DOI: 10.15514/ISPRAS-2025-37(4)-18

Evaluating Structural Complexity of Workflow Nets Modeling Asynchronous Agent Interactions

E. Zemlyanoy, ORCID: 0009-0007-3452-1698 eozemlyanoy@edu.hse.ru
R. Nesterov, ORCID: 0000-0002-4162-9070 rnesterov@hse.ru

HSE University,
11 Pokrovsky Bulvar, Moscow, 101000, Russia.

Abstract. The structure of a process model discovered from an event log of a multi-agent system often does not reflect the system architecture with respect to agent interactions. The existing conformance checking quality dimensions mainly evaluate the extent to which the behavior a discovered model corresponds to event sequences recorded in an event log. These behavioral dimensions might be insufficient to differentiate process models discovered from an event log of the same multi-agent system with respect to the independence of agents and the complexity of their interactions. In this work, we propose a theoretically grounded approach to measuring the structural complexity of a process model representing a multi-agent system with asynchronously interacting agents. We also report the key outcomes from a series of experiments to evaluate the sensitivity of the proposed approach to structural modifications in process models.

Keywords: multi-agent systems; asynchronous interaction; process mining; Petri nets; workflow nets; structural complexity.

For citation: Zemlyanoy E., Nesterov R. Evaluating Structural Complexity of Workflow Nets Modeling Asynchronous Agent Interactions. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 4, part 2, 2025, pp. 47-68. DOI: 10.15514/ISPRAS-2025-37(4)-18

Acknowledgements. This work was supported by the Basic Research Program at the HSE University, Moscow (Russia).

Анализ структурной сложности сетей потоков работ для моделирования асинхронного взаимодействия агентов

Е.О. Земляной, ORCID: 0009-0007-3452-1698 <eozemlyanoy@edu.hse.ru>
Р.А. Нестеров, ORCID: 0000-0002-4162-9070 <rnesterov@hse.ru>
Национальный исследовательский университет «Высшая школа экономики», Россия, 101000, Москва, Покровский бульвар, д. 11.

Аннотация. Структура модели процесса, полученной на основе журнала событий многоагентной системы, часто не отражает архитектуру системы с точки зрения взаимодействий между агентами. Существующие метрики проверки соответствия в основном оценивают степень соответствия поведения обнаруженной модели последовательностям событий, зафиксированным в журнале. Однако такие поведенческие метрики могут оказаться недостаточными для того, чтобы отличить модели процессов, извлеченные из журнала одной и той же многоагентной системы, с учетом степени независимости агентов и сложности их взаимодействий. В данной работе предлагается теоретически обоснованный подход к измерению структурной сложности модели процесса, представляющей многоагентную систему с асинхронным взаимодействием агентов. Также представлены ключевые результаты серии экспериментов, направленных на оценку чувствительности предложенного подхода к структурным изменениям в моделях процессов.

Ключевые слова: многоагентные системы; асинхронное взаимодействие; синтез моделей процессов; сети Петри; сети потоков работ; структурная сложность.

Для цитирования: Земляной Е.О., Нестеров Р.А. Анализ структурной сложности сетей потоков работ для моделирования асинхронного взаимодействия агентов. Труды ИСП РАН, том 37, вып. 4, часть 2, 2025 г., стр. 47–68 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(4)–18.

Благодарности: Работа выполнена при поддержке Программы фундаментальных исследований НИУ ВШЭ, Москва.

1. Introduction

Modern information systems perform a wide variety of operations recorded in event logs that help to track system performance, as well as to investigate and eliminate unwanted behavior. Event logs consist of ordered sequences of events called traces. Event logs are used in process mining to discover real process models with the help of various automated techniques [1-2]. Discovered process models reflect behavioral deviations from the reference models prepared at the early stages of the information system life-cycle. These deviations can be caused by different reasons, including, for example, infrastructural changes or unexpected user behavior.

Process models discovered from event logs can be represented in numerous notations: Petri nets, heuristic nets, causal nets, or Business Process Model and Notation (BPMN). In our work, we use Petri nets [3] – a well-known formalism for modeling the process control-flow which represents causal relations among actions recorded in an event log. In addition, Petri nets can serve as the mathematical basis for practice- oriented modeling notations, such as BPMN [4].

A wide variety of process discovery algorithms allow us to discover different process models from the same event log. In this case, the problem of selecting the most appropriate model arises. This is especially acute for models discovered from event logs of multi-agent systems with asynchronously inter- acting agents, as we show by the following example. Graph-theoretic characteristics, such as density, miss the architectural aspects of a discovered model, as they do not use the fact that the model structure should represent agent interactions.

Conformance checking [5] behavioral dimensions, fitness and precision, may also not be of great use for differentiating process models discovered from the event logs of a multi- agent system. Most of process discovery algorithms seek for the perfect fitness, when a discovered model can execute all traces in an event log, and for the high precision, when the behavior of a model is narrowed to the traces in an event log.

Consider the following example. Figure 1a shows the Petri net model for a multi-agent system with two agents exchanging messages via two channels. This model perfectly fits the trace given in Table 1, as it can be fully replayed. Precision of this Petri net is 0.85 [6]. Figure 1b shows the other Petri net perfectly fitting the event log in Table 1, and its precision is also perfect, since it can execute only this trace. Such a Peti net can be discovered using, for example, Inductive miner [7].

What is more important, the structure of the Petri net in Fig. 1b hides the architecture of a multi-agent system with two interacting agents – they cannot be fully identified as subnets. In other words, the structure of the Petri net in Fig. 1a is more complicated than the of the Petri net in Fig. 1b with respect to the independence of agents.

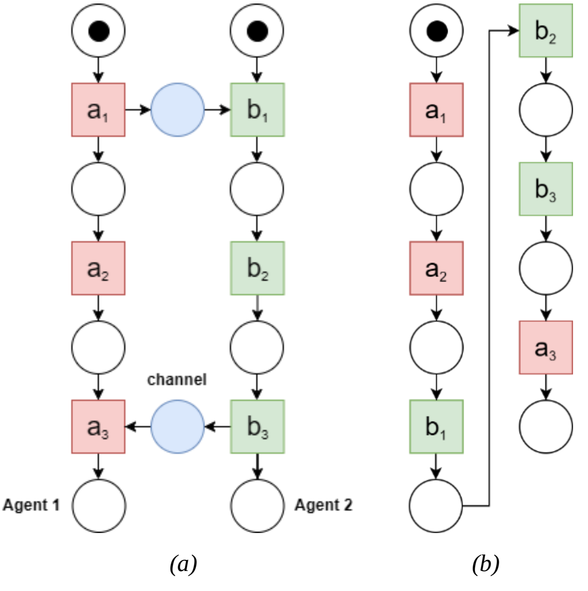


Fig. 1. Two models of the same multi-agent system.

This example highlights the problem that structural and behavioral conformance checking dimensions might be insufficient to evaluate the structural complexity of process models discovered from event logs of multi-agent systems. Moreover, the definitions of these dimensions vary across different approaches, as noted in [8], where the authors also stress the lack of the formal basis behind process model quality evaluation.

In this paper, we aim to propose a theoretically grounded approach to evaluating the structural complexity of process models discovered from event logs of multi-agent system, such that the independence of agents and the complexity of their asynchronous interactions are taken into account.

Table 1. The trace in an event log of a multi-agent system.

Timestamp	Action	Agent
2025-03-25 14:07:46.580465000	a_1	Agent 1
2025-03-25 14:07:47.392390000	a_2	Agent 1
2025-03-25 14:07:48.683925000	b_1	Agent 2
2025-03-25 14:07:49.485992000	b_2	Agent 2
2025-03-25 14:07:50.349081000	b_3	Agent 2
2025-03-25 14:07:51.218921000	a_3	Agent 1

The main contributions of our paper are:

1) A collection of structural complexity dimensions measuring the relative independence of agents in the structure of a multi-agent system model.
2) The key properties of the proposed dimensions, including compositionality and monotonicity with respect to the size of compared models.
3) The results of the experimental study aimed to assess the proposed approach sensitivity to structural modifications in process models of multi-agent systems.

The remainder of our paper is organized as follows. In the following section, we outline the theoretical background concerning Petri nets and the concept of neighboring transitions. Section 3 describes our approach to evaluating the structural complexity of process models based on various methods to aggregate and average the number of neighboring transitions. In Section 4, we highlight the key properties of our approach and report the main conclusions from a series of experiments. In Section 5, we discuss the related research, and Section 6 concludes the paper by emphasizing the main results and future work directions.

2. Preliminaries

In this section, we collect the basic definitions concerning Petri nets and their structure based on [3]. A restricted class of Petri nets is used in our paper to model multi-agent systems and asynchronous communications among agents. We also present the concept of neighboring transitions – the basis for evaluating the structural complexity of a multi-agent system model.

2.1 Workflow Nets

Workflow (WF) nets are extensively used to model the structure of processes discovered from event logs of information systems [9]. A WF-net is a restricted class of a Petri net.

Definition 1 [3]. A Petri net is a triple $N = (P, T, F)$, where P is a set of places, T is a set of transitions, $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation.

Graphically, a Petri net is shown as a bipartite graph, where places are circles, transitions are boxes, and the flow relation is shown by directed arcs. Places represent states, whereas transitions – actions executed by the agents in a multi-agent system. Figure 2 shows the example of a Petri net modeling the multi-agent system with two cyclic agents exchanging messages via channel places a and b.

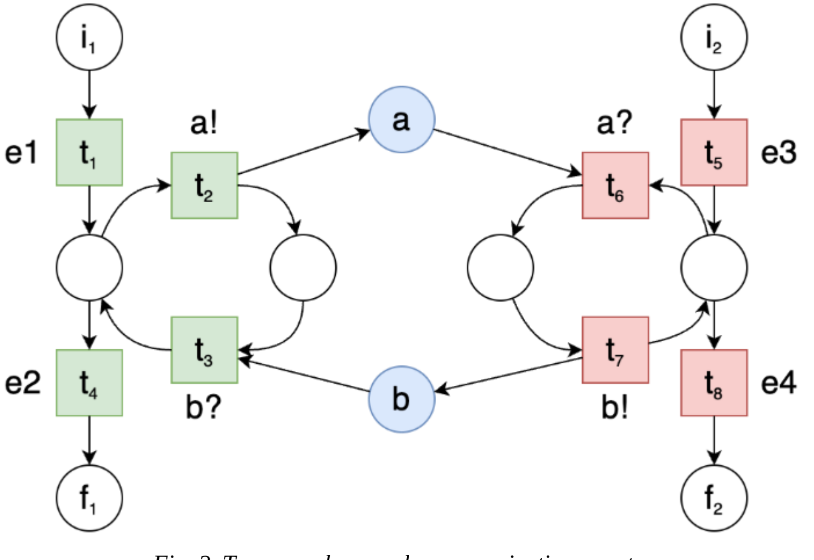


Fig. 2. Two asynchronously communicating agents.

Definition 2 [3]. Let $N = (P, T, F)$ be a Petri net. For any node $x \in P \cup T$:

1) $\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ is the preset of x;
2) $x \bullet = \{y \in P \cup T \mid (x, y) \in F\}$ is the postset of x.

The •-notion can be extended to a set of nodes $X \subseteq P \cup T$ in a Petri net $N = (P, T, F)$, i.e., $\bullet X = \bullet x$ and $X \bullet = x \bullet$ for all $x \in X$. A subnet of N generated by $X \subseteq P \cup T$ is denoted $N(X) = (P \cap X, T \cap X, F \cap (X \times X))$. We consider Petri nets without self-loops, i.e., $\bullet x \cap x \bullet = \emptyset$, and isolated transitions, i.e., $|\bullet t| \geq 1$ and $|t \bullet| \geq 1$ for all $t \in T$.

A WF-net has the distinguished initial and final state rep- resented by the corresponding subsets of places.

Definition 3 [9]. A WF-net $N = (P, T, F, in, fin)$ is a Petri net equipped with the initial state $in \subseteq P$ and the final state $fin \subseteq P$, where:

1) $\bullet in = fin \bullet = \emptyset$;
2) $\forall x \in P \cup T \; \exists \, i \in in \; \exists \, f \in fin: (i, x), (x, f) \in F^*$,

where F^* is the reflexive transitive closure of F.

Intuitively, according to the second requirement of this definition, every node in a WF-net lies on a path from its initial to its final state. The Petri net shown in Fig. 2 is also the WF- net, where the initial and final states of two communicating agents can be easily identified.

We assign labels to transitions in a WF-net. Transition labels correspond to actions recorded in the event log of a multi-agent system. Thus, we define a labeled WF-net.

Definition 4. Let A be a finite set of actions. A labeled WF-net $N = (P, T, F, in, fin, \lambda)$ is a WF-net equipped with a total transition labeling function $\lambda: T \to A \cup \{\tau\}$, where τ is the label of the silent action.

$T_{VIS} = \{t \in T \mid \lambda(t) \neq \tau\}$ denotes the set of visible transitions. Silent actions can also be referred to as invisible which means that they are internal system actions and are not assigned to an agent. Apart from that, silent actions are also used in process mining algorithms to organize the structure of a model discovered from an event log.

For example, the labels of the transitions in the WF-net shown in Fig. 2 are provided next to the transitions. We will further work only with labeled WF-nets. Therefore, we will omit the term “labeled”, and we will not clutter figures with transition labels. By convention, transitions corresponding to silent actions will be shown by black boxes.

In our study, we consider multi-agent systems with k asynchronously communicating agents. Thus, the set of actions A in Definition 4 can be partitioned into k subsets, i.e., $A = \{A_1, A_2, \dots, A_k\}$, where A_i with $i = 1, \dots, k$ is the set of actions executed by the specific agent.

This partition of A naturally induces the partition of transitions in a WF-net $N = (P, T, F, in, fin, \lambda)$ into k subsets with respect to their labels. Consequently, $T = \{T_1, T_2, \dots, T_k\}$, where $T_i = \{t \in T \mid \lambda(t) \in A_i\}$ with $i = 1, \dots, k$.

By convention, we will use colors to distinguish transitions belonging to different subsets in the partition of T. For example, the green and red colors distinguish two agents in the WF-net shown in Fig. 2.

2.2 Composing Labeled WF-nets via Channel Places

To model the asynchronous interactions among agents in a multi-agent system, we use the composition of WF-nets with labeled transitions studied in [10]. This operation involves inserting channel places between specifically labeled transitions:

1) label “c!” – sending a message to channel “c”;
2) label “c?” – receiving a message from channel “c”.

In Fig. 2, the channel place “a” connects two transitions with complement labels (“a!” and “a?”). The same holds for the channel place “b”. By convention, channel places are colored blue. The composition of two WF-nets, denoted by $N_1 \circledast N_2$, is also a WF-net. In addition, this operation is both commutative and associative. Thus, it is a convenient tool to model multi-agent systems with three or more interacting agents.

Our paper is focused on the structural analysis of multi- agent system models. Thus, we do not recall the behavior-preserving properties of the asynchronous WF-net composition proven in [10].

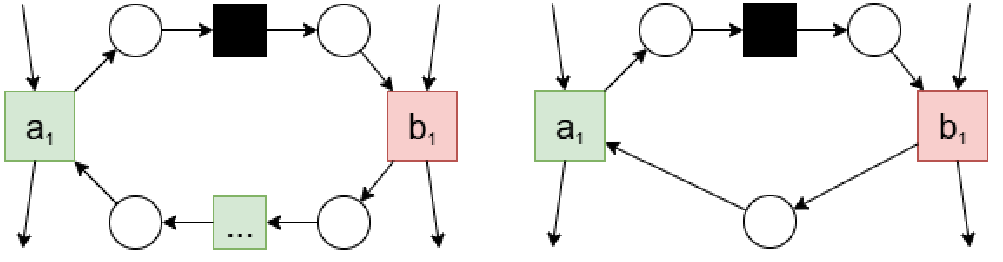
2.3 Neighboring Transitions

The main intuition behind our approach to evaluating the structural complexity of a multi-agent system model, given by a WF-net, lies in identifying the structural connections among transitions representing actions executed by different agents. In this light, we adopt the notion of neighboring transitions discussed earlier in [11].

Definition 5. Let $N = (P, T, F, in, fin, \lambda)$ be a WF-net. Neighboring transitions are defined by a relation $nbt \subseteq \mathrm{T} \times \mathrm{T}$, where $\forall t_1, t_2 \in nbt$ if and only if:

1) $\lambda(t_1) \neq \tau$ and $\lambda(t_2) \neq \tau$;
2) $\forall\, t \in T\ \ \{t_1, t_2\}: (t_1, t), (t, t_2) \in F^* \Rightarrow \lambda(\mathrm{t}) = \tau$

If $(t_1, t_2) \in nbt$, we refer to t_2 as the neighbor of t_1. The second requirement in Definition 5 tells that transition t_2 is the neighbor of t_1 if and only if all other transitions lying on a path from t_1 to t_2 are silent. In addition, the nbt relation is not necessarily symmetric, since there can be different paths connecting t_1 and t_2 in both directions (see Fig. 3, where silent transitions are shown by black boxes).



(a) $(t_1, t_2) \in nbt$ *(b)* $(t_1, t_2), (t_2, t_1) \in nb$

Fig. 3. Neighboring transitions are not necessarily symmetric.

Next, using the nbt relation, we identify the neighbors of transition $t \in T$ in a WF-net with respect to the partition of T induced by the labeling.

Definition 6. Let $N = (P, T, F, in, fin, \lambda)$ be a WF-net, where $T = \{T_1, T_2, \dots, T_k\}$ according to λ. For $t \in T_i$ with $i = 1, \dots, k$ define:

1) $all(t) = \{t' \in T \mid (t, t') \in nbt\}$;
2) $diff(t) = \{t' \in all(t) \mid t' \notin T_i\}$

Let $T_{WN} = \{t \in T \mid all(t) \neq \emptyset\}$ denote the subset of transitions that have neighbors.

The set $all(t)$ contains all neighbors of transition t according to nbt. The set $diff(t)$ includes only those neighbors that correspond to actions executed by agents different from that of t according to the labeling. In Fig. 2, for example, transition t_2 labeled by “a!” has $all(t_2) = \{t_3, t_6\}$ and $diff\ (t_2) = \{t_6\}$. Note that transitions t_4 and t_6 have $all(t_4) = all(t_6) = \emptyset$, since they both have the single outgoing place not connected to other transitions.

In the next section, we explore the structural complexity of the WF-net representing a multi-agent system with k asynchronously interacting agents with the help of neighboring transitions discussed above.

3. Structural Complexity of WF-nets

In this section, we present our approach to evaluating the structural complexity of WF-nets modeling multi-agent systems with asynchronously interacting agents. The approach is based on the identification of neighboring transitions. Intuitively, the number of neighboring transitions $diff(t)$ across a complete WF-net reflects the extent to which different agents are mutually independent.

We say that the structure of a WF-net reflects the interaction-oriented viewpoint of the multi-agent system architecture if the aggregated number of neighboring transitions $diff(t)$ is minimized to those directly involved in the asynchronous agent interactions via channel places. Given a WF-net $N = (P, T, F, in, fin, \lambda)$, we introduce the general notion of Neighbor Independence $NI(N)$ – a [0, 1]- normalized value with respect to the following two principles for all $t \in T_{WN}$:

1) p0: $NI(N) = 0 \Leftrightarrow diff(t)$ is equal to $all(t)$;
2) p1: $NI(N) = 1 \Leftrightarrow diff(t)$ contains no transitions.

Figure 4 illustrates the main intuition behind these principles, which we discuss further in more detail.

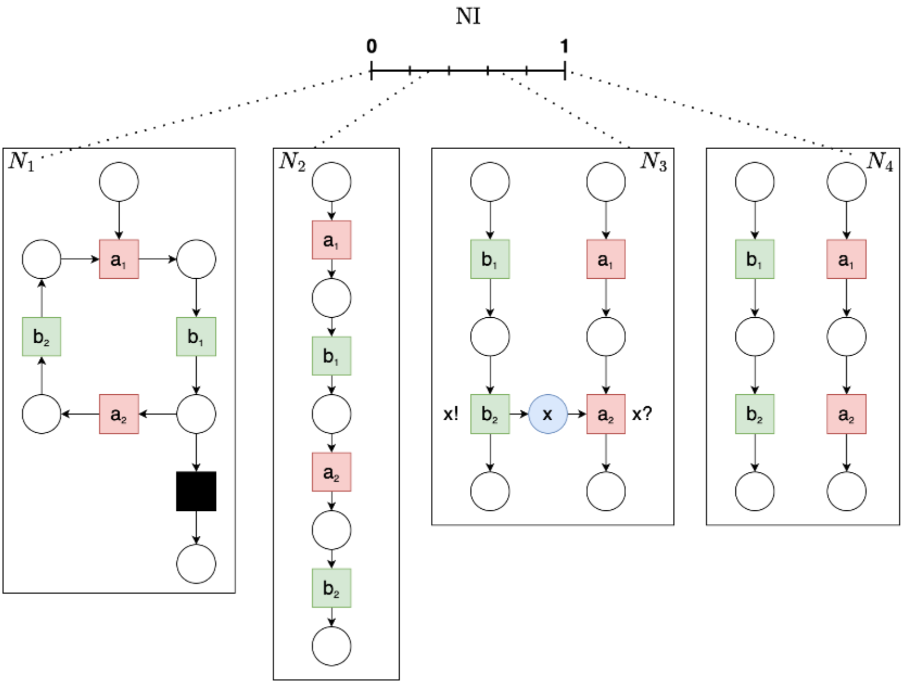


Fig. 4: The basic principles of Neighbor Independence.

Table 2 provides the configuration of the neighboring transitions for WF-nets N_1, N_2, N_3 and N_4 shown in Fig. 4. These WF-nets represent four different models of the multi-agent system with two asynchronously communicating agents. Each agent executes two actions. In this table, we construct a couple $\langle all(t), diff(t) \rangle$ for every transition in the mentioned WF-nets.

The first principle p0 corresponds to a WF-net whose structure does not allow us to distinguish individual agents. Such structure does not reflect the interaction-oriented viewpoint of the multi-agent system architecture. For example, in N_1, shown in Fig. 4, transitions corresponding to the behavior of different agents alternate. N_2 slightly differs from N_1, since there is a single transition b_2 in N_2 without neighbors.

Table 2. Neighboring transitions for WF-nets from Fig. 4.

	a_1	a_2	b_1	b_2
N_1	$\langle\{b_1\},\{b_1\}\rangle$	$\langle\{b_2\},\{b_2\}\rangle$	$\langle\{a_2\},\{a_2\}\rangle$	$\langle\{a_1\},\{a_1\}\rangle$
N_2	$\langle\{b_1\},\{b_1\}\rangle$	$\langle\{b_2\},\{b_2\}\rangle$	$\langle\{a_2\},\{a_2\}\rangle$	$\langle\emptyset,\emptyset\rangle$
N_3	$\langle\{a_2\},\emptyset\rangle$	$\langle\emptyset,\emptyset\rangle$	$\langle\{b_2\},\emptyset\rangle$	$\langle\{a_2\},\{a_2\}\rangle$
N_4	$\langle\{a_2\},\emptyset\rangle$	$\langle\emptyset,\emptyset\rangle$	$\langle\{b_2\},\emptyset\rangle$	$\langle\emptyset,\emptyset\rangle$

The second principle p1 corresponds to a WF-net with completely isolated agents (see WF-net N_4 in Fig. 4). This is also demonstrated by the emptiness of $diff(t)$ for every transition in N_4 as given in Table 2. Multi-agent systems with isolated agents and, correspondingly, with the perfect value of NI should be considered as the degenerate case, since the usual architecture of a system involves agent interactions.

In other cases, the value of $NI(N)$ should be strictly between 0 and 1, according to the number of (channel) places connecting transitions corresponding to actions executed by different agents. For example, N_3 shown in Fig. 4 does not fall directly into p0 or p1.

In the following paragraphs, we propose different methods for aggregating and averaging the ratios of $diff(t)$ to $all(t)$ for all transitions in a WF-net in order to obtain the [0, 1]-normalized value. If we detect "pathological" cases violating the basic principles p0 and p1, we identify the main causes and consider how to handle these cases.

3.1 Global Neighbor Independence

In this paragraph, we present our first attempt to evaluate the structural complexity of a multi-agent system model. Global Neighbor Independence directly evaluates the aggregated ratio of $|diff(t)|$ to $|all(t)|$ for all transitions in a WF-net N.

Definition 7. Let $N = (P, T, F, in, fin, \lambda)$ be a WF-net. The Global Neighbor Independence $GNI(N)$ is calculated as follows:

$$GNI(N) = 1 - \frac{\sum_{t\in T}|diff(t)|}{\sum_{t\in T}|all(t)|} \quad (1)$$

Table 3 provides the GNI values computed for the four WF-nets N_1, N_2, N_3 and N_4 from Fig. 4 as well as for the WF-net shown in Fig. 2. It can be clearly seen that the ordering of these values generally conforms with the basic principles p0 and p1 of Neighbor Independence discussed earlier.

Table 3. Comparison of GNI values.

	N_1	N_2	N_3	Fig. 2	N_4
$GNI(.)$	0	0	0.667	0.833	1

We should note that comparing neighbor connectivity values of WF-nets representing completely different multi-agent systems is useless. For instance, $GNI(N) > GNI(N_2)$, where N is the WF-net from Fig. 2, but the overall structure of N is perceived to be more complex in comparison with N_2 from Fig. 4 at least because of the number of channel places.

The ordering $GNI(N_1) < GNI(N_2) < GNI(N_3) < GNI(N_4)$ is easy to explain, since the structure of these WF-nets demonstrates different levels of mutual independence of two asynchronously communicating agents. GNI does not distinguish N_2 from N_1, since the absence of neighbors for b_2 in N_2 cannot influence the fraction in (1).

We can obtain $GNI(N)$ close to 1 for a WF-net N with a lot of channel places connecting transitions of different agents. The following proposition and example shown in Fig. 5 illustrates this problem.

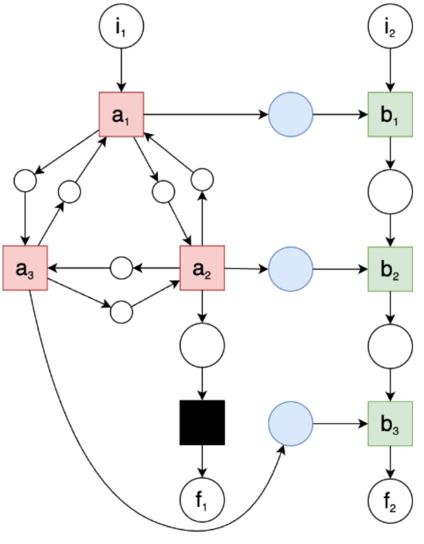


Fig. 5. A pathological WF-net with GNI close to 1.

The main idea behind this WF-net that that there are transitions, such that $diff(t)$ represents a tiny portion of $all(t)$ making the fraction in Defintion 7 close to 0.

Proposition 1. There exists a labeled WF-net N with $GNI(N)$ tending to 1, such that the principle p1 is violated.

Proof. Consider a WF-net $N = (P, T, F, in, fin, \lambda)$ shown in Fig. 5. This WF-net models asynchronous interactions between two agents through three channels.

Transitions in the leftmost (red) agent form the cyclic structure, where every transition is the neighbor to the others. The rightmost (green) agent has the simple sequential structure with the same number of visible transitions as the red agent. If we remove channel places from this WF-net, $GNI(N)$ will be 1, since there are isolated agents.

Let us introduce channels as shown in Fig. 5, where every transition in the first agent connected via a single channel place to a single transition in the second agent. We will have that $GNI(N) = 8/11$ which is close to 1, but the agents are tightly connected. In addition, we generalize this case to n transitions in each agent:

- $X = \sum_{t \in T} |diff(t)| = n$ and
- $Y = \sum_{t \in T} |all(t)| = n(n-1) + 2n - 1 = n^2 + n - 1$

Taking the value of n to infinity, we obtain that the ratio X/Y (required by Definition 7) tends to 0, and GNI tends to 1. Thus, we come to the violation of the principle p1, since it implies isolated agents, but the structure of N with many connections between agents reflects the exact opposite situation. We obtain the exaggerated value of GNI for a WF-net when there are too many internal neighbors within the same agent. The relative impact of inserting a new channel place between a pair of transitions, i.e., increasing $|diff\ (t)|$, can be negligible. We call this the neighbor explosion problem.

A possible immediate solution to this problem could be obtained by evaluating the global ratio of internal transitions of agents to all visible transitions in a WF-net, as formalized in the following definition.

Definition 8. Let $N = (P, T, F, in, fin, \lambda)$ be a WF-net. Let $T_{IN} = \{t \in T \mid diff(t) = \emptyset\}$. The Perimeter of Neighbor Independence $PNI(N)$ is calculated as follows:

$$PNI(N) = \frac{|T_{IN}|}{|T_{VIS}|} \quad (2)$$

Consider the values of PNI provided in Table 5 for the WF-nets shown in Fig. 2, Fig. 4 and Fig. 5. Firstly, we see that PNI resolves the pathological case of GNI shown in Fig. 5 by collapsing the value to 0.5 as soon as Formula 2 eliminates the impact of the local neighbor structure of the red agent. Secondly, it is clear that PNI conforms with the basic principles p0 and p1.

Table 4. Comparison of PNI values.

	N_1	N_2	Fig. 5	Fig. 2	N_3	N_4
$PNI(.)$	0	0.250	0.500	0.750	0.750	1

Note that PNI has the locality property in the sense that changes in the structure of a WF-net do not require the complete recalculation as opposed to GNI. If we insert (remove) a channel place connecting internal transitions of different agents in a WF-net N, we only need to subtract (add) $1/|T|$ to (from) the current value $PNI(N)$. For example, if we insert a new channel between transitions b_1 and a_1 in N_3 shown in Fig. 4, the new value $PNI(N_2) = 0.75 - 0.25 = 0.5$.

However, inserting a new channel between two transitions of different agents that already have neighbors in the other agent does not affect the value of PNI. For example, if we insert a new channel place between transitions b_2 and a_1 in N_3 from Fig. 4, PNI remains the same (0.75), since T_{IN} does not change. Thus, the disadvantage to PNI is the lack of sensitivity to changes introduced into the way of agent interactions. In general, we consider PNI to be a partial improvement of GNI for the resolution of the neighbor explosion problem.

To sum up our reasoning on Global Neighbor Independence, it is necessary to highlight the special WF-net for which the denominator in (1) becomes 0. This WF-net represents a multi-agent system with k isolated agents – each agent can execute only a single action modeled by a single transition, respectively (see Fig. 6).

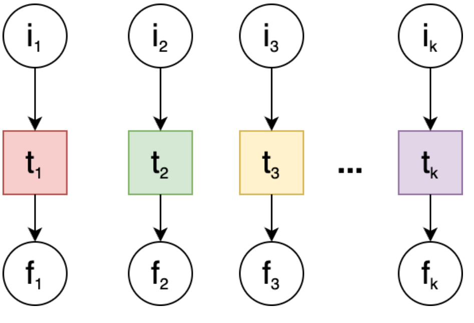


Fig. 6. A WF-net with single-transition isolated agents.

In order to overcome this zero-denominator problem, we need to exclude transitions without neighbors with $all(t) = \emptyset$ from the consideration. That is why Definition 6 introduces the set $T_{WN} = \{t \in T \mid all(t) \neq \emptyset\}$.

3.2 Local Neighbor Independence

This paragraph studies the averaged version of GNI. In fact, for each transition t in a WF-net N, we evaluate the ratio of $diff(t)$ to $all(t)$ and aggregate them separately as formalized below. Recall that T_{VIS} denotes the subset of visible transitions whose label is different from τ.

Definition 9. Let $N = (P, T, F, in, fin, \lambda)$ be a WF-net. Local Neighbor Independence $LNI(N)$ is calculated as follows:

$$LNI(N) = 1 - \frac{1}{|T_{VIS}|} \sum_{t \in T_{WN}} \frac{|diff(t)|}{|all(t)|} \tag{3}$$

Note that LNI does not take transitions without neighbors into account. Intuitively, these transitions do not make a meaningful contribution to the complexity of a WF-net representing a multi-agent system with asynchronously interacting agents.

Consider the values of LNI computed for WF-nets shown in Fig. 2, Fig. 4 and Fig. 5. Overall, we see that LNI does not violate the basic principles p0 and p1, as shown by the ordering of the four WF-nets in Fig. 4.

Table 5. Comparison of LNI values.

	N_1	N_2	N_3	Fig. 5	Fig. 2	N_4
$LNI(.)$	0	0.250	0.750	0.833	0.896	1

LNI is sensitive to all changes that can be introduced into a WF-net N regarding the way agents communicate as opposed to GNI and PNI. The corresponding change in the value of LNI can be easily calculated. Let N' be a new WF- net obtained after inserting a new channel place connecting transition t_1 to transition t_2 in N, such that t_1 and t_2 do not belong to the same set in the partition of T according to λ. Then, $LNI(N') = LNI(N) + \Delta$, where Δ is:

$$\frac{1}{T_{VIS}} \left(\frac{|diff(t_1)|}{|all(t_1)|} - \frac{|diff(t_1)|+1}{|all(t_1)|+1} \right) \tag{4}$$

For example, let us insert a channel place from transition a_1 to transition b_1 in N_3 shown in Fig. 4, i.e., make b_1 a neighbor to a_1. The direct application of Definition 9 gives us $LNI(N') = \frac{5}{8} = 0.625$. The change Δ, calculated by (4), is $1/4(0/1 - 1/2) = -1/8 = -0.125$. Therefore, $LNI(N') = LNI(N_3) + \Delta = 0.750 - 0.125 = 0.625$, which conforms with Definition 9 applied directly.

Similarly, we can derive the formula for the change in the value of LNI corresponding to the removal of a channel place connecting transition t_1 to transition t_2 in N, i.e., decreasing the number of neighbors for t_1, such that t_1 and t_2 do not belong to the same set in the partition of T according to λ. Then, $LNI(N') = LNI(N) + \Delta$, where Δ is:

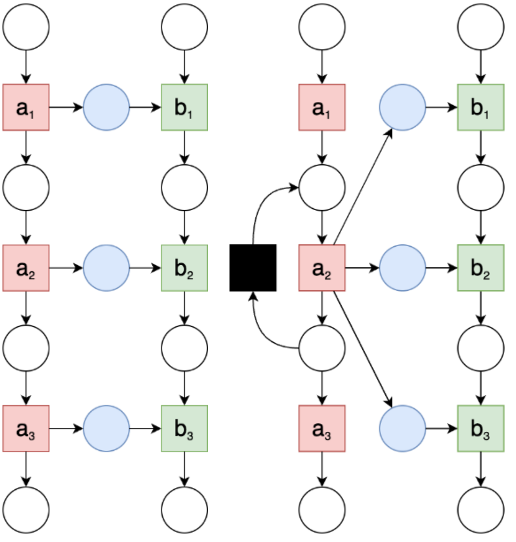
$$\frac{1}{T_{VIS}} \left(\frac{|diff(t_1)|}{|all(t_1)|} - \frac{|diff(t_1)|-1}{|all(t_1)|-1} \right) \tag{5}$$

For example, let us remove a channel between transitions a_1 and b_1 in the WF-net N shown in Fig. 5. Note that the original $LNI(N) = 5/6 = 0.833$. The direct application of Definition 9 gives us $LNI(N') = 16/18 = 0.889$. The corresponding change Δ, calculated by (5), is $1/6(1/3 - 0/2) = 1/18$. Thus, $LNI(N') = LNI(N) + \Delta = 5/6 + 1/18 = 16/18$, which again agrees with Definition 9 applied directly.

Overall, (4) and (5) help eliminate the need to recalculate LNI for a whole WF-net, once changes to the structure of agent communication are introduced.

Next, we discuss the key example to differentiate LNI from GNI. Consider WF-nets N_{MT} and N_{ST} shown in Fig. 7. They represent two ways of asynchronous interactions between two agents via a fixed number of channels using multiple transitions and a single transition, respectively. N_{ST} cannot be obtained using the composition described in Section 2.2.

By Definition 7, $GNI(N_{MT}) = GNI(N_{ST}) = 1 - 3/7 = 0.571$. Global Neighbor Independence cannot distinguish the inner structure of asynchronous agent interactions, since the effect of $diff(t)$ is spread among all other neighbors. However, by Definition 9, $LNI(NMT) = 0.667 < LNI(NST) = 0.875$. Local Neighbor Independence not only aggregates and averages the number of transition neighbors, but also reveals the details on the asynchronous interaction.



(a) N_{MT} – multi-transition *(b) N_{ST} – single-transition*

Fig. 7. Asynchronous interactions via several channels.

Thus, decreasing the number of transitions involved in asynchronous interactions, while maintaining the same number of channel places, makes the structure of model simpler and increases the value of LNI. Consider the generalization of WF-nets shown in Fig.7 up to n transitions in each agent:

$$LNI(N_{MT}) = 1 - \frac{1}{2n}\left(\frac{n-1}{2} + 1\right) = \frac{3n-1}{4n} \quad (6)$$

$$LNI(N_{ST}) = 1 - \frac{1}{2n}\left(\frac{n}{n+1}\right) = \frac{2n+1}{2n+2} \quad (7)$$

It is easy to see that $LNI(N_{MT}) < LNI(N_{ST})$ for $n > 1$.

To sum up, LNI does not solve the neighbor explosion problem (see Fig. 5 and Proposition 1). The localized calculation of $|diff(t)|/|all(t)|$ does not help us to avoid tending LNI to 1, since each transition in the leftmost agent has too many neighbors, whereas only one of them is connected with the rightmost agent. We can try to "shift" the weight of the aggregated fraction $|diff\ (t)|/|all(t)|$ in order to move the value LNI to the left of 1 – a possible way for this will be discussed in the following paragraph.

Nonetheless, the advantages to Local Neighbor Independence, discussed above, outweigh the problem with the pathological WF-net.

3.3 Balanced Local Neighbor Independence

Recall that the set of visible transitions T_{VIS} in a WF-net N are partitioned into k subsets – $T_{VIS} = \{T_1, T_2, \dots, T_k\}$, where T_i is the subset of transitions corresponding to the actions executed by the specific agent with $i = 1, 2, \dots, k$.

Here, we consider the use of a weighted average of LNI's computed separately for each agent in a WF-net. In order to keep the notation simple, the definition provided below implicitly considers transitions only with neighbors, where $all(t) \neq \emptyset$.

Definition 10. Let $N = (P, T, F, in, fin, \lambda)$ be a WF-net, where $T_{VIS} = \{T_1, T_2, \dots, T_k\}$ is a partition of T_{VIS} according to λ. Let $w_1, w_2, \dots, w_k$ be weights, where $w_1 + w_2 + \dots + w_k = 1$. Balanced Local Neighbor Independence $BLNI(N)$ is calculated as follows:

$$BLNI(N) = 1 - \sum_{i=1}^{k} \frac{w_i}{|T_i|} \sum_{t \in T_i} \frac{|diff(t)|}{|all(t)|} \quad (8)$$

The main question with $BLNI$ is how to arrange weights $w1, w2, \ldots, wk$. On the one hand, weights can be assigned equally, i. e., $w_1 = w_2 = \cdots = w_k = 1/k$. In this case, every agent has the equal contribution to the structural complexity of a WF-net. On the other hand, the uneven arrangement of weights can be used to increase (decrease) the relative impact of an agent on the structural complexity of a WF-net.

Table 6 provides the values of $BLNI$ for the pathological WF-net from Fig. 5 with different arrangement of agent weights, where w_R is the weight for the leftmost (red) agent and w_G is the weight for the rightmost (green) agent. Since we want to increase the impact of the red agent, we show how the value of $BLNI$ changes with the increase in the weight w_R.

As we can see in Table 6, increasing the weight w_R of the red agent leads to a decrease of the value of $BLNI$ up to its minimum when $w_R = 1$. However, this cannot completely resolve our pathological WF-net example in Fig. 5 in the general case. Therefore, we can admit that localization of the structural complexity evaluation does not overcome the neighbor explosion problem when there are transitions for which $|diff(t)|/|all(t)|$ tends to 0.

Table 6. BLNI with different weight arrangements for the WF-net from Fig. 5.

	$W_R = 0.5$ $W_G = 0.5$	$W_R = 0.7$ $W_G = 0.3$	$W_R = 1$ $W_G = 0$
$BLNI(.)$	0.833	0.767	0.667

Another possible reason to assign uneven weights w_i to agents in (8) is the significant difference in their size (the number of transitions). Consider the WF-net shown in Fig. 8, where the agent sending a message has a single transition, while the other agent has eight transitions. The values of LNI and $BLNI$ with various weight arrangements for this WF-net are given in Table 7 below, where w_R is the weight of the red agent, and w_G is the weight of the green agent.

As we can see in Table 7, the equal weight assignment for agents in the WF-net in Fig. 8 significantly lowers the structural complexity evaluation compared to LNI. The main reason for this is that the only transition in the green agent is involved into the communication with the red agent. In addition, we note that the arbitrarily arranged weights ($w_G = 0.7, w_R = 0.3$) that ignore, for instance, size ratio, can result in the poorly interpreted result.

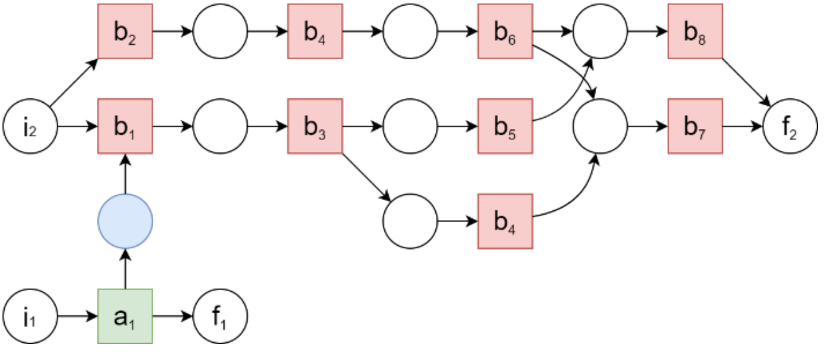


Fig. 8. A WF-net with significantly different agents.

Thus, according to the values provided in Table 7, LNI gives the most appropriate evaluation of the structural complexity of a multi-agent system with two agents (with a very different size) communicating via a single channel.

To sum up, $BLNI$ is a flexible version of LNI, since one can assign different weights to agents. However, weights assigned to agents should not conflict with the real structure of a WF-net. Otherwise, one can dramatically change the original LNI For example, if we set $w_G = 0$ and $w_R =$

1 for the pathological WF-net N in Fig. 5, we obtain $BLNI(N) = 0$, but the structure of this WF-net allows us to identify asynchronously communicating agents as subnets.

Table 7. LNI and BLNI with different weight arrangements for the WF-net from Fig. 8.

$LNI(.)$	$BLNI(.)$		
	$W_R = 0.5$ $W_G = 0.5$	$W_R = 0.7$ $W_G = 0.3$	$W_R = 1$ $W_G = 0$
0.900	0.500	0.300	0.700

Therefore, the basic recommendation for $BLNI$ can be either to use the standard equal weight arrangement $w_1 = w_2 = \dots = w_k = 1/k$ or to use an arrangement scheme that agrees with the internal structure of agents, for instance, based on the number of transitions. For the equal weight arrangement $w_1 = w_2 = \dots = w_k = 1/k$, the following obvious property, connecting LNI and $BLNI$, holds.

Proposition 2. Let $N = (P, T, F, in, fin, \lambda)$ be a WF-net, where $TVIS = \{T_1, T_2, \dots, T_k\}$ is the partition of T according to λ, and $|T_1| = |T_2| = \dots = |T_k|$. Let $w_1 = w_2 = \dots = w_k = 1/k$ be the weights. Then $BLNI(N) = LNI(N)$.

Proof. Substitute the given weights into (8):

$$BLNI(N) = 1 - \sum_{i=1}^{k} \frac{1}{k|T_i|} \sum_{t \in T_i} \frac{|diff(t)|}{|all(t)|} \quad (9)$$

As soon as $|T_1| = |T_2| = \dots = |T_k|$:

$$BLNI(N) = 1 - \frac{1}{|T_{VIS}|} \sum_{i=1}^{k} \sum_{t \in T_i} \frac{|diff(t)|}{|all(t)|} \quad (10)$$

Propagate the sum over i in (10):

$$BLNI(N) = 1 - \frac{1}{|T_{VIS}|} \sum_{t \in T_{WN}} \frac{|diff(t)|}{|all(t)|} = LNI(N)$$

since Definition 10 does not consider transitions without neighbors.

3.4 Neighbor Independence: Discussion

Here, we summarize the main aspects of our approach to measuring the structural complexity of a multi-agent system model in terms of the extent to which the structure of a WF-net reflects the interaction-oriented viewpoints of the system architecture. We introduced Neighbor Independence to evaluate how tight or loose the internal structure of asynchronous agent interactions in a WF-net is. Various methods of aggregating and averaging neighboring transitions gave rise to the following notions:

1) Global Neighbor Independence (GNI);
2) Perimeter of Neighbor Independence (PNI);
3) Local Neighbor Independence (LNI);
4) Balanced Local Neighbor Independence (BLNI).

Based on the analysis performed in the previous paragraphs, we compare these variations of Neighbor Independence against the following criteria:

- **c1**: conformance with the basic principles;
- **c2**: sensitivity to the communication structure;
- **c3**: sensitivity to changes in the communication structure;
- **c4**: locality of recalculation.

The criterion c1 also considers the neighbor explosion problem, which is the main reason for the violation of the principle **p1**. Table 8 provides the concise comparison protocol.

Table 8: Comparison of the various versions of Neighbor Independence.

	GNI	LNI	$BLNI$	PNI
c1	partial			yes
c2	no	yes	yes	yes
c3	yes	yes	yes	no
c4	no	yes	yes	yes

Based on these comparison results, we consider LNI to be the best choice for evaluating the structural complexity of a multi-agent system model with respect to the way agents interact. The neighbor explosion problem can be considered as a degenerate modeling case which is very unlikely to occur. In the next section, we discuss the key properties associated with the calculation of LNI and conduct a series of experiments to confirm the main conclusions obtained here.

4. Properties and Experimental Evaluation

In this section, we discuss several key properties of our approach to evaluating the structural complexity of a WF-net, which represents a multi-agent system with k asynchronously interacting agents. Firstly, we demonstrate that:

1) LNI can be computed in a compositional way;
2) LNI is monotone with respect to the size of WF-nets connected by an abstraction/refinement relation.

Secondly, to analyze the behavior of Neighbor Independence concerning structural changes in WF-nets, we conduct a series of experiments and report the main outcomes from them.

4.1 Compositionality of LNI

If we partition the set of transitions T in a WF-net N and substitute this partition into (3), we obtain that $LNI(N)$ is a weighted average as proven in the following proposition, which considers the basic case of partitioning T into two subsets.

Proposition 3. Let $N = (P, T, F, in, fin, \lambda)$ be a labeled WF-net. Let $T_{VIS} = T_1 \cup T_2$, where $T_1 \cap T_2 = \emptyset$. Then:

$$LNI(N) \;=\; w_1\, LNI(N_1) + w_2\, LNI(N_2)$$

where N_i is the subnet of N generated by T_i and $w_i = |T_i| / (|T_1| + |T_2|)$ with $i = 1, 2$.

Proof. The proof is by construction. By Definition 9 for N:

$$LNI(N) = 1 - \frac{1}{|T_1|+|T_2|} \sum_{t \in T_1^{WN} \cup T_2^{WN}} \frac{|diff(t)|}{|all(t)|} \tag{11}$$

Expand the sum over transitions with neighbors in (11):

$$LNI(N) = 1 - \frac{1}{|T_1| + |T_2|} \sum_{i=1}^{2} \sum_{t \in T_i^{WN}} \frac{|diff(t)|}{|all(t)|}$$

By Definition 9, for the subnet N_i with $i = 1,2$:

$$LNI(N_i) = 1 - \frac{1}{|T_i|} \sum_{t \in T_i^{WN}} \frac{|diff(t)|}{|all(t)|} \tag{12}$$

Multiply both sides in (12) by w_i with $i = 1,2$:

$$w_i\ LNI(N_i) = w_i - \frac{1}{|T_1| + |T_2|} \sum_{t \in T_i^{WN}} \frac{|diff(t)|}{|all(t)|}$$

Since $w_1 + w_2 = 1$ by the proposition assumption, we have:

$$\sum_{i=1}^{2} w_i\ LNI(N_i) = 1 - \frac{1}{|T_1| + |T_2|} \sum_{i=1}^{2} \sum_{t \in T_i^{WN}} \frac{|diff(t)|}{|all(t)|}$$

which exactly corresponds to $LNI(N)$.

Thus, $LNI(N)$ corresponds to the average of $LNI(N_i)$ computed independently for the subnet N_i of N and weighed by the size of these subnets (the number of transitions T_i). In fact, Proposition 3 allows us to choose arbitrary subnets (possibly having common places) in a WF-net N to calculate $LNI(N)$ (see Fig. 9a, where subnets contain transitions belonging to different agents). However, we believe that the appropriate way to identify subnets in a WF-net N is to isolate transitions belonging to different agents, as shown in Fig. 9b, which corresponds to the composition of WF-nets as discussed in Section 2.2. This WF-net models a multi-agent system with three interacting agents, i.e., the set of transitions is partitioned into three subsets $T = \{T_1, T_2, T_3\}$, where $T_1 = \{a_1, c_1\}$, $T_2 = \{b_1\}$ and $T_3 = \{d_1\}$.

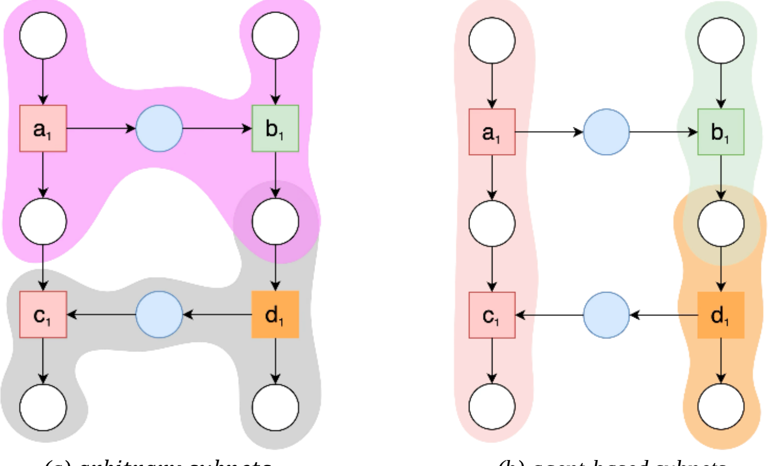


Fig. 9. Two ways of choosing subnets in a WF-net.

Now consider the calculation of $LNI(N)$ for the WF-net N in Fig. 9. The direct application of Definition 9 gives $LNI(N) = 1 - 1/4\ (1/2 + 1/1 + 1/1) = 3/8 = 0.375$. Let us use the arbitrary subnets given by Fig. 9a to calculate $LNI(N)$, which, in this case, is equal to the following:

$$\frac{1}{2}\left(1 - \frac{1}{2}\left(\frac{1}{2} + \frac{1}{1}\right)\right) + \frac{1}{2}\left(1 - \frac{1}{2}\left(\frac{1}{1}\right)\right) = \frac{3}{8} \quad (13)$$

Let us also use the agent-based subnets given by Fig. 9b to calculate $LNI(N)$, which is equal to the following in this case:

$$\frac{2}{4}\left(1 - \frac{1}{2}\left(\frac{1}{2}\right)\right) + \frac{2}{4}\left(1 - \frac{1}{1}\left(\frac{1}{1}\right)\right) = \frac{3}{8} \quad (14)$$

Both (13) and (14) yield the same result corresponding to Definition 9 applied directly.

The compositionality behind Local Neighbor Independence allows us to decompose the calculation process for the large multi-agent system model into several independent parts that can be executed in parallel.

4.2 Monotonicity of LNI

As we mentioned in Section 3, comparing models of completely different multi-agent system with respect to Neighbor Independence is useless. However, in this paragraph, we aim to consider a restricted class of related models whose structural complexity comparison is meaningful. These are abstract WF-nets and their refinements related via α-morphisms [12]. The example of an α-morphism $\varphi: N_1 \to N_2$, where N_1 is a refinement of the abstract WF-net N_2, is shown in Fig. 10. N_2 can be seen as an interface which is implemented by N_1. In addition, both N_1 and N_2 can be seen as a composition of two WF-nets via two channels used to exchange messages – N_1 exhibits the refined behavior of the red agent. In fact, α-moprhisms are extensively used in [11] to identify which structural and behavioral properties are preserved in the composition of WF-nets.

Intuitively, within an α-morphism $\varphi: N_1 \to N_2$, a place in N_2 is refined by a subnet in N_2. For instance, a place p_2 in N_2 is refined (shown by the dashed arrow in Fig. 10) by a subnet with two transitions in N_1.

An α-morphism $\varphi: N_1 \to N_2$ is a surjective map, fully defined by the way transitions in N_1 are mapped on transitions in N_2. If transition t in N_1 is mapped to place p in N_2, then $\bullet\, t\, \bullet$ is mapped to p as well. If transition t_1 in N_1 is mapped to transition t_2 in N_2, then $\varphi(\bullet\, t_1) = \bullet\, t_2$ and $\varphi(t_1\, \bullet) = t_2\, \bullet$.

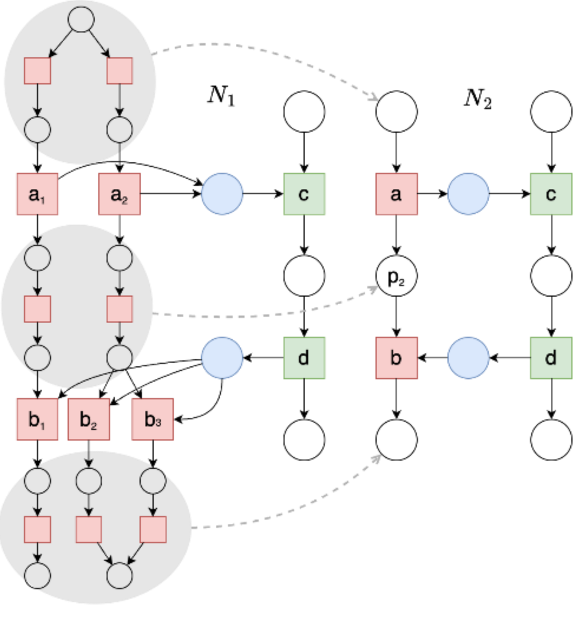


Fig. 10. The α-morphism between two WF-nets N_1 and N_2.

As proven in the following proposition, Local Neighbor Independence is monotone with respect to the size of WF- nets related by an α-morphism under the certain restrictions.

Proposition 4. Let $N_i = (P_i, T_i, F_i, in_i, fin_i, \lambda_i)$ be a WF- net with $i = 1,2$. Let $\varphi: N_1 \to N_2$ be an α-morphism, s.t. $|T_1 \cap \varphi^{-1}(P_2)| \geq |\varphi^{-1}(T_2)|$. Then, $LNI(N_1) \geq LNI(N_2)$.

Proof. The requirement $|T_1 \cap \varphi^{-1}(P_2)| \geq |\varphi^{-1}(T_2)|$ implies that the number of transitions in N_1 that are mapped on places in N_2 is at least as big as the number of transitions in N_1 that are mapped on transitions in N_2. Since φ is surjective, $|T_1| \geq 2\,|T_2|$. By Definition 9, $LNI(N_1)$ is:

$$1 - \frac{1}{|T_1|}\sum_{t_1 \in T_1} \frac{|diff(t_1)|}{|all(t_1)|} \leq 1 - \frac{1}{2|T_2|}\sum_{t_1 \in T_1} \frac{|diff(t_1)|}{|all(t_1)|} \quad (15)$$

By Definition 9, $LNI(N_2)$ is:

$$1 - \frac{1}{|T_2|}\sum_{t_2 \in T_2} \frac{|diff(t_2)|}{|all(t_2)|} \quad (16)$$

Note that the value of the sum in (15) over the set T1 is determined by those transitions in T_1 that are also mapped on transitions in N_2, because, for each transition t_1 mapped to a place in N_2, $|diff(t)| = 0$. There are at least $|T_2|$ transitions in N_1 mapped on transitions in N_2.

In N_1, there can be several cases concerning transitions mapped by φ on transitions. Firstly, $|diff(t_1)|$ can remain the same as $|diff(\varphi(t_1))|$ in N_2, and $|all(t_1)|$ can increase by new internal neighbors or remain the same. Then, (15) ≥ (16), which means $LNI(N_1) \geq LNI(N_2)$. Secondly, both $|diff(t_1)|$ and $|all(t_1)|$ can grow. Then, $|T_1| \geq a\,|T_2|$, where $a > 2$. Correspondingly, the value subtracted from 1 in (15) will be adjusted downward. Thus, $LNI(N_1) \geq LNI(N_2)$.

In other words, we show that a refined WF-net N_1, obtained by substituting places for subnets, cannot be structurally worse than an abstract WF-net N_2 in terms of asynchronous agent interactions. A refined WF-net introduces more internal agent transitions and lowers the relative impact of connections to other agents. For instance, $LNI(N_1) = 0.857 > LNI(N_2) = 0.625$ for the α-morphism provided in Fig. 10. We admit that we impose the rather strong restriction on the size of N_1, that can be weakened during the future research.

4.3 Experiment Layout

The behavior of Neighbor Independence concerning changes in WF-nets is analyzed using a family of randomly generated models. The experimental procedure includes:

1) Initialization. We specify parameters for the random WF-net generation:
   - the number of agents;
   - the number of places, transitions within each agent;
   - the number of places used for agent interactions;
   - the so-called density of arcs within each agent d_{in} and between them d_{out}.

In the context of our experiments, we work with agents of the same size (the number of transitions).

2) Generation. We generate the required number of places and transitions and randomly add arcs between them according to densities chosen in the previous step.
3) Evaluation. We calculate the values of GNI and LNI for the generated WF-nets. $BLNI$ is out of the scope of experiments, since it coincides with LNI for equally-sized agents (see Proposition 2).

Densities of arcs are defined by a number $0 \leq d_{in}, d_{out} \leq 1$ which represent the random discrete variable ξ_{in} and ξ_{out} – both of them is defined as shown in Table 9.

Table 9: Probability values for ξ_{in} and ξ_{out}

	$1 - d_{\{in/out\}}$	$d_{\{in/out\}}/2$	$d_{\{in/out\}}/2$
ξ	no arc between a place and a transition	arc from a place to a transition	arc from a transition to a place

Experiments are conducted to verify that:

h1: increasing the number of channels between unchanged

h2: increasing the number of places in agents with unchanged channels increases the values of GNI and LNI.

4.4 Experimental Results

Here, we report the key outcomes from the experiments conducted to analyze the behavior of GNI and LNI with respect to **h1** and **h2**. According to the layout described previously, we generated random WF-nets with $d_{in}, d_{out} = 0.8$ varying the number of transitions, places and channels. For each set of generation parameters, we produced and analyzed 100 WF-nets by averaging and assessing the standard deviation of GNI and LNI, respectively.

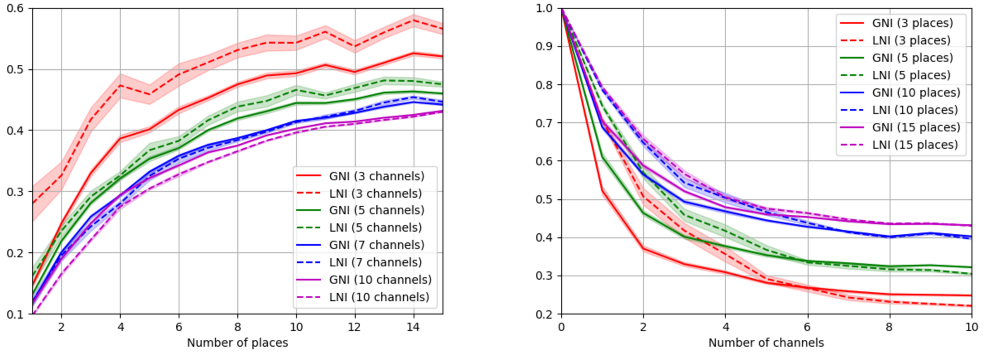


Fig. 11. A selection of experimental results for WF-nets with 5 transitions per agent.

The complete specification of the experimental results can be found in the repository [13], while Fig. 11 provides a selection of these results for WF-nets where each agent consists of 5 transitions. The shaded areas around the lines show the standard deviation.

Firstly, the left chart in Fig. 11 supports **h1**, showing a decrease in dimension values as the number of channels increases. Moreover, both GNI and LNI tend to similar values which rise as the number of internal agent places increases. Secondly, the right chart in Fig. 11 supports **h2**. The growth in the internal complexity of agent increases only the denominator in (1) and (3) leading to the respective increase of GNI and LNI, as shown in this chart.

To sum up, **h1** and **h2** follow from the way GNI and LNI are calculated according to Definition 7 and Definition 9. Moreover, complete experimental results shows that the trends in Fig. 11 are the same regardless of the size of agents.

5. Related work

As we mentioned in Introduction, there are many algorithms for the automated discovery of process models from event logs of information systems. Among the most wide-spread are Inductive miner [7], Fuzzy miner [14] and Region theory-based miner [15]. They overcome typical problems of event logs – incompleteness (a finite event log cannot cover all possible process scenarios) and noise (missing or duplicated activities). Inductive miner can also guarantee the perfect fitness and the behavioral correctness of a discovered WF-nets.

The existence of a large number of different process discovery algorithms makes the quality evaluation of discovered models a very important problem. Conformance checking [5] provides several basic quality dimensions – fitness, precision, generalization and simplicity. Fitness, precision and generalization are focused on measuring the correspondence between a discovered model and an event log from a behavioral point of view. Simplicity is aimed at assessing the structure of a discovered model, identifying, for example, useless nodes. Process models discovered from event logs of multi-agent systems requires special attention, since behavior-based quality dimensions may not differentiate models regarding the extent to which their structure reflects key interaction-oriented view- points of the multi-agent system architecture.

General graph-theoretic complexity measures, including density and cyclomatic complexity [16], consider the entire structure of a model, which is not sufficient for a model representing a multi-agent system with asynchronously inter- acting agents. Similarly, in [17], the authors discussed several empirical approaches to measuring the general complexity of Petri net. That is why, we developed the special approach to evaluating the structural complexity of a WF-net.

The problem of discovering process models from event logs with the understandable structure was considered in various contexts. A set of papers [18-20] proposed different approaches to improving the structure of discovered models by the localization of the event environment in a log and by composing fragments of the regular behavior with the rare and exceptional scenarios. Discovery of hierarchical process models, where a high-level event represents a subprocess, was studied in [21], [22], where the authors also considered sophisticated cyclic and concurrence relations between subprocesses. The identification of low-level and high-level events in an event log is a natural way to improve the structural representation of a process model. The paper [23] initialized a new direction of modeling and discovering object-centric Petri nets from event logs. Interactions of objects are represented through complex synchronizations which allow one to model consumption and production of objects of different types. Compositional discovery of behaviorally correct and architecture-aware process models, whose structure reflects individual agents and their interactions, from event logs of multi-agent systems was studied in [11].

Our study continues [11] in a way that we expand on the notion of neighboring transitions. Based on the number of places connecting transitions of different agents in a WF-net representing a multi-agent system, we proposed several methods for aggregating and averaging the number of neighboring transitions. These methods exhibit various levels of sensitivity to changes in the structure of asynchronous agent interactions.

6. Conclusion and Future Work

In this paper, we considered the problem of assessing the structural complexity of process models representing multi-agent systems with asynchronously interacting agents. Specifically, we worked with WF-nets that can be discovered from event logs recording the historical data on multi-agent system operation. The structural complexity of a WF-net was expressed through the notion of Neighbor Independence intuitively showing how loose the inter-agent connections are. Using this idea with the various levels of locality, we proposed three [0, 1]-valued dimensions of structural complexity showing the extent to which the structure of a WF-net reflects the interaction-oriented viewpoints of the system architecture.

We formulated the neighbor explosion problem – a special pathological case when the proposed dimensions tend to 1, since there are too many internal neighboring transitions within an agent. This can be resolved by refusing to perform the localized analysis of agent interactions. However, we consider this pathology to be a degenerate case. Its occurrence is unlikely in process modeling practice.

The core part of our study was devoted to identifying the main properties of the proposed structural quality dimensions. We proved that Local Neighbor Independence possesses compositionality and monotonicity. The latter holds for WF- nets related by a special abstraction/refinement relation under certain restrictions on their size. Sensitivity of the proposed dimensions to structural WF-net modifications was analyzed through the series of experiments. The experimental results are in agreement with the formal definitions. Inserting new channel places has a greater impact on the value of a dimension than increasing the internal complexity of an individual agent. Future research will be focused on the following directions. Firstly, we suggest extending the applicability of the proposed approach to general Petri nets. Secondly, we plan to develop an algorithm for localizing process model fragments that contribute most to the complexity of agent interactions.

References

[1]. W. van der Aalst, Process Mining: Data Science in Action. Springer Heidelberg, 2016. URL: https://link.springer.com/book/10.1007/978-3-662-49851-4.

[2]. A. Augusto, R. Conforti, M. Dumas, M. L. Rosa, F. M. Maggi, A. Marrella, M. Mecella, and A. Soo, Automated discovery of process models from event logs: Review and benchmark, IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 4, pp. 686–705, 2019. URL: https://ieeexplore.ieee.org/document/8368306.

[3]. W. Reisig, Understanding Petri nets. Modeling techniques, analysis methods, case studies. Springer Heidelberg, 2013. URL: https://link.springer.com/book/10.1007/978-3-642-33278-4.

[4]. A. Kalenkova, A. Burattin, and M. a. de Leoni, Discovering high-level BPMN process models from event data, Business Process Management Journal, vol. 25, pp. 995–1019, 2019. URL: https://www.emerald.com/bpmj/article/25/5/995/256745/Discovering-high-level-BPMN-process-models-from.

[5]. J. Carmona, B. van Dongen, A. Solti, and M. Weidlich, Conformance Checking: Relating Processes and Models. Springer Heidelberg, 2018. URL: https://link.springer.com/book/10.1007/978-3-319-99414-7.

[6]. A. Adriansyah, J. Munoz-Gama, J. Carmona, B. van Dongen, and W. van der Aalst, Alignment based precision checking, in Business Process Management Workshops, ser. LNBIP, vol. 132. Springer Heidelberg, 2013, pp. 137–149. URL: https://link.springer.com/chapter/10.1007/978-3-642-36285-9_15.

[7]. S. Leemans, D. Fahland, and W. van der Aalst, Discovering block- structured process models from event logs – a constructive approach, in PETRI NETS, ser. LNCS, J. Colom and J. Desel, Eds., vol. 7927. Springer Heidelberg, 2013, pp. 311–329. URL: https://link.springer.com/chapter/10.1007/978-3-642-38697-8_17.

[8]. W. M. van der Aalst, Relating process models and event logs – 21 conformance propositions, in ATAED@Petri Nets/ACSD, 2018. [Online]. URL: https://api.semanticscholar.org/CorpusID:49303433

[9]. W. van der Aalst, Workflow verification: Finding control-flow errors using petri-net-based techniques, in Business Process Management: Models, Techniques, and Empirical Studies, ser. LNCS, vol. 1806. Springer, Heidelberg, 2000, pp. 161–183. URL: https://link.springer.com/chapter/10.1007/3-540-45594-9_11.

[10]. L. Bernardinello, I. Lomazova, R. Nesterov, and P. L., Soundness-preserving composition of synchronously and asynchronously interacting workflow net components, Journal of Parallel and Distributed Computing, vol. 179, Article 104704, 2023. URL: https://www.sciencedirect.com/science/article/pii/S0743731523000679.

[11]. R. Nesterov, L. Bernardinello, I. Lomazova, and P. L., Discovering architecture-aware and sound process models of multi-agent systems: a compositional approach, Software and Systems Modeling, vol. 22, pp. 351–375, 2023. URL: https://link.springer.com/article/10.1007/s10270-022-01008-x.

[12]. L. Bernardinello, E. Mangioni, and L. Pomello, Local state refinement and composition of elementary net systems: An approach based on morphisms, in Transactions on Petri Nets and Other Models of Concurrency VIII, ser. LNCS, vol. 8100. Springer, Heidelberg, 2013, pp. 48–70. URL: https://link.springer.com/chapter/10.1007/978-3-642-40465-8_3.

[13]. E. Zemlyanoy, Structural complexity of workflow nets [Online]. URL: https://github.com/egorz734/structural-complexity-of-workflow-nets.

[14]. C. Gunther and W. van der Aalst, Fuzzy mining – adaptive process simplification based on multi-perspective metrics, in Business Process Management, ser. LNCS, G. Alonso, P. Dadam, and M. Rosemann, Eds., vol. 4714. Springer Heidelberg, 2007, pp. 328–343. URL: https://link.springer.com/chapter/10.1007/978-3-540-75183-0_24.

[15]. J. van der Werf, B. van Dongen, C. Hurkens, and A. Serebrenik, Process discovery using integer linear programming, in PETRI NETS, ser. LNCS, K. van Hee and R. Valk, Eds., vol. 5062. Springer Heidelberg, 2008, pp. 368–387. URL: https://link.springer.com/chapter/10.1007/978-3-540-68746-7_24.

[16]. T. McCabe, A complexity measure, IEEE Transactions on Software Engineering, vol. SE-2, no. 4, pp. 308-320, 1976. URL: https://ieeexplore.ieee.org/document/1702388.

[17]. L. Gang Soo and Y. Jung-Mo, An empirical study on complexity metrics of petri nets, Microelectronics Reliability, vol. 32, no. 9, pp. 1215–1221, 1992. URL: https://www.sciencedirect.com/science/article/pii/002627149290061O.

[18]. A. Kalenkova, A. Lomazova, and W. van der Aalst, Process model discovery: A method based on transition system decomposition, in PETRI NETS, ser. LNCS, G. Ciardo and E. Kindler, Eds., vol. 8489.

Springer Heidelberg, 2014, pp. 71–90.
URL: https://link.springer.com/chapter/10.1007/978-3-319-07734-5_5.

[19]. A. Kalenkova and I. Lomazova, Discovery of cancellation regions within process mining techniques, Fundamenta Informaticae, vol. 133, pp. 197–209, 2014. URL: https://journals.sagepub.com/doi/abs/10.3233/FI-2014-1071.

[20]. W. van der Aalst, A. Kalenkova, V. Rubin, and E. Verbeek, Process discovery using localized events, in PETRI NETS, ser. LNCS, R. Devillers and A. Valmari, Eds., vol. 9115. Springer Heidelberg, 2015, pp. 287–308. URL: https://link.springer.com/chapter/10.1007/978-3-319-19488-2_15.

[21]. A. Begicheva, I. Lomazova, and R. Nesterov, Discovering hierarchical process models: an approach based on events partitioning, Modeling and Analysis of Information Systems, vol. 31, no. 3, pp. 294–315, 2024. URL: https://www.mais-journal.ru/jour/article/view/1879.

[22]. A. Begicheva and I. Lomazova, Discovering high-level process models from event logs, Modeling and Analysis of Information Systems, vol. 24, no. 2, pp. 125–140, 2017. URL: https://www.mais-journal.ru/jour/article/view/504.

[23]. W. van der Aalst and A. Berti, Discovering object-centric petri nets, Fundamenta Informaticae, vol. 175, pp. 1–40, 2020. URL: https://journals.sagepub.com/doi/10.3233/FI-2020-1946.

Information about authors

Егор Олегович ЗЕМЛЯНОЙ – стажер-исследователь научно-учебной лаборатории процессно-ориентированных информационных систем факультета компьютерных наук НИУ Высшая школа экономики, магистр по направлению «Программная инженерия» с 2025 г. Его научные интересы включают моделирование и анализ поведения распределенных информационных систем с применением сетей Петри и их различных расширений.

Egor Olegovich ZEMLYANOY – a research assistant at the Laboratory of Process-Aware Information Systems (PAIS Lab), HSE University (Moscow, Russia), MSc in Software engineering since 2025. Research interests: modeling and analysis of distributed information systems behaviors using Petri nets and their extensions.

Роман Александрович НЕСТЕРОВ – доцент департамента программной инженерии факультета компьютерных наук, заведующий научно-учебной лабораторией процессно-ориентированных информационных систем НИУ Высшая школа экономики, кандидат компьютерных наук НИУ Высшая школа экономики с 2022 г. Его научные интересы включают подходы к моделированию и анализу поведения сложно организованных информационных систем, с помощью сетей Петри, теорию категорий и общую теорию параллелизма.

Roman Alexandrovich NESTEROV – Cand. Sci. (Computer Science), Associate Professor at the Department of Software Engineering at the Faculty of Computer Science, Head of the Laboratory of Process-Aware Information Systems (PAIS Lab), HSE University (Moscow, Russia). His research interests include approaches to modeling and analyzing the behavior of complex information systems using Petri nets, category theory, and the general theory of parallelism.