviour of *acnet*. Note that well-formedness is stronger than safeness. In particular, it prevents the same transition from executing more than once. So, having an acyclic net represents a single execution, meaning that each transition can only be executed once.

2.2. Communication structured acyclic net

A *communication structured acyclic net (or* CSA*-net)* is a tuple

$$csan = (acnet_1, \dots, acnet_n, Q, W) \qquad (n > 1)$$

such that:

1. $acnet_1, \dots, acnet_n$ are well-formed acyclic nets with disjoint sets of nodes (i.e., places and transitions). We also denote:

$$\begin{array}{rcl} P_{csan} & = & P_{acnet_1} \cup \dots \cup P_{acnet_n} \\ T_{csan} & = & T_{acnet_1} \cup \dots \cup T_{acnet_n} \\ F_{csan} & = & F_{acnet_1} \cup \dots \cup F_{acnet_n} \ . \end{array}$$

[1] $t \neq u \in U$ *means* $t, u \in U$ *and* $t \neq u$

2. Q is a finite set of *buffer places* disjoint from $P_{csan} \cup T_{csan}$.
3. $W \subseteq (Q \times T_{csan}) \cup (T_{csan} \times Q)$ is a set of arcs.
4. For every buffer place q: (i) there is at least one transition t such that tWq; and (ii) if tWq and qWu then transitions t and u belong to different component acyclic nets.

That is, in addition to requiring the disjointness of the component acyclic nets and the buffer places, it is required that buffer places pass tokens between different acyclic nets.
For a node $x \in P_{csan} \cup T_{csan} \cup Q_{csan}$,

$$\begin{array}{rcl} \text{pre}_{csan}(x) & = & \{y \mid yF_{csan}x \vee yWx\} \\ \text{post}_{csan}(x) & = & \{y \mid xF_{csan}y \vee xWy\}\} \end{array}$$

denote the direct predecessors and successors of x, respectively.

The *markings* of *csan*, denoted by markings$(csan)$, are the subsets of $P_{csan} \cup Q_{csan}$. The default *initial* marking is $M^{init}_{csan} = M^{init}_{acnet_1} \cup \cdots \cup M^{init}_{acnet_n}$, and the steps are:

$$\text{steps}(csan) = \{U \in 2^{T_{csan}} \mid \forall t \neq u \in U : \text{pre}_{csan}(t) \cap \text{pre}_{csan}(u) = \varnothing\}\ .$$

A step U of *csan* is *enabled* at marking M if

$$\text{pre}_{csan}(U) \subseteq M \cup (\text{post}_{csan}(U) \cap Q).$$

It can then be *executed* and yield

$$M' = (M \cup \text{post}_{csan}(U) \setminus \text{pre}_{csan}(U).$$

This is denoted by $M[U\rangle_{csan}M'$.

Note that here enabling a step amounts to having all input places belonging to the component acyclic nets present in a global state. Moreover, if an input buffer place is not present, then it must be an output place of a transition belonging to the step. Such a mechanism allows one to synchronise transitions coming from different component acyclic nets. The same mechanism of simultaneous output to and input from a place is not available within the component acyclic nets.

The dynamic behaviour of *csan* is captured by mixed step sequences, as follows. Let $(M^{init}_{csan} =)M_0, M_1 \ldots, M_k$ $(k \geq 0)$ be markings and $U_1, \ldots, U_k$ be steps of *csan* such that we have $M_{i-1}[U_i\rangle_{csan}M_i$, for every $1 \leq i \leq k$. Then

$$\mu = M_0U_1M_1 \ldots M_{k-1}U_kM_k$$

is a *mixed step sequence* of *csan*. This is denoted by $\mu \in \text{mixsseq}(csan)$.

Again, a basic consistency criterion is well-formedness, and *csan* is *well-formed* if the following hold, for every $M_0U_1M_1 \ldots M_{k-1}U_kM_k \in \text{mixsseq}(csan)$:

- $\text{post}_{csan}(t) \cap \text{post}_{csan}(u) = \varnothing$, for every $1 \leq i \leq k$ and all $t \neq u \in U_i$.
- $\text{post}_{csan}(U_i) \cap \text{post}_{csan}(U_j) = \varnothing$, for all $1 \leq i < j \leq k$.

Example 1. *Figure 1 shows a* CSA*-net. Some of its mixed step sequences are:*

$$\begin{array}{rcl} \mu_1 & = & \{p_1,p_7\}\{d\}\{p_2,p_5,p_7\}\{a,b,c\}\{p_3,p_6,p_8\}\{e\}\{p_4,p_8\} \\ \mu_2 & = & \{p_1,p_7\}\{d\}\{p_2,p_5,p_7\}\{a\}\{p_3,q_1,p_5,p_7\}\{b,c\}\{p_3,p_6,p_8\}\{e\}\{p_4,p_8\} \\ \mu_3 & = & \{p_1,p_7\}\{d\}\{p_2,p_5,p_7\}\{a,b\}\{p_3,q_1,p_6,q_2,p_7\}\{e\}\{,q_1,q_2,p_4,p_7\}\{c\}\{p_4,p_8\}. \end{array}$$

3. Towards parameterisation

The concept of parameterisation has gained significant attention in the field of system modelling, as it provides a means to monitor a system's operation under different conditions. Prior to the use of parameterisation, investigating systems under different conditions was a difficult task. In order to address this issue, [9] proposed using the parameterisation technique with coloured Petri nets to represent complex data and events within a system. This allowed for the investigation of systems influenced by parameters that determine the outputs. This technique has been applied to model systems in various fields, including networking. For example, [10] used parameterisation to model switched Ethernet networks, providing a means to investigate system performance by providing reusable networks for testing different parameters.

Addressing large scale problems, characterised by a large amount of data, as one unit is a difficult task and makes the model under investigation hard to interpret and visualise. So, the one proposed solution is to tackle such a problem by dividing it into smaller units that can later be put together to form the entire system. In the meantime, modelling is needed to better understand larger systems. This is because modelling reduces debugging time and allows the reuse of frequently used components (like events and conditions) which, in turn, saves time and effort. Moreover, modelling assures the validity of the system as modules that have been extensively tested in one environment are likely to run in a similar environment as expected, so reuse increases the reliability of accuracy. This is known as parameterisation concept [11]. That is, the parameterisation concept has been applied to Petri nets where a part of the net remains unspecified until it is executed. Specifically, there was a need to parameterise a part of the net and find a way to substitute it in the net. This is achieved by hiding part of the system [11].

More recently, the modeling of biological systems has become a challenging problem due to the large-scale data involved. As a result, Petri nets have been used to model biological systems. This is accomplished by representing a group of similarly behaving components as one component, with each of these components being represented by a specific color, thereby distinguishing them. This provides a compact representation of a complex system while maintaining its semantic meaning. Additionally, the colored Petri net has another significant advantage, as adding a new component is an easy task by simply adding a new color. With these benefits in mind, colored Petri nets have been successfully employed in discriminating metabolites, modeling multicellular systems, and analyzing spatial diffusion in biological systems [12, 13].

Another direction of applying colored Petri nets is to focus on concurrent systems. However, concurrent systems often involve a vast amount of highly detailed data, which can be overwhelming. Moreover, every single step is usually described in detail, which can be distracting for the modeler. To address this, animation graphics have been employed for visualization, which provides more appropriate feedback [9, 14].

3.1. Master buffer places

This section will extend the work done in [8] by adding the formalisation of the *Master Buffer Places (*MBP*)* concept. CSA-nets consist of sets of acyclic nets that communicate with each other through a set of buffer places. Such relations can generate a large number of buffer places, which can make the model difficult to visualise and analyse, especially for large CSA-nets. Therefore,

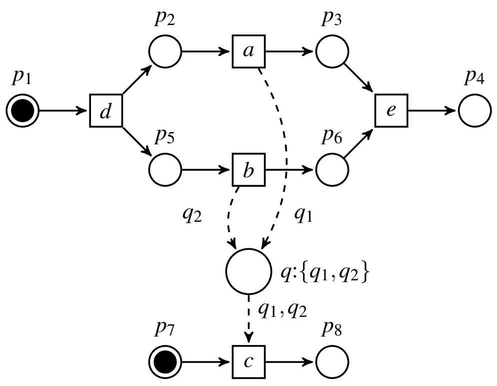
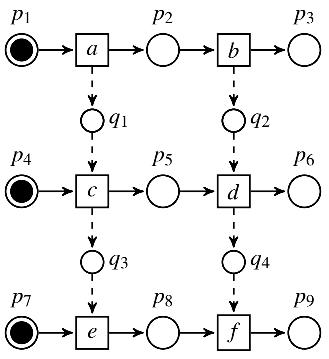


Figure 2: A version of the CSA-net of Figure 1 with a master buffer place.



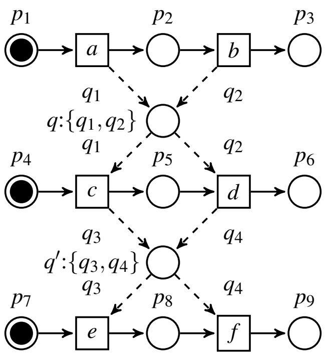


Figure 3: A CSA-net and its version with two master buffer places.

we propose to introduce Master Buffer Places (MBPs) to provide conciseness to the CSA-net by collapsing (folding) buffer places into (MBPs). In other words, we propose to allow buffer places to represent more than one token at a time to avoid having a large number of distracting buffer places. Additionally, it will allow the component acyclic nets to communicate through a unique buffer place. Inside a (MBP), there will be a set of tokens represented by unique colors. A specific token will appear in the (MBP) without conflicting with other tokens in each execution since it is color-safe, given that the original CSA-net was safe. This contribution will enhance CSA-net visualisation and contribute to a more readable and understandable model. Figure 2 depicts the CSA-net of Figure 1, in which the two buffer places (q_1 and q_2) were folded into a single place q. It is important to emphasize that the annotations are sets of the original buffer places. That is, the execution rules follow the standard idea of coloured Petri nets (CPNs), which ensure that the tokens ’flowing’ along the arcs match the annotations (the tokens in (MBPs) are simply the original buffer places).

Figure 3 presents another example of using master buffer places. The image shows a well-structured display, which avoids any crossing of the arcs adjacent to the original buffer places. This improves the representation of the CSA-net. Introducing a single master buffer place, however, may lead to unwanted crossing of arcs, as illustrated in Figure 3. If a single master buffer place is added to the CSA-net in Figure 3, the crossing would be unavoidable unless placed ‘outside’ the outline of the underlying occurrence nets, which is generally unacceptable, since the individual acyclic nets would be too large to fit in a single screen. One way to address this issue is to generate multiple master buffer places, as shown in Figure 3. This raises an interesting problem of determining the number and placement of the master buffer places automatically or semi-automatically to optimize the visualization of CSA-nets. This problem is the subject of our ongoing study.

3.2. Parameterised CSA-nets

With the above in mind, we are able to improve the visualization of CSA-nets by folding the set of buffer places into master buffer places. In this section, we will use the same concept of folding to fold the components of CSA-nets while preserving their main rules (as being acyclic). For example, Figure 4 depicts a parameterised version of a very simple acyclic nets where two concurrent parts are folded (collapsed) into a single parameterised structure. Clearly, the two parts have the same net structure. Thus, the idea is to determine the set of components that are behaving identically, and then represent them as a single substructure. This uses typed parameters to achieve the desired effect through passing tokens to parameterised transition. That is, this process can allow for the reuse of the model multiple times and increase comprehension, making larger systems under investigation easier to handle. (Formally, $\iota(r_1) = \{p_1, p_3\}$, $\iota(t_1) = \{a, b\}$ and $\iota(r_2) = \{p_2, p_4\}$.) This extension can have positive effect on the model in both increasing the abstraction and visualisation of the model under investigation.

4. Formalisation

In this section, we provide a full formalisation of the approach proposed in this paper. We first introduce parameterised acyclic nets, and then parameterised communication structured acyclic nets.

4.1. Parameterised acyclic nets

We first introduce a parameterised version of acyclic nets.

Definition 1 (parameterised acyclic net)**.** *A* parameterised acyclic net (or PA-net) *is a tuple* $pacnet = (P, T, F, acnet, \iota, col)$*, where:*

- $acnet = (P', T', F')$ *is a well-formed acyclic net, and col is a finite set of* colours *(or* parameters*).*
- P *(*$P \neq \varnothing$*) and* T *are disjoint finite sets of* places *and* transitions*, respectively.*
- $F \subseteq (P \times T) \cup (T \times P)$ *is a* flow relation.
- $\iota : P \cup T \to 2^{P' \cup T'} \setminus \{\varnothing\}$ *is a* node annotation *mapping.*

We denote, for all $x \in P \cup T$*,* $p \in P'$*, and* $u \in T'$*:*

$$\begin{array}{lcllcl} \mathrm{pre}_{pacnet}(x) & = & \{z \mid zFx\} & \mathrm{pre}_{pacnet}(u,p) & = & \iota(p) \cap \mathrm{pre}_{acnet}(u) \\ \mathrm{post}_{pacnet}(x) & = & \{z \mid xFz\} & \mathrm{post}_{pacnet}(u,p) & = & \iota(p) \cap \mathrm{post}_{acnet}(u)\,. \end{array}$$

We then assume that the following hold.

1. $(P \cup T) \cap (P' \cup T') = \varnothing$*,* $\bigcup \iota(P) = P'$*, and* $\bigcup \iota(T) = T'$*.*
2. *For all* $t \in T$ *and* $u \in \iota(t)$ *we have the following, where* $\biguplus$ *is assumed to be applied to non-empty sets:*[2]
$$\begin{array}{lcl} \mathrm{pre}_{acnet}(u) & = & \biguplus\{\mathrm{pre}_{pacnet}(u,p) \mid p \in \mathrm{pre}_{pacnet}(t)\} \\ \mathrm{post}_{acnet}(u) & = & \biguplus\{\mathrm{post}_{pacnet}(u,p) \mid p \in \mathrm{post}_{pacnet}(t)\}\,. \end{array}$$
3. *If* $u \in T'$ *and* $R \subseteq P$ *are such that*
$$\mathrm{pre}_{acnet}(u) = \biguplus\{\mathrm{pre}_{pacnet}(u,p) \mid p \in R\}$$
then there is $t \in T$ *such that* $u \in \iota(t)$ *and* $\mathrm{pre}_{pacnet}(t) = R$*.*
4. *For every* $p \in P$*,*
$$\iota(p) \setminus M^{init}_{acnet} = \bigcup\{\mathrm{post}_{pacnet}(u,p) \mid\ u \in \bigcup \iota(\mathrm{pre}_{pacnet}(p))\}\,.$$
5. *For every* $p \in P^{init}_{acnet}$*, there is exactly one place* $p^{init} \in P$ *such that* $p \in \iota(p^{init})$*.*

Note that we do not assume that F is acyclic since the acyclity of the behaviour of *acnet* will lead to the acyclicity of the behaviour of *pacnet*.

Example 2. *Figure 4 shows a parameterised acyclic net*

$$pacnet = (P, T, F, acnet, \iota, col)$$

such that

$$acnet = (\{p_1, p_2, p_3, p_4\}, \{a, b\}, \{(p_1, a), (a, p_2), (p_3, b), (b, p_4)\}),$$

and

$$\begin{array}{lcllcllcl} P & = & \{r_1, r_2\} & T & = & \{t\} & F & = & \{(r_1, t), (t, r_2)\} \\ \iota(r_1) & = & \{p_1, p_3\} & \iota(t) & = & \{a, b\} & \iota(r_2) & = & \{p_2, p_4\} \\ col & = & \{1, 2\}. & & & & & & \end{array}$$

We then have, for example, $\mathrm{pre}_{pacnet}(a, r_1) = \{p_1\}$ *and* $\mathrm{post}_{pacnet}(b, r_2) = \{p_4\}$*.*

[2]Thus, for example, $\{\mathrm{pre}_{pacnet}(u,p) \mid p \in \mathrm{pre}_{pacnet}(t)\}$ is a partition of $\mathrm{pre}_{acnet}(u)$.

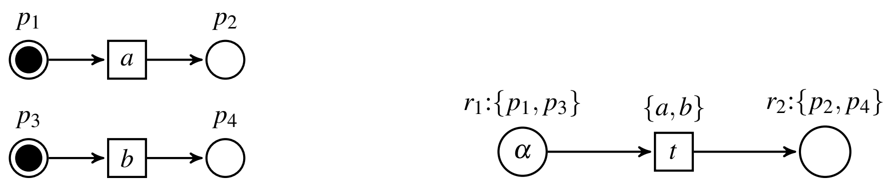


Figure 4: An acyclic net and its parameterised version with $col = \{1,2\}$, where $\alpha = \{p_1{:}1, p_1{:}2, p_3{:}1, p_3{:}2\}$.

Intuitively, $\iota(t)$ specifies different 'modes' of the execution of a transition $t \in T$, and $\iota(p)$ specifies the places of *acnet* which can be 'present' in $p \in P$ in the executions of *pacnet*. Definition 1(2) means that, for each mode $u \in \iota(t)$, the arcs incoming to t 'deliver' the input places of u, and similarly for the output arcs. Definition 1(3) means that, for any potential distribution of the 'preset' of $u \in T_{acnet}$, there is a transition $t \in T$ with the mode u which can input this distribution of the preset. Definition 1(4) states that $\iota(p)$ is determined by the modes of the input transitions.

To capture the behaviour of *pacnet*, we will use parameterised places and transitions:

$$\begin{array}{rcl} P_{pacnet} & = & \{(p,r,c) \in P' \times P \times col \mid p \in \iota(r)\} \\ T_{pacnet} & = & \{(u,t,c) \in T' \times T \times col \mid u \in \iota(t)\} \end{array}$$

Intuitively, (p,r,c) is an instance of place p of the original acyclic net coloured by c, which resides in the place r of *pacnet*. (Another way of interpreting (p,r,c) is that it represents coloured token (p,c) present in the place r.) Similarly, (u,t,c) represents mode in which transition t operates as if it were transition u of the original acyclic net coloured by c. Moreover, we denote for $(u,t,c) \in T_{pacnet}$ (and similarly for subsets of T_{pacnet}):

$$\begin{array}{rcl} \text{pre}_{pacnet}((u,t,c)) & = & \bigcup\{\text{pre}_{acnet}(u,p) \times \{p\} \times \{c\} \mid p \in \text{pre}_{pacnet}(t) \wedge c \in col\} \\ \text{post}_{pacnet}((u,t,c)) & = & \bigcup\{\text{post}_{acnet}(u,p) \times \{p\} \times \{c\} \mid p \in \text{post}_{pacnet}(t) \wedge c \in col\}\ . \end{array}$$

Intuitively, $\text{pre}_{pacnet}((u,t,c))$ and $\text{post}_{pacnet}((u,t,c))$ determine the tokens which transition t in mode u consumes and produces when fired.

Example 3. *For the* PAN*-net in Figure 4 we have, for example,* $\text{pre}_{pacnet}((a,t,1)) = \{(p_1,r_1,1)\}$ *and* $\text{pre}_{pacnet}((b,t,2)) = \{(p_4,r_2,2)\}$.

The *markings* of *pacnet*, denoted by markings(*pacnet*), are the subsets of P_{pacnet}. The default *initial* marking is:

$$M^{init}_{pacnet} = \{(p,p^{init},c) \mid p \in M^{init}_{acnet} \wedge c \in col\}. \tag{1}$$

Given a marking M and $r \in P$, we denote

$$M(r) = \{(p,c) \mid (p,r,c) \in M\} = \{(p_1,c_1),\ldots,(p_k,c_k)\},$$

and in the diagrams place $p_1{:}c_1,\ldots,p_k{:}c_k$ inside p. In other words, $p_1{:}c_1,\ldots,p_k{:}c_k$ are coloured tokens present in p at marking M. Moreover, each place p of *pacnet* is annotated by $p{:}\iota(p)$, and each transition t by $\iota(t)$. The same convention is used for the channel places and CSA-nets later on. We also say that M is *colour-safe* if $M(r) \cap M(s) = \varnothing$, for all $r \neq s \in P$. Note that M^{init}_{pacnet} is colour-safe.

The *steps* of *pacnet*, denoted by steps(*pacnet*), are:

$$\text{steps}(pacnet) = \{U \in 2^{T_{pacnet}} \mid \forall t \neq u \in U : \text{pre}_{pacnet}(t) \cap \text{pre}_{pacnet}(u) = \varnothing\} \ .$$

A step U of *pacnet* is *enabled* at marking M if $\text{pre}_{pacnet}(U) \subseteq M$. It can then be *executed* and yield

$$M' = (M \cup \text{post}_{pacnet}(U)) \setminus \text{pre}_{pacnet}(U).$$

This is denoted by $M[U\rangle_{pacnet} M'$. Let $(M^{init}_{pacnet} =) M_0, M_1 \dots, M_k$ $(k \geq 0)$ be markings and $U_1, \dots, U_k$ be steps of *pacnet* such that we have $M_{i-1}[U_i\rangle_{pacnet} M_i$, for every $1 \leq i \leq k$. Then

$$\mu = M_0 U_1 M_1 \dots M_{k-1} U_k M_k$$

is a *mixed step sequence* of *pacnet*. This is denoted by $\mu \in \text{mixsseq}(pacnet)$. Moreover, μ is *well-formed* if the following hold:

- $\text{post}_{pacnet}(t) \cap \text{post}_{pacnet}(u) = \varnothing$, for every $1 \leq i \leq k$ and all $t \neq u \in U_i$.
- $\text{post}_{pacnet}(U_i) \cap \text{post}_{pacnet}(U_j) = \varnothing$, for all $1 \leq i < j \leq k$.

Note that in such a case M_i is colour-safe, for every $0 \leq i \leq k$.

Example 4. *The initial marking of the* PA*-net Figure 4 is*

$$M^{init}_{pacnet} = \{(p_1, r_1, 1), (p_1, r_1, 2), (p_3, r_1, 1), (p_3, r_1, 2)\}.$$

Two of its mixed step sequences are:

$$\begin{array}{rcl} \mu_1 & = & M^{init}_{pacnet} \quad \{(a,t,2)\} \quad M_1 \ \{(b,t,2),(b,t,1)\} \ M_2 \ \{(a,t,1)\} \ M_3 \\ \mu_2 & = & M^{init}_{pacnet} \ \{(b,t,1),(a,t,2)\} \ M_4 \ \{(a,t,1),(b,t,2)\} \ M_3, \end{array}$$

where:

$$\begin{array}{rcl} M_1 & = & \{(p_1,r_1,1),(p_1,r_2,2),(p_3,r_1,1),(p_3,r_1,2)\} \\ M_2 & = & \{(p_1,r_1,1),(p_1,r_2,2),(p_3,r_2,1),(p_3,r_2,2)\} \\ M_3 & = & \{(p_1,r_2,1),(p_1,r_2,2),(p_3,r_2,1),(p_3,r_2,2)\} \\ M_4 & = & \{(p_1,r_1,1),(p_1,r_2,2),(p_3,r_1,1),(p_3,r_2,2)\}. \end{array}$$

To express full consistency between the behaviour of *pacnet* and that of the underlying acyclic net *acnet*, we need a notion of projection of the steps and markings of *pacnet* onto the steps and markings of *acnet*.
For every subset X of $P_{pacnet} \cup T_{pacnet}$ and $c \in col$, $X \downarrow c = \{x \mid (x,y,c) \in X\}$. Moreover, for every sequence $\xi = X_1 \dots X_k$ of subsets of $P_{pacnet} \cup T_{pacnet}$, we denote $\xi \downarrow c = X_1 \downarrow c \dots X_k \downarrow c$.

Example 5. *Let μ_1 be as in Example 4. Then*

$$\begin{array}{rcl} \mu_1 \downarrow 1 & = & \{p_1,p_3\} \ \ \varnothing \ \ \{p_1,p_3\} \ \{b\} \ \{p_1,p_4\} \ \{a\} \ \{p_2,p_4\} \\ \mu_1 \downarrow 2 & = & \{p_1,p_3\} \ \{a\} \ \{p_2,p_3\} \ \{b\} \ \{p_2,p_4\} \ \{a\} \ \{p_2,p_4\}. \end{array}$$

Note that both $\mu_1 \downarrow 1$ and $\mu_1 \downarrow 2$ are mixed step sequences of the original acyclic net.

Proposition 1. *Let pacnet be as in Definition 1.*

1. *If* $\mu \in \mathrm{mixsseq}(pacnet)$ *then* μ *is well-formed and* $\mu \downarrow c \in \mathrm{mixsseq}(acnet)$*, for every* $c \in col$.
2. *If* $\mu^c \in \mathrm{mixsseq}(acnet)$*, for every* $c \in col$*, are sequences of the same length, then there is* $\mu \in \mathrm{mixsseq}(pacnet)$ *such that* $\mu \downarrow c = \mu^c$*, for every* $c \in col$.

Proof. (1) Let $\mu = M_0 U_1 M_1 \dots M_{k-1} U_k M_k$ and $c \in col$. The result is proven by induction on k.

If $k = 0$ then $\mu = M_0 = M^{init}_{pacnet}$. Then μ is well-formed and $\mu \downarrow c \in \mathrm{mixsseq}(acnet)$ follow from Definition 1(5) and Eq. (1).

The inductive case follows from the well-formedness of *acnet*, Definition 1(3), and, for every $(u,t,c) \in T_{pacnet}$, $\mathrm{pre}_{pacnet}((u,t,c)) \downarrow c = \mathrm{pre}_{acnet}(u)$ and $\mathrm{post}_{pacnet}((u,t,c)) \downarrow c = \mathrm{post}_{acnet}(u)$. Also, it is important that the tokens an modes of firing in different colours do not interfere with each other.

(2) Follows by a straightforward induction on the common length of the mixed step sequences. Again, it is important that the tokens an modes of firing in different colours do not interfere with each other. Moreover, Definition 1(3) ensures that a $u \in T'$ can be 'executed' with parameter $c \in col$ when M is a colour-safe marking of *pacnet* such that $\mathrm{pre}_{acnet}(u) \subseteq M \downarrow c$. □

The assumption that the mixed step sequences have the same length can be easily satisfied by adding extra empty steps to the 'shorter' ones.

Proposition 1(1) means that the behaviour of *pacnet* is well-formed and based on the behaviour of *acnet*. Conversely, Proposition 1(2) means that all the behaviours of *acnet* are represented by the behaviours of *pacnet*.

4.2. Parameterised communication structured acyclic net

We now introduce a parameterised version of communication structured acyclic nets. Since the definitions closely follow those in the previous section, we keep explanations short.

Definition 2. *A* parameterised communication structured acyclic net (or PCSA-net) *is a tuple* $pcsan = (pacnet_1, \dots, pacnet_n, Q, W, Q', W', \iota', col)$ $(n \geq 1)$ *such that:*

- $pacnet_i = (P_i, T_i, F_i, acnet_i, \iota_i, col)$*, for* $1 \leq i \leq n$*, are parameterised acyclic nets.*
- $csan = (acnet_1, \dots, acnet_n, Q, W)$ *is a well-formed csa-net.*
- Q' *is a set of* (master) buffer *places, and* $W' \subseteq Q' \times T \cup T \times Q'$*, where* $T = T_1 \cup \dots \cup T_n$*, is a set of arcs adjacent to these places.*
- $\iota' : Q' \to 2^Q \setminus \{\varnothing\}$ *is an annotation mapping such that* $Q = \biguplus\{\iota(q) \mid q \in Q'\}$. *For every* $q \in Q$*,* q_{pcsan} *will denote the unique channel place in* Q' *such that* $q \in \iota'(q_{pcsan})$.

We also assume that the sets of nodes of $pacnet_1, \dots, pacnet_n, Q, Q'$ *are pairwise disjoint and*

$$W' = \{(x,y) \in Q' \times T \cup T \times Q' \mid (\iota(x) \times \iota(y)) \cap W \neq \varnothing\},$$

where $\iota = \iota_1 \cup \dots \cup \iota_n \cup \iota'$.

Example 6. *Figure 5 shows an example of* PCSA*-net.*

We also denote $P = P_1 \cup \cdots \cup P_n \cup Q'$ and $F = F_1 \cup \cdots \cup F_n \cup W'$, and, for all $x \in P \cup T$, $p \in P$, and $u \in T_{csan}$, we denote:

$$\begin{array}{rclrcl} \text{pre}_{pcsan}(x) & = & \{z \mid zFx\} & \text{pre}_{pcsan}(u,p) & = & \iota(p) \cap \text{pre}_{csan}(u) \\ \text{post}_{pcsan}(x) & = & \{z \mid xFz\} & \text{post}_{pcsan}(u,p) & = & \iota(p) \cap \text{post}_{csan}(u) \end{array}$$

Proposition 2. *For all $t \in T_{pcsan}$ and $u \in \iota(t)$:*

$$\begin{array}{rcl} \text{pre}_{csan}(u) & = & \biguplus\{\text{pre}_{pcsan}(u,p) \mid p \in \text{pre}_{pcsan}(t)\} \\ \text{post}_{csan}(u) & = & \biguplus\{\text{post}_{pcsan}(u,p) \mid p \in \text{post}_{pcsan}(t)\} \end{array}$$

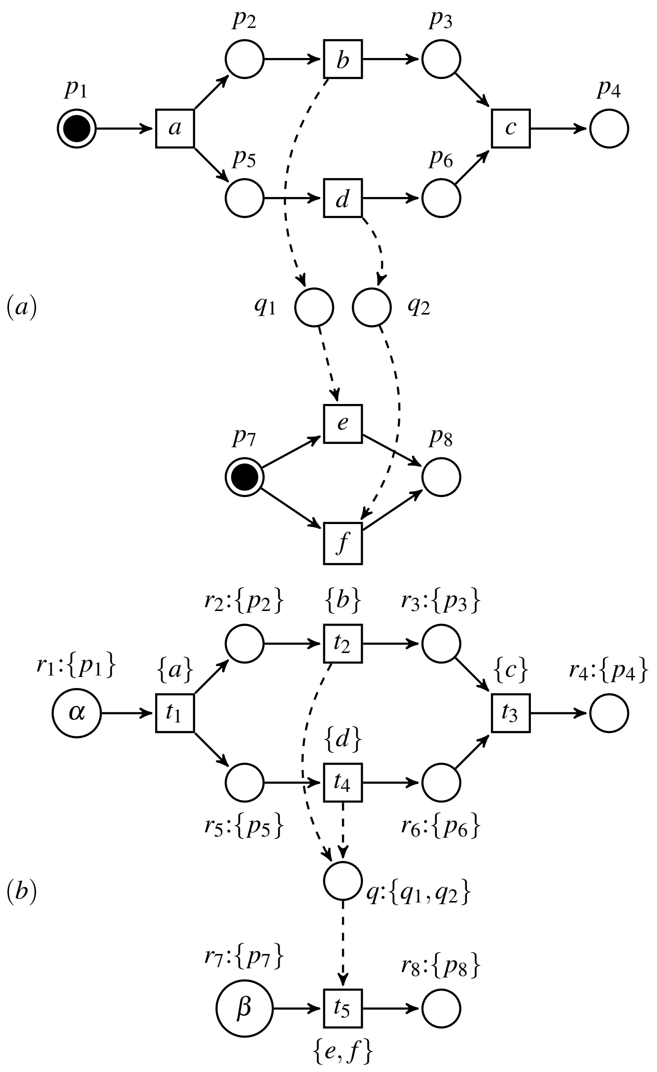


Figure 5: A CSA-net after introducing a master buffer place (a); and folding the lower acyclic net with $col = \{1,2\}$, where $\alpha = \{p_1{:}1, p_1{:}2\}$ and $\beta = \{p_7{:}1, p_7{:}2\}$ (b).

The idea of Definition 2 is similar to that of parameterised acyclic net, and so the following notations are close to those introduced before :

$$T_{pcsan} = T_{pacnet_1} \cup \cdots \cup T_{pacnet_1} \;\; and \;\; P_{pcsan} = P_{pacnet_1} \cup \cdots \cup P_{pacnet_1} \cup \widetilde{Q}$$

where $\widetilde{Q} = \{(q, q_{pcsan}, c) \mid q \in Q \wedge c \in col\}$. Moreover, we denote, for all $1 \leq i \leq n$ and $(u,t,c) \in T_{pacnet_i}$:

$$\begin{array}{lcl} \text{pre}_{pcsan}((u,t,c)) & = & \text{pre}_{pacnet_i}((u,t,c)) \;\cup \{(q, q_{pcsan}, c) \in \widetilde{Q} \mid qWu\} \\ \text{post}_{pcsan}((u,t,c)) & = & \text{post}_{pacnet_i}((u,t,c)) \cup \{(q, q_{pcsan}, c) \in \widetilde{Q} \mid uWq\} \end{array}$$

The *markings* of *pcsan*, denoted by markings$(pcsan)$, are the subsets of P_{pcsan}. The default *initial* marking is

$$M^{init}_{pcsan} = M^{init}_{pacnet_1} \cup \cdots \cup M^{init}_{pacnet_n}$$

and the steps are

$$\text{steps}(pcsan) = \{U \in 2^{T_{pcsan}} \mid \forall t \neq u \in U : \text{pre}_{pcsan}(t) \cap \text{pre}_{pcsan}(u) = \varnothing\} \,.$$

A step U of *pcsan* is *enabled* at marking M if

$$\text{pre}_{pcsan}(U) \subseteq M \cup (\text{post}_{pcsan}(U) \cap \widetilde{Q}).$$

It can then be *executed* and yield

$$M' = (M \cup \text{post}_{pcsan}(U) \setminus \text{pre}_{pcsan}(U).$$

This is denoted by $M[U\rangle_{pcsan}M'$. Let $(M^{init}_{pcsan} =)M_0, M_1 \ldots, M_k$ $(k \geq 0)$ be markings and $U_1, \ldots, U_k$ be steps of *pcsan* such that we have $M_{i-1}[U_i\rangle_{pcsan}M_i$, for every $1 \leq i \leq k$. Then

$$\mu = M_0 U_1 M_1 \ldots M_{k-1} U_k M_k$$

is a *mixed step sequence* of *pcsan*. This is denoted by $\mu \in$ mixsseq$(pcsan)$. Moreover, μ is *well-formed* if the following hold:

- $\text{post}_{pcsan}(t) \cap \text{post}_{pcsan}(u) = \varnothing$, for every $1 \leq i \leq k$ and all $t \neq u \in U_i$.
- $\text{post}_{pcsan}(U_i) \cap \text{post}_{pcsan}(U_j) = \varnothing$, for all $1 \leq i < j \leq k$.

Note that in such a case M_i is colour-safe, for every $0 \leq i \leq k$.

We also use the same notion of projection as for parameterised acyclic nets, and then obtain a result showing a full consistency between the behaviour of a PCSA-net and its underlying CSA-net.

Proposition 3. *Let pcsan be as in Definition 2.*

1. *If* $\mu \in$ mixsseq$(pcsan)$ *then* μ *is well-formed and* $\mu \downarrow c \in$ mixsseq$(csan)$*, for every* $c \in col$.
2. *If* $\mu^c \in$ mixsseq$(csan)$*, for every* $c \in col$*, are sequences of the same length, then there is* $\mu \in$ mixsseq$(pcsan)$ *such that* $\mu \downarrow c = \mu^c$*, for every* $c \in col$.

Proof. The proof is similar to that of Proposition 1. □

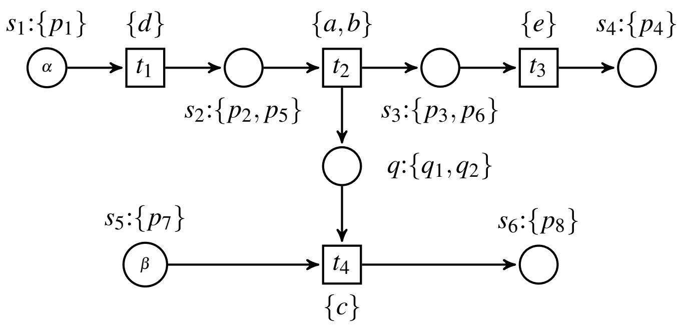


Figure 6: A parameterised version of the CSA-net in Figure 1 with $col = \{1,2\}$, where $\alpha = \{p_1{:}1, p_1{:}2\}$ and $\beta = \{p_7{:}1, p_7{:}2\}$.

Example 7. *Figure 5(a) represents the original model in where two acyclic nets (the upper one exhibits concurrency and the lower exhibits alternative) communicate through two buffer places. Figure 5(b) shows the corresponding representation of Figure 5(a) after parameterising. That is,* $\iota(r_1) = \{p_1\}$ *and* $\iota(t_1) = \{a\}$ *in the upper acyclic net, while* $\iota(r_7) = \{p_7\}$ *and* $\iota(t_5) = \{e, f\}$ *in the lower acyclic net. In other words,* $\iota(t_5)$ *specifies different 'modes' of the execution of transitions e and f for each mode* $u \in \iota(t_5)$*, the arcs incoming to transition* $\{t_5\}$ *'deliver' the input places of u, and similarly for the output arc.*

Example 8. *Figure 6 depicts a parameterised version of* CSA*-net of Figure1 after folding both acyclic nets. More specifically,* s_2 *parameterised places* p_2 *and* p_5*, while* t_2 *parameterised transition a and b. In other words,* $\iota(s_2)$ *specifies the places which can be 'present' in* s_2 *in the executions of pacnet. This process allows for both places and transitions to accept more than one tokens (parameters) which can enhance both visualisation and analysis of the models under investigation.*

5. Conclusion

This paper introduced two extensions for CSA-nets using the concepts of folding and parameterisation. The analysis and visualisation of CSA-nets is not an easy task, especially for large and complex systems. Thus, this paper proposed the parameterisation technique by collapsing original components of CSA-nets to parameteric ones. This technique aims to extend both conditions and events to accept more than one token to be distinguished using the passed parameter. This can contribute to improving visualisation and increase the ability to understand models under investigation.

Future work will consider other form of folding, e.g., merging together entire components of acyclic nets and subsequently the extension to behaviour structured acyclic nets. We will implement the aforementioned extensions in SONCRAFT. In addition, the extension work will consider the behaviour structured acyclic nets. Both directions of future work aim to be implemented in SONCRAFT. Moreover, we will apply these concepts and extension to cybercrime scenarios to analyse and visualise such systems by CSA-nets.

Acknowledgement

I would like to thank the reviewers for their insightful comments on the submitted paper.

References

[1] D. Probst, J.-L. Reymond, Visualization of very large high-dimensional data sets as minimum spanning trees, Journal of cheminformatics 12 (2020) 1–13.

[2] P. Chen, Visualization of real-time monitoring datagraphic of urban environmental quality, Eurasip Journal on Image and Video Processing 2019 (2019) 1–9.

[3] M. Koutny, B. Randell, Structured occurrence nets: A formalism for aiding system failure prevention and analysis techniques, Fundamenta Informaticae 97 (2009) 41–91.

[4] B. Randell, Occurrence nets then and now: the path to structured occurrence nets, in: International Conference on Application and Theory of Petri Nets and Concurrency, Springer, 2011, pp. 1–16.

[5] T. Alshammari, Towards automatic extraction of events for SON modelling, in: PNSE@Petri Nets, volume 3170 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 188–201.

[6] N. Almutairi, M. Koutny, Verification of communication structured acyclic nets using SAT, in: PNSE@Petri Nets, volume 2907 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 175–194.

[7] N. Almutairi, Probabilistic communication structured acyclic nets, in: PNSE@Petri Nets, volume 3170 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 168–187.

[8] M. Alahmadi, Master channel places for communication structured acyclic nets., in: PNSE@ Petri Nets, 2021, pp. 233–240.

[9] K. Jensen, L. M. Kristensen, L. Wells, Coloured Petri nets and CPN tools for modelling and validation of concurrent systems, International Journal on Software Tools for Technology Transfer 9 (2007) 213–254.

[10] Z. Dmitry A, S. Tatiana R, A parametric colored Petri net model of a switched network, International journal of Communications, Network and System Sciences 2011 (2011).

[11] T. Mailund, Parameterised coloured Petri nets, in: Proceedings of the Workshop on Practical Use of Coloured Petri Nets and Design/CPN, Department of Computer Science, University of Aarhus, 1999.

[12] F. Liu, M. Heiner, D. Gilbert, Coloured Petri nets for multilevel, multiscale and multidimensional modelling of biological systems, Briefings in bioinformatics 20 (2019) 877–886.

[13] K. Jensen, Coloured Petri nets: basic concepts, analysis methods and practical use, volume 1, Springer Science & Business Media, 1996.

[14] K. Jensen, An introduction to the practical use of coloured petri nets, Lectures on Petri Nets II: Applications: Advances in Petri Nets 3 (1998) 237–292.