Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 9–15
DOI: 10.15439/2021F002
ISSN 2300-5963 ACSIS, Vol. 25

Using Free-Choice Nets for Process Mining and Business Process Management

Wil M.P. van der Aalst
Process and Data Science (Informatik 9), RWTH Aachen University, Aachen, Germany and
Fraunhofer-Institut für Angewandte Informationstechnik (FIT), Sankt Augustin, Germany
Email: wvdaalst@pads.rwth-aachen.de

***Abstract*—Free-choice nets, a subclass of Petri nets, have been studied for decades. They are interesting because they have many desirable properties normal Petri nets do not have and can be analyzed efficiently. Although the majority of process models used in practice are inherently free-choice, most users (even modeling experts) are not aware of free-choice net theory and associated analysis techniques. This paper discusses free-choice nets in the context of process mining and business process management. For example, state-of-the-art process discovery algorithms like the inductive miner produce process models that are free-choice. Also, hand-made process models using languages like BPMN tend to be free-choice because choice and synchronization are separated in different modeling elements. Therefore, we introduce basic notions and results for this important class of process models. Moreover, we also present new results for free-choice nets particularly relevant for process mining. For example, we elaborate on home clusters and lucency as closely-related and desirable correctness notions. We also discuss the limitations of free-choice nets in process mining and business process management, and suggest research directions to extend free-choice nets with non-local dependencies.**

I. Introduction

FREE-CHOICE nets can be used to model processes that include process patterns such as sequence, choice, loop, and concurrency. Compared to general Petri nets they require choice and synchronization to be separable. This is automatically the case in languages having explicit split and join operators (also called connectors or gateways) that do not mix choice and synchronization. For example, when using *Business Process Modeling Notation* (BPMN) with only AND and XOR gateways, the behavior is automatically *free-choice*. Although BPMN allows for many advanced constructs, the most widely used BPMN constructs can be easily mapped onto free-choice nets.

In this paper, we relate recent developments in free-choice nets to *Business Process Management* (BPM) in general and *process mining* in particular. The desire to manage and improve processes is not new. The field of scientific management emerged in the 1890-ties with pioneers like Frederick Winslow Taylor (1856-1915) [31]. Taylor already systematically analyzed manually recorded data in order to uncover potential process improvements. With the availability of computers, the focus shifted to automation. In the 1970-ties there was the expectation that office would would become increasingly automated, not requiring human intervention. Pioneers like Skip Ellis [18] and Michael Zisman [34] worked on so-called *office automation systems*. The ideas lead to the development of Workflow Management (WFM) systems in the 1990-ties (see [8]). Later, BPM systems broadened the scope from automation to management. In hindsight, these approaches were not so successful. For example, as the longitudinal study in [28] shows, many workflow implementations failed. As a result, WFM/BPM technology is often considered too expensive and only feasible for highly-structured processes. At the same time, people continued to model processes using flowchart-like description languages. For example, modeling tools such as ARIS and Signavio have been used to model millions of processes all over the globe. Modeling is less costly than automation, but the effect is often limited. Due to the disconnect between reality and such hand-made models, the BPM market was shrinking until recently. However, this changed with the uptake of *process mining* [2].

Process mining dramatically changed the way we look at process models and operational processes. Even seemingly simple processes like Purchase-to-Pay (P2P) and Order-to-Cash (O2C) are often amazingly complex, and traditional hand-made process models fail to capture the true fabric of such processes. Process mining bridges the gap between between *process science* (i.e., tools and techniques to improve operational processes) and *data science* (i.e., tools and techniques to extract value from data).

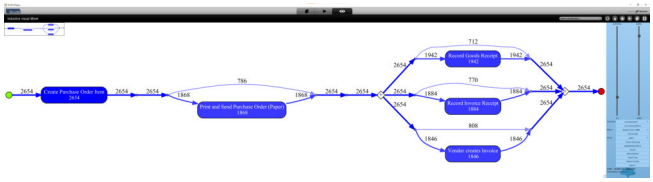

Fig. 1. Process model discovered using ProM's inductive miner.

Figure 1 shows ProM's inductive miner [22] in action. Based on (heavily filtered) data from SAP's Purchase-to-Pay (P2P) process, a process model is derived. Process discovery is just one of several process mining tasks. First, event data need to be extracted from information systems like SAP. *Process discovery* techniques transform such event data into process models (e.g., BPMN, Petri nets, and UML activity diagrams). There are simple approaches like creating so-called Directly-Follows-Graphs (DFGs) that do not discover concurrency thus having obvious problems [4]. Dozens, if not hundreds, of

more sophisticated algorithms were proposed [12], [2], [13], [20], [21], [22], [33]. Using replay and alignment techniques it is possible to do *conformance checking* and relate process models (hand-made or discovered) with event data. This can be used to discover differences between reality and model [2], [16], [30]. Moreover, the model can be extended with additional perspectives, e.g., organizational aspects, decisions, and temporal aspects.

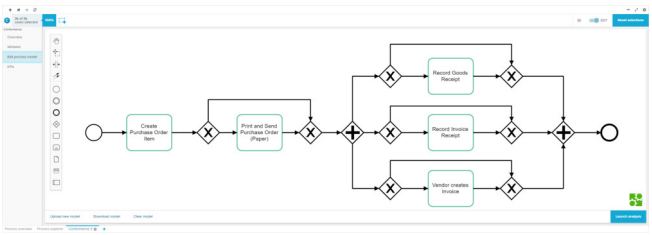

Fig. 2. BPMN model discovered using Celonis.

Currently, there are over 35 commercial process mining vendors (ABBYY Timeline, ARIS Process Mining, BusinessOptix, Celonis Process Mining, Disco/Fluxicon, Everflow, Lana, Mavim, MPM, Minit, PAFnow, QPR, etc.) and process mining is applied in most of the larger organizations. Figure 2 shows a BPMN model discovered using the Celonis process mining software. The same model can also be used for conformance checking and show where reality and model deviate.

Unlike traditional WFM/BPM technologies, there is a direct connection to the data. This allows stakeholders to spot inefficiencies, delays, and compliance problems in real-time. Process mining revitalized the BPM discipline, as is proven by the valuation of process mining firms. For example, Celonis is currently the first and only German "Decacorn" (i.e., a start-up whose value is considered to be over $10 billion).

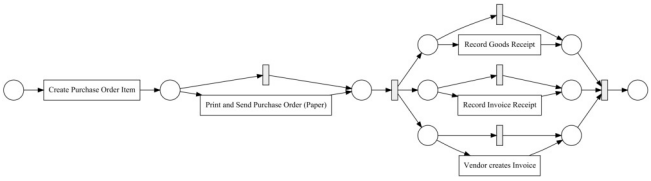

Fig. 3. A free-choice net generated from the models in Figures 1 and 2.

So how this related to free-choice nets? Process models play a key role in BPM and process mining, and these models can often be viewed as free-choice. Commonly used process notations are DFGs, BPMN models, Petri nets, and process trees. For example, the inductive mining approach uses *process trees* [22]. Although not visible, Figures 1 and 2 were actually generated using this approach. Process trees can be visualized using BPMN or Petri nets. Figure 3 shows the Petri net representation of the process tree. Any process tree corresponds to a so-called *free-choice net* having the same behavior. Later we will provide a formal definition for these notions. At this stage, it is sufficient to know that, in a free-choice net, choice and synchronization can be separated.

Any process tree can be converted to a free-choice net. Moreover, a large class of BPMN models is inherently free-choice. In a BPMN model there are flow objects such as events, activities, and gateways that are connected through directed arcs and together form a graph [26]. There are many modeling elements, but most process modelers use only a small subset [24]. For example, in many models, only exclusive gateways (for XOR-splits/joins) and parallel gateways (for AND-splits/joins) are used. Such models can be converted to free-choice nets [27]. It is also possible to convert BPMN models with inclusive gateways (i.e., OR-splits/joins) into free-choice nets (as long as the splits and joins are matching).

Since most process discovery techniques discover process models that are free-choice and also people modeling processes tend to come up with free-choice models, this is an interesting class to be studied. Therefore, this paper focuses on free-choice models. The goal is to expose people interested in BPM and process mining to free-choice-net theory.

Section II introduces preliminaries, including Petri nets, free-choice nets, and lucency. *Luceny* is a rather new notion which states that there *cannot* be two states enabling the same set of activities. Section III focuses on the class of process models having so-called *home clusters*. This class extends the class of sound models that can always terminate (e.g., no deadlocks) with the class of models that have a regeneration point. Free-choice nets with home clusters are guaranteed to be lucent. Hence, these nets are interesting for a wide range of applications and an interesting target class for process mining. Section IV discusses the limitations of free-choice nets, e.g., the inability to express non-local (i.e., long-term) dependencies. These insights may help to develop better process discovery techniques that produce more precise models. Section V concludes this paper.

II. Preliminaries

Free-choice nets are well studied [14], [15], [19], [32]. The definite book on the structure theory of free-choice nets is [17]. To keep the paper self-contained, first standard Petri net notions are introduced. If unclear, consider reading one of the standard introductions [11], [25], [29]. Most of the notations used are adopted from [6].

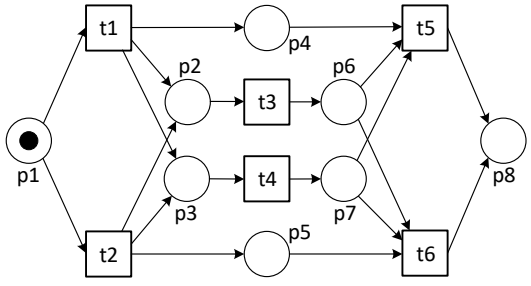

Fig. 4. A Petri net $N = (P, T, F)$ with $P = \{p1, p2, \ldots p8\}$, $T = \{t1, t2, \ldots, t6\}$, and $F = \{(p1, t1), (p1, t2), (t1, p4), \ldots, (t6, p8)\}$ that is not free-choice. The initial marking is $M = [p1]$, i.e., only place $p1$ contains a token.

A. Petri Nets

Figure 4 shows a Petri net with eight places, six transitions, and twenty arcs.

Definition 1 (Petri Net): A *Petri net* is a tuple $N = (P, T, F)$ with P the non-empty set of places, T the non-empty set of transitions such that $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ the flow relation such that the graph $(P \cup T, F)$ is (weakly) connected.

Definition 2 (Pre- and Post-Set): Let $N = (P, T, F)$ be a Petri net. For any $x \in P \cup T$: $\bullet x = \{y \mid (y, x) \in F\}$ and $x\bullet = \{y \mid (x, y) \in F\}$.

For example, in Figure 4, $\bullet p2 = \{t1, t2\}$, $\bullet t5 = \{p4, p6, p7\}$, $t1\bullet = \{p2, p3, p4\}$, and $p8\bullet = \emptyset$.

Definition 3 (Marking): Let $N = (P, T, F)$ be a Petri net. A *marking* M is a multiset of places, i.e., $M \in \mathcal{B}(P)$.[1] (N, M) is a marked net.

In the marking shown in Figure 4, transitions $t1$ and $t2$ are *enabled*. An enabled transition t can fire consuming a token from each input place in $\bullet t$ and producing a token for each output place in $t\bullet$.

Definition 4 (Enabling, Firing Rule, Reachability): Let (N, M) be a marked net with $N = (P, T, F)$. Transition $t \in T$ is enabled if $\bullet t \subseteq M$.[2] This is denoted by $(N, M)[t\rangle$ (each of t's input places $\bullet t$ contains at least one token). $en(N, M) = \{t \in T \mid (N, M)[t\rangle\}$ is the set of enabled transitions. Firing an enabled transition t results in marking $M' = (M \setminus \bullet t) \cup t\bullet$. $(N, M)[t\rangle(N, M')$ denotes that t is enabled in M and firing t results in marking M'. A marking M' is *reachable* from M if there exists a *firing sequence* σ such that $(N, M)[\sigma\rangle(N, M')$. $R(N, M) = \{M' \in \mathcal{B}(P) \mid \exists_{\sigma \in T^*} (N, M)[\sigma\rangle(N, M')\}$ is the set of all reachable markings. $(N, M)[\sigma\rangle$ denotes that the sequence σ is enabled when starting in marking M (without specifying the resulting marking).

Let N be the Petri net shown in Figure 4. $(N, [p1])[\sigma_1\rangle(N, [p4, p6, p7])$ with $\sigma_1 = \langle t1, t3, t4\rangle$ and $(N, [p1])[\sigma_2\rangle(N, [p8])$ with $\sigma_2 = \langle t2, t4, t3, t6\rangle$. We also define the usual properties for Petri nets.

Definition 5 (Live, Bounded, Safe, Dead, Deadlock-free, Well-Formed): A marked net (N, M) is *live* if for every reachable marking $M' \in R(N, M)$ and for every transition $t \in T$ there exists a marking $M'' \in R(N, M')$ that enables t. A marked net (N, M) is k-bounded if for every reachable marking $M' \in R(N, M)$ and every $p \in P$: $M'(p) \leq k$. A marked net (N, M) is *bounded* if there exists a k such that (N, M) is k-bounded. A 1-bounded marked net is called *safe*. A place $p \in P$ is *dead* in (N, M) when it can never be marked (no reachable marking marks p). A transition $t \in T$ is *dead* in (N, M) when it can never be enabled (no reachable marking enables t). A marked net (N, M) is *deadlock-free* if each reachable marking enables at least one transition. A Petri net N is *structurally bounded* if (N, M) is bounded for any marking M. A Petri net N is *structurally live* if there exists a marking M such that (N, M) is live. A Petri net N is *well-formed* if there exists a marking M such that (N, M) is live and bounded.

Definition 6 (Proper Petri Net): A Petri net $N = (P, T, F)$ is *proper* if all transitions have input and output places, i.e., for all $t \in T$: $\bullet t \neq \emptyset$ and $t\bullet \neq \emptyset$.

Definition 7 (Strongly Connected): A Petri net $N = (P, T, F)$ is *strongly connected* if there is a directed path between any pair of nodes.

Note that a strongly connected net is also proper. Figure 4 shows that the converse does not hold, the net is proper, but not strongly connected.

Definition 8 (Home Marking): Let (N, M) be a marked net. A marking M_H is a *home marking* if for every reachable marking $M' \in R(N, M)$: $M_H \in R(N, M')$.

The marked Petri net in Figure 4 has one home marking: $[p8]$.

B. Free-Choice Nets

The concepts and notations discussed apply to any Petri net. Now we focus on the class of *free-choice nets*. As indicated in the introduction, this is an important class because most process models used in the context of BPM and process mining are free-choice.

Definition 9 (Free-choice Net): Let $N = (P, T, F)$ be a Petri net. N is *free-choice net* if for any $t_1, t_2 \in T$: $\bullet t_1 = \bullet t_2$ or $\bullet t_1 \cap \bullet t_2 = \emptyset$.

The Petri net in Figure 4 is not free-choice because $\bullet t_5 \cap \bullet t_6 = \{p6, p7\} \neq \emptyset$, but $\bullet t_5 \neq \bullet t_6$. If we remove the places $p4$ and $p5$, then the net becomes free-choice. The places model a so-called *long-term (or non-local) dependency*: The choice between $t1$ and $t2$ in the beginning is controlling the choice between $t5$ and $t6$ at the end.

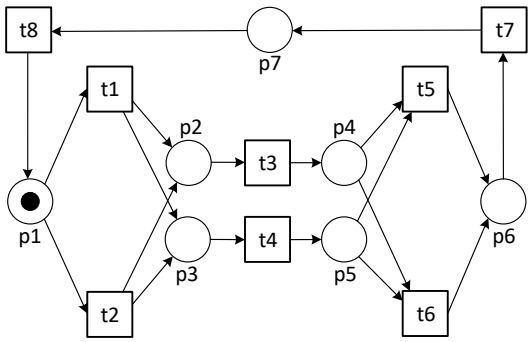

Fig. 5. A strongly-connected free-choice net.

Figure 5 is free-choice. Transitions $t1$ and $t2$ share an input place, but $\bullet t_1 = \bullet t_2 = \{p1\}$. Transitions $t5$ and $t6$ share an input place, but $\bullet t_5 = \bullet t_6 = \{p4, p5\}$.

The process model discovered using ProM (Figure 1) and Celonis (Figure 2) based on filtered SAP data is free-choice. Figure 3 shows the corresponding free-choice net.

[1] In a multiset elements may appear multiple times, e.g., $M = [p1, p2, p2, p2] = [p1, p2^3]$ is a multiset with four elements (three have the same value).

[2] $M_1 \subseteq M_2$ (inclusion), $M_1 \cup M_2$ (union), $M_1 \setminus M_2$ (difference) are defined for multisets in the usual way (i.e., taking into account the cardinalities. Sets are treated as multisets where all elements have cardinality 1.

C. Lucency

The notion of lucency was first introduced in [3]. A marked Petri net is *lucent* if there are no two different reachable markings enabling the same set of transitions, i.e., states are fully characterized by the transitions they enable.

Definition 10 (Lucent Petri nets): Let (N, M) be a marked Petri net. (N, M) is *lucent* if and only if for any $M_1, M_2 \in R(N, M)$: $en(N, M_1) = en(N, M_2)$ implies $M_1 = M_2$.

The marked Petri nets in Figures 3 and 5 are lucent, i.e., there are no two reachable markings that enable the same set of transitions. The marked Petri net in Figure 4 is not lucent. Markings $M_1 = [p2, p3, p4]$ and $M_2 = [p2, p3, p5]$ are both reachable and enable transitions $t3$ and $t4$.

Lucency is often a desirable property. Think, for example, of an information system that has a user interface showing what the user can do. In this setting, lucency implies that the offered actions fully determine the internal state and the system will behave consistently from the user's viewpoint. If the information system would not be lucent, the user could encounter situations where the set of offered actions is the same, but the behavior is very different. Another example is the worklist of a workflow management system that shows the workitems that can or should be executed. Lucency implies that the state of a case can be derived based on the workitems offered for it [6].

Characterizing the class of systems that are lucent is a foundational and also challenging question [3], [6], [7].

III. Free-choice Nets with Home Clusters

Workflow nets form a subclass of Petri nets starting with a source place *start* and ending with a sink place *end* [9]. The modeled workflow can be instantiated by putting tokens on the input place *start*. In the context of workflow nets, a correctness criterion called *soundness* has been defined [9]. A workflow net is sound if and only if the following three requirements are satisfied: for each case it is always still possible to reach the state which just marks place *end* (option to complete), if place *end* is marked all other places are empty for a given case (proper completion), and it should be possible to execute an arbitrary activity by following the appropriate route through the workflow net (no dead transitions) [9]. In [1], it was shown that soundness is decidable and can be translated into a liveness and boundedness problem, i.e., a workflow is sound if and only if the corresponding short-circuited net (i.e., the net where place *end* is connected to place *start*) is live and bounded. This can be checked in polynomial time for free-choice nets [1]. Figures 3 and 4 show two sound workflow nets. Figures 5 and 6 show free-choice nets that do not have a designated start and end place. Hence, soundness is not defined for these models.

A strongly-connected Petri net cannot be a workflow net. However, the lion's share of Petri net theory focuses on strongly-connected Petri nets. Therefore, [6] investigated a new subclass of Petri nets having a so-called *home cluster*.

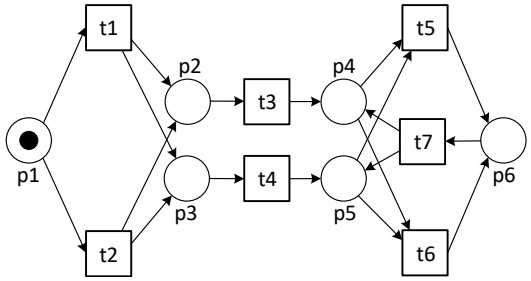

Fig. 6. A lucent free-choice net having two home clusters.

First, we define the notion of a *cluster*. A cluster is a maximal set of connected nodes, only considering arcs connecting places to transitions.

Definition 11 (Cluster): Let $N = (P, T, F)$ be a Petri net and $x \in P \cup T$. The *cluster* of node x, denoted $[x]_c$ is the smallest set such that (1) $x \in [x]_c$, (2) if $p \in [x]_c \cap P$, then $p\bullet \subseteq [x]_c$, and (3) if $t \in [x]_c \cap T$, then $\bullet t \subseteq [x]_c$. $[N]_c = \{[x]_c \mid x \in P \cup T\}$ is the set of clusters of N. $Mrk(C) = [p \in C \cap P]$ is the marking which only marks the places in C.

Figure 6 has five clusters: $[N]_c = \{\{p1, t1, t2\}, \{p2, t3\}, \{p3, p4\}, \{p4, p5, t5, t6\}, \{p6, t7\}\}$.

A home cluster is a cluster that serves as a "target" that can always be reached again. Hence, it can be seen as a generalization of soundness.

Definition 12 (Home Clusters): Let (N, M) be marked Petri net. C is a *home cluster* of (N, M) if and only if $C \in [N]_c$ (i.e., C is a cluster) and $Mrk(C)$ is a home marking of (N, M). If such a C exists, we say that (N, M) has a home cluster.

Figure 6 has two home clusters: $C_1 = \{p4, p5, t5, t6\}$ and $C_2 = \{p6, t7\}$.

Property 1 (Sound Workflow Nets Have A Home Cluster): Let (N, M) be a sound workflow net. (N, M) has a home cluster.

Also, all short-circulated sound workflow nets are guaranteed to have a home cluster. All marked Petri nets show thus far (i.e., Figures 3-6) have a home cluster. However, the nets in Figures 5 and 6 are not workflow nets.

Most of the results for Petri nets and in particular free-choice nets are defined for well-formed nets [11], [14], [15], [17], [19], [25], [29], [32]. Recall that a Petri net is well-formed if there exists a marking that is live and bounded. Some well-known properties of well-formed free-choice nets:

- A well-formed free-choice net is strongly connected.
- A bounded and strongly-connected marked free-choice net is live if and only if it is deadlock free.
- A marked free-choice net is live if and only if every proper siphon includes a marked trap.
- Well-formed free-choice nets are covered by P-components and T-components.
- Well-formedness can be decided in polynomial time for free-choice nets.
- Live and bounded free-choice nets have home markings.

Interestingly, marked free-choice nets having a home cluster do *not* need to be well-formed. Yet, free-choice nets having a home cluster have interesting properties as demonstrated in [6]. A surprising result is that free-choice nets having a home cluster are lucent.

Theorem 1 (Home Clusters Ensure Lucency [6]): Let (N, M) be a marked proper free-choice net having a home cluster. (N, M) is lucent.

The theorem can be used to show that the process models in Figures 3, 5, and 6 are lucent.

Theorem 1 is surprising since there are T-systems (i.e., marked graphs) that are live, bounded, safe, well-formed, and strongly connected that are not lucent. A proof of Theorem 1 is outside of the scope of this paper (see [6] for details). However, it is important to note that the proof does not rely on any of the classical results for well-formed nets. Instead, several new concepts are introduced, such as:

- *Expediting transitions* in a firing sequence of a free-choice net. As long as the order per cluster is maintained, transitions can fire earlier without causing any problems (e.g., deadlocks).
- The notion of *disentangled paths*, i.e., paths in the net that start and end with a place and do not contain elements that belong to the same cluster. A C-rooted disentangled path ends with a place in cluster C.
- A C-rooted disentangled path is *safe* if C is a home cluster. This implies that marked proper free-choice nets having a home cluster must be safe.
- The notion of *conflict-pairs*, i.e., a pair of markings such that no transition is enabled in both markings, but if a transition is enabled in one marking, the other marking must mark at least one of its input places.
- A marked proper free-choice net having a home cluster *cannot* have any conflict pairs.

These results make free-choice nets having a home cluster interesting candidate models in the context of BPM and process mining. However, as discussed next, there are also some limitations.

IV. Adding Non-Local Dependencies

Although many process discovery techniques return models that can be seen as free-choice and process modelers using BPMN are more-or-less forced to draw free-choice models, there are some limitations when using free-choice nets. Consider again the Petri net in Figure 4, which is not free-choice due to the places $p4$ and $p5$. The process model allows for the following four traces $L_1 = \{\langle t1, t3, t4, t5\rangle, \langle t1, t4, t3, t5\rangle, \langle t2, t3, t4, t6\rangle, \langle t2, t4, t3, t6\rangle\}$. Note that $t1$ is always followed by $t5$, and $t2$ is always followed by $t6$. In BPMN, we cannot express such dependencies (without resorting to data or other more advanced constructs). Ignoring the non-local dependencies represented by the places $p4$ and $p5$ leads to the BPMN model shown in Figure 7.

The corresponding free-choice net is shown in Figure 8. Both the BPMN model and the free-choice net allow for the following eight traces $L_2 = \{\langle t1, t3, t4, t5\rangle, \langle t1, t3, t4, t6\rangle, \langle t1, t4, t3, t5\rangle, \langle t1, t4, t3, t6\rangle, \langle t2, t3, t4, t5\rangle, \langle t2, t3, t4, t6\rangle, \langle t2, t4, t3, t5\rangle, \langle t2, t4, t3, t6\rangle\}$. Hence, the number of possibilities doubled.

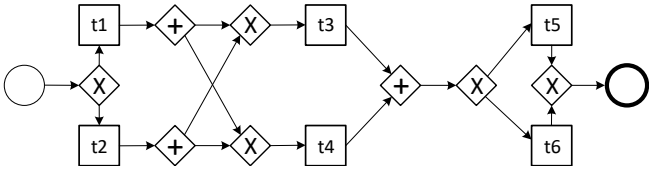

Fig. 7. A BPMN model that aims to describe the behavior in Figure 4 without local dependencies.

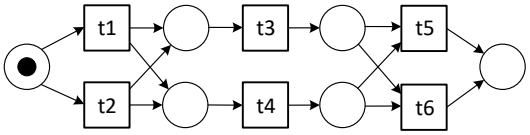

Fig. 8. The free-choice net corresponding to the BPMN model in Figure 7.

Most process discovery techniques will be unable to capture such non-local dependencies. Given an event log with only traces from L_1, most discovery techniques will produce a process model that allows for L_2. Some of the region-based process mining techniques can discover the process model allowing for only L_1. However, these techniques have many other problems: they tend to produce over-fitting models, cannot handle infrequent behavior, and are very time-consuming. Therefore, it may be better to first discover a *free-choice backbone model* that is then extended to make it more precise. Concretely, one can first discover a Petri net using the inductive mining approach and then add non-local dependencies. One can use, for example, a variant of the approach in [23] to add places. It is also possible to combine two types of arcs as in hybrid process models [10]. In [10], we use *hybrid Petri nets* and first discover a causal graph based on the event log. Based on different (threshold) parameters, we scan the event log for possible causalities. In the second phase, we try to learn places based on explicit quality criteria. Places added can be interpreted in a precise manner and have a guaranteed quality. Causal relations that cannot or should not be expressed in terms of places are added as sure or unsure arcs. A similar approach can be used for strongly correlating choices in a free-choice net.

There is also an interesting connection to the notion of *confusion*. Confusion is the phenomenon that the order of executing concurrent transitions may influence choices in the model. Here, we consider a simpler notion and consider a Petri net to be confusion-free when transitions that share an input place either cannot be both enabled or have the same set of input places.

Definition 13 (Confusion-Free): A marked Petri net (N, M) with $N = (P, T, F)$ is *confusion-free* if for any two transitions $t_1, t_2 \in T$ with $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ and $\bullet t_1 \neq \bullet t_2$ there is no reachable marking $M' \in R(N, M)$ such that $\{t_1, t_2\} \subseteq en(N, M)$.

All models in this paper are confusion free. Note that free-choice nets are by definition confusion-free. An interesting question is to develop automatic conversions from models that are "almost free-choice".

Thus far concepts such as confusion-free, lucency, and home clusters have not been exploited in process mining using traditional event logs. In [5], an algorithm is presented assuming translucent event logs that explicitly show the enabling of activities. However, such event logs are rarely available.

V. Conclusion

In this paper, we discussed recent results in free-choice net theory and related these results to Business Process Management (BPM) in general and process mining in particular. Although most discovery techniques produce free-choice models, this property is rarely exploited explicitly. Assuming that the process model is a free-choice net with a home cluster, provides many valuable properties relevant for process discovery. As shown in this paper, such models are, for example, guaranteed to be lucent. This implies that there cannot be two states enabling the same set of activities. Also, disentangled paths rooted in a home cluster are safe, i.e., such paths cannot contain two tokens. The open question is how to exploit this in process mining.

We also discussed the need to add non-local dependencies. Such dependencies destroy elegant properties such as lucency. Hence, they can be seen as a secondary layer of annotations. For example, we can connect clusters that are strongly correlated. The goal is to make the process models more precise without overfitting the data or destroying the structure of the model.

Acknowledgment

The author thanks the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

[1] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[2] W.M.P. van der Aalst. *Process Mining: Data Science in Action*. Springer-Verlag, Berlin, 2016.

[3] W.M.P. van der Aalst. Markings in Perpetual Free-Choice Nets Are Fully Characterized by Their Enabled Transitions. In V. Khomenko and O. Roux, editors, *Applications and Theory of Petri Nets 2018*, volume 10877 of *Lecture Notes in Computer Science*, pages 315–336. Springer-Verlag, Berlin, 2018.

[4] W.M.P. van der Aalst. A Practitioner's Guide to Process Mining: Limitations of the Directly-Follows Graph. In *International Conference on Enterprise Information Systems (Centeris 2019)*, volume 164 of *Procedia Computer Science*, pages 321–328. Elsevier, 2019.

[5] W.M.P. van der Aalst. Lucent Process Models and Translucent Event Logs. *Fundamenta Informaticae*, 169(1-2):151–177, 2019.

[6] W.M.P. van der Aalst. Free-Choice Nets With Home Clusters Are Lucent. *Fundamenta Informaticae*, 181(4):273–302, 2021.

[7] W.M.P. van der Aalst. Reduction Using Induced Subnets to Systematically Prove Properties for Free-Choice Nets. In D. Buchs and J. Carmona, editors, *Applications and Theory of Petri Nets and Concurrency (PN 2021)*, volume 12734 of *Lecture Notes in Computer Science*, pages 208–229. Springer-Verlag, Berlin, 2021.

[8] W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT Press, Cambridge, MA, 2002.

[9] W.M.P. van der Aalst, K.M. van Hee, A.H.M. ter Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve, and M.T. Wynn. Soundness of Workflow Nets: Classification, Decidability, and Analysis. *Formal Aspects of Computing*, 23(3):333–363, 2011.

[10] W.M.P. van der Aalst, R. De Masellis, C. Di Francescomarino, and C. Ghidini. Learning Hybrid Process Models From Events: Process Discovery Without Faking Confidence. In J. Carmona, G. Engels, and A. Kumar, editors, *International Conference on Business Process Management (BPM 2017)*, volume 10445 of *Lecture Notes in Computer Science*, pages 59–76. Springer-Verlag, Berlin, 2017.

[11] W.M.P. van der Aalst and C. Stahl. *Modeling Business Processes: A Petri Net Oriented Approach*. MIT Press, Cambridge, MA, 2011.

[12] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[13] A. Augusto, R. Conforti, M. Marlon, M. La Rosa, and A. Polyvyanyy. Split Miner: Automated Discovery of Accurate and Simple Business Process Models from Event Logs. *Knowledge Information Systems*, 59(2):251–284, May 2019.

[14] E. Best, J. Desel, and J. Esparza. Traps Characterize Home States in Free-Choice Systems. *Theoretical Computer Science*, 101:161–176, 1992.

[15] E. Best and H. Wimmel. Structure Theory of Petri Nets. In K. Jensen, W.M.P. van der Aalst, G. Balbo, M. Koutny, and K. Wolf, editors, *Transactions on Petri Nets and Other Models of Concurrency (ToPNoC VII)*, volume 7480 of *Lecture Notes in Computer Science*, pages 162–224. Springer-Verlag, Berlin, 2013.

[16] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich. *Conformance Checking: Relating Processes and Models*. Springer-Verlag, Berlin, 2018.

[17] J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1995.

[18] C.A. Ellis and G.J. Nutt. *Computer Science and Office Information Systems*. Xerox, Palo Alto Research Center, 1979.

[19] J. Esparza. Reachability in Live and Safe Free-Choice Petri Nets is NP-Complete. *Theoretical Computer Science*, 198(1-2):211–224, 1998.

[20] S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Discovering Block-structured Process Models from Event Logs: A Constructive Approach. In J.M. Colom and J. Desel, editors, *Applications and Theory of Petri Nets 2013*, volume 7927 of *Lecture Notes in Computer Science*, pages 311–329. Springer-Verlag, Berlin, 2013.

[21] S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In N. Lohmann, M. Song, and P. Wohed, editors, *Business Process Management Workshops, International Workshop on Business Process Intelligence (BPI 2013)*, volume 171 of *Lecture Notes in Business Information Processing*, pages 66–78. Springer-Verlag, Berlin, 2014.

[22] S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Scalable Process Discovery and Conformance Checking. *Software and Systems Modeling*, 17(2):599–631, 2018.

[23] L. Mannel and W.M.P. van der Aalst. Finding Complex Process-Structures by Exploiting the Token-Game. In S. Donatelli and S. Haar, editors, *Applications and Theory of Petri Nets 2019*, volume 11522 of *Lecture Notes in Computer Science*, pages 258–278. Springer-Verlag, Berlin, 2019.

[24] M. Zur Muehlen and J. Recker. How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In Z. Bellahsene and M. Léonard, editors, *Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE'08)*, volume 5074 of *Lecture Notes in Computer Science*, pages 465–479. Springer-Verlag, Berlin, 2008.

[25] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[26] OMG. Business Process Model and Notation (BPMN). Object Management Group, formal/2011-01-03, 2011.

[27] C. Ouyang, M. Dumas, A.H.M. ter Hofstede, and W.M.P. van der Aalst. Pattern-Based Translation of BPMN Process Models to BPEL Web Services. *International Journal of Web Services Research*, 5(1):42–62, 2007.

[28] H.A. Reijers, I.T.P. Vanderfeesten, and W.M.P. van der Aalst. The Effectiveness of Workflow Management Systems: A Longitudinal Study. *International Journal of Information Management*, 36(1):126–141, 2016.

[29] W. Reisig. *Petri Nets: Modeling Techniques, Analysis, Methods, Case Studies*. Springer-Verlag, Berlin, 2013.
[30] A. Rozinat and W.M.P. van der Aalst. Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, 33(1):64–95, 2008.
[31] F.W. Taylor. *The Principles of Scientific Management*. Harper and Bothers Publishers, New York, 1919.
[32] P.S. Thiagarajan and K. Voss. A Fresh Look at Free Choice Nets. *Information and Control*, 61(2):85–113, 1984.
[33] S.J. van Zelst, B.F. van Dongen, W.M.P. van der Aalst, and H.M.W Verbeek. Discovering Workflow Nets Using Integer Linear Programming. *Computing*, 100(5):529–556, 2018.
[34] M.D. Zisman. *Representation, Specification and Automation of Office Procedures*. PhD thesis, University of Pennsylvania, Warton School of Business, 1977.