Observable Liveness

Jörg Desel[1] and Görkem Kılınç[1,2]

[1] Fakultät für Mathematik und Informatik, FernUniversität in Hagen, Germany
[2] Universitá degli Studi di Milano-Bicocca, Italy

Abstract. Whereas the traditional liveness property for Petri nets guarantees that each transition can always occur again, observable liveness requires that, from any reachable marking, each observable transition can be forced to fire by choosing appropriate controllable transitions; hence it is defined for Petri nets with distinguished observable and controllable transitions. We introduce observable liveness and show this new notion generalizes liveness in the following sense: liveness of a net implies observable liveness, provided the only conflicts that can appear are between controllable transitions. This assumption refers to applications where the uncontrollable part models a deterministic machine (or several deterministic machines), whereas the user of the machine is modeled by the controllable part and can behave arbitrarily.

1 Introduction

Liveness and boundedness have turned out to be the most prominent behavioral properties of Petri nets – a Petri net is considered to behave well if it is live and bounded. This claim is supported by many publications since decades, and in particular by the nice correspondences between live and bounded behavior of a Petri net and its structure, see e.g. [4, 11]. Nowadays workflow Petri nets receive a particular interest, and with them the behavioral soundness property. However, as shown in [16], soundness of workflow nets is identical to the combination of liveness and boundedness of the net obtained by addition of a feedback place (between the final and the initial transition) to a workflow net. This way, these behavioral properties are also applied to models of processes, that have a start and an end action.

This paper concentrates on liveness, but looks at yet another scenario: Petri nets with transitions that can be observable or unobservable (silent transitions), and can be controllable or not. These nets are inspired by Petri net applications in control theory [8, 2], but can also be seen as a generalization of Petri nets with silent transitions. We provide a notion of liveness which is tailored for Petri nets with observable and controllable transitions, or for the systems modeled by these nets. Observable liveness of a model of a software system (embedded or not) with a user interface roughly means liveness from the user's perspective.

The standard definition of liveness for traditional Petri nets reads as follows:

> A transition t is live if, for each reachable marking m, there is a marking m' reachable from m that enables t. A net is live if all its transitions are live.

We consider Petri net models of software systems where only some activities are observable, and only a subset of these can be controlled by a user (like a vending machine, which has a user interface and an internal behavior). Our liveness notion applies to such nets, which also have observable transitions and, among them, controllable ones. This liveness notion still follows the idea that, no matter which marking m was reached, an occurrence sequence can be constructed which includes a given transition t. However, in contrast to the traditional definition,

- we only consider observable transitions t (i.e., if a transition cannot be observed then we do not care about it),
- we assume that instead of constructing the entire sequence, we (i.e., the user) can only control the net by choosing controllable transitions whenever they are enabled, whereas the net is always free to fire uncontrollable transitions arbitrarily. In particular, if a controllable transition is in conflict with an uncontrollable one, the controllable one might fire but cannot be enforced by the user.

This paper consists of two main parts with two different aims: In the first part of the paper we motivate observable liveness notion for observable software system models. The second part concentrates on the special case where the uncontrollable part of the considered software system behaves deterministically, that means conflict situation can only occur between two controllable transitions. We show that liveness implies observable liveness if no uncontrollable part ever is in conflict with any other transition. This assumption refers to applications where the uncontrollable part models a deterministic machine, whereas the user of the machine is modeled by the controllable part and can behave arbitrarily.

The paper is organized as follows. In Section 2, we introduce our setting and illustrate a simple example. Section 3 is devoted to basic definitions. In Section 4 , we introduce the notion of observable liveness. Section 5 discusses some properties of the new notion and relate it with the traditional liveness. Section 6 is devoted to the case of deterministic uncontrollable behavior. We finish the paper with conclusions, related work and further ideas.

2 The Setting

When defining observable liveness, several design decisions had to be made. We had a particular setting of a modeled system in mind, that motivated our choices. This section aims at explicating this setting and motivating our design decisions.

The generic software system to be modeled consists of a machine (or several machines), a user interface to this machine, and perhaps of activities and conditions which do not belong to the machine. The user can observe and control all activities outside the machine, he can neither control nor observe any activities inside the machine. Concerning the user interface, there are activities that the user can only observe but not control, whereas other interface activities might be both observable and controllable.

One might argue that instead of activities, only local states of machines are observable, for example a light which can be on or off. Then, instead of observing this state, in our setting we observe the activities that cause the changes of the state. In terms of nets, instead of observing a place, we observe the (occurrences of) transitions in the pre- or post-set of the place.

Controllable activities can be those not connected to the machine or can be activities of the interface. Whereas a controllable activity outside the machine is clearly also observable, one might argue that this is not obvious for controllable interface activities. In fact, if the activity can be caused by pressing a button, the user cannot be sure that with every use of this button the activity takes place. An additional prerequisite is that the activity is enabled by the machine, whereas buttons can always be pressed. So we implicitly assume that the user sees whether a controllable transition is enabled or not and can thus distinguish activities from non-activities caused by buttons.

Assume that a user wants to enforce an observable activity a after some previous run of the system. Then, depending on what he has observed so far, he should have a strategy to control activities in such a way that eventually he can observe a. By translating activities to transitions, the same holds for the Petri net model. The strategy is formalized by a function that maps an arbitrary sequence of observable transitions to a set of controllable transitions: if a sequence was observed, then one of these controllable transitions can be fired. Since the domain of this function is infinite in general, and its co-domain finite (theoretically exponential in the number of controllable transitions, but usually linear), different sequences are mapped to the same set. We assume that the user can effectively compute this function by using, e.g., only a finite history or an automata based approach. For generality of our approach, we nevertheless consider a strategy an arbitrary function as above.

There might be states in which controllable activities and uncontrollable ones are enabled, i.e., both the machinery and the user can do something. In such a state, we cannot expect that the user is able to do his controllable activity first. This means that, in case of competition between activities, the user does not have control if not only controllable activities are involved.

For an observably live activity, we want that the user can enforce the occurrence of this activity. Therefore, we provide an appropriate behavioral model of the net. Clearly, the user can only enforce any reaction from the machine if the machine obeys some progress assumption: we do not consider runs in which an uncontrollable transition is enabled, does not occur, and is not in conflict with any other occurring transition. Progress is only assumed for controllable transitions if they are persistently chosen by the response function and moreover concurrent to uncontrollable ones.

Throughout the paper, a controllable transition is illustrated via a black filled rectangle, an observable transition is illustrated by a bold rectangle, while unobservable ones are drawn by not bold rectangles. The incoming and outgoing arcs which are not connected to any place or transition are used when only a part of a net is shown.

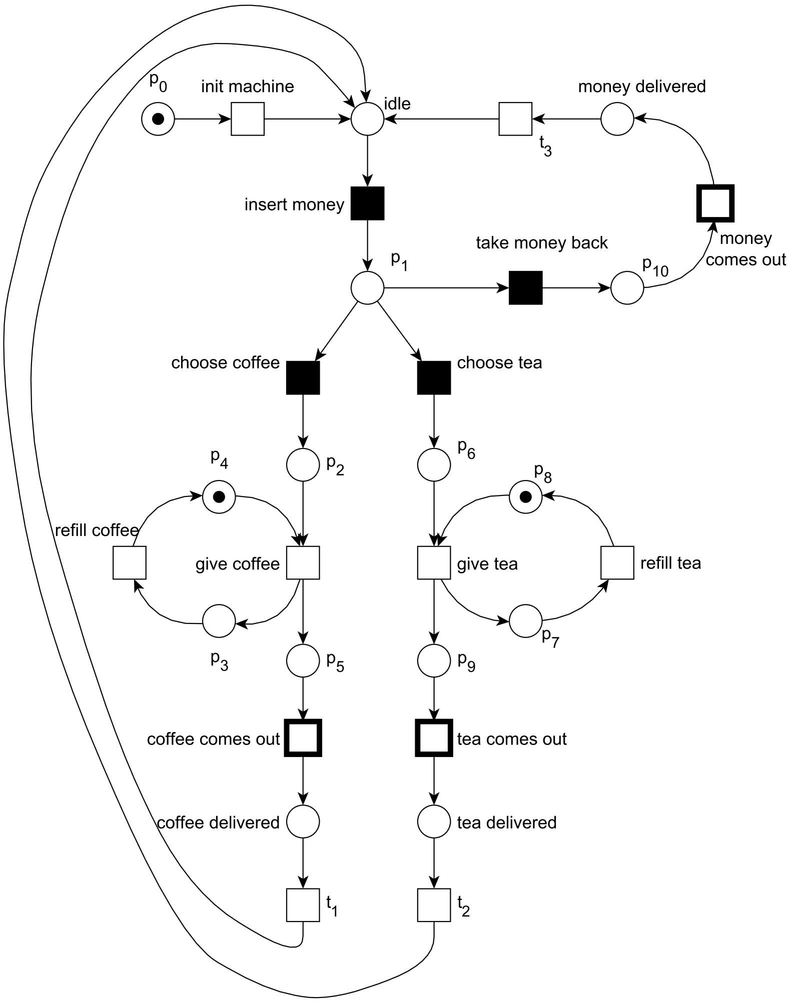

Fig. 1. An observably live net which represents a vending machine.

The example net shown in Fig. 1 models a vending machine with coffee and tea options. The user can operate the machine by inserting a coin and using three buttons (*insert coin, choose coffee, choose tea* and *take money back* are controllable transitions). Using these controllers, the user can take coffee, take tea or take his money back. The transitions *coffee comes out*, *tea comes out* and *money comes out* are observable, and the user can always force these transitions to occur by using the controllable ones. In other words, each of the observable transitions in the net is observably live and so the entire net is observably live. In case that there is no more coffee or tea, the machine needs a refill operation. In this case the user has to wait until the refill operation is done. Regarding the progress assumption, the refill operation will be done since *refill coffee* and *refill tea* transitions will fire eventually, and they are not in conflict with any transitions which can disable them. Note that the entire net is not live since the unobservable part includes a transition which can only fire once (*init machine*). However, this behavior does not affect our notion of observable liveness since the observable transitions can still be forced to fire. Considering such a machine, observable liveness is a useful notion to express the serviceability of a machine via an interface. We can generalize this for models of all kinds of software systems with a user interface. In this case, observable liveness expresses the liveness of a software system from the user's point of view.

3 Basic Definitions

An (initially marked) place/transition net N consists of a finite and non-empty set of places P, a finite and non-empty set of transitions T with $P \cap T = \emptyset$, a set of arcs $F \subseteq (P \times T) \cup (T \times P)$ and an initial marking $m_0 \colon P \to \mathbb{N}$. For a place or transition x, we denote its pre-set by ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$. Similarly, the post-set of x is denoted by $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$.

A marking m is an arbitrary mapping $m \colon P \to \mathbb{N}$. It enables a transition t if each place $p \in {}^\bullet t$ satisfies $m(p) > 0$. If it enables t then t can fire, which leads to the successor marking m', defined by

$$m'(p) = \begin{cases} m(p) + 1 & \text{if } p \in t^\bullet, p \notin {}^\bullet t \\ m(p) - 1 & \text{if } p \in {}^\bullet t, p \notin t^\bullet \\ m(p) & \text{otherwise} \end{cases}$$

We denote this by $m \ \xrightarrow{t}\ m'$.

The set of reachable markings of the net N, $\mathcal{R}(N)$, is the smallest set of markings that contains the initial marking m_0 and satisfies

$$[m \in \mathcal{R}(N) \ \wedge \ m \ \xrightarrow{t}\ m'] \implies m' \in \mathcal{R}(N).$$

The place/transition net is called *bounded* if $\mathcal{R}(N)$ is finite. Equivalently, it is bounded if and only if there exists a bound b such that each marking $m \in \mathcal{R}(N)$ satisfies for each place p: $m(p) \leq b$. It is called 1-bounded if this condition holds for $b = 1$.

If $m_1 \xrightarrow{t_1} m_2 \xrightarrow{t_2} m_3 \xrightarrow{t_3} m_4 \cdots$, then $t_1\,t_2\,t_3\,t_4 \ldots$ is called *occurrence sequence* (enabled at marking m_1). If an occurrence sequence σ is finite, i.e. $\sigma = t_1\,t_2 \ldots t_n$, then we write $m_1 \xrightarrow{\sigma} m_{n+1}$.

The place/transition net is *live* if, for each reachable marking m and each transition t, there exists a marking m' reachable from m that enables t. Equivalently, it is live if and only if for each transition t and each finite occurrence sequence σ enabled at m_0 there exists a transition sequence τ such that $\sigma\,\tau\,t$ is an occurrence sequence enabled at m_0. Note that in order to append two sequences, the left hand one is supposed to be finite. In turn, when writing $\sigma\,\tau$ we implicitly express that σ is finite.

Transitions can be observable or non-observable, and they can be controllable or non-controllable. We denote by $O \subseteq T$ the set of observable transitions and by $C \subseteq O$ the set of controllable ones.

A place/transition net with observable and controllable transitions is called *observable place/transition net* $N = (P, T, F, m_0, O, C)$. Given an occurrence sequence σ of the place/transition net, its projection $\overline{\sigma}$ to the observable transitions is called observable occurrence sequence. Conversely, a sequence $t_1\,t_2\,t_3 \ldots$ of observable transitions is an observable occurrence sequence if and only if there are finite sequences $\sigma_0, \sigma_1, \sigma_2, \ldots$ of unobservable transitions such that $\sigma_0\;t_1\;\sigma_1\;t_2\;\sigma_2\;t_3 \ldots$ is an occurrence sequence.

An infinite occurrence sequence $t_1\;t_2\;t_3 \ldots$ enabled at some marking m is called *weakly unfair* w.r.t. some transition t if, for some $k \in \mathbb{N}$, $t_1\;t_2 \ldots t_k\;t$ is enabled at m and, for each $j > k$, we have $^\bullet t_j \cap {}^\bullet t = \emptyset$ (after some finite initial phase, t is persistently enabled and not in structural conflict with any occurring transition). Notice that this definition is slightly weaker than the usual definition of weak fairness which only demands that t is persistently enabled. The occurrence sequence is *weakly fair* w.r.t. t if it is not weakly unfair w.r.t. t. By this definition, every finite occurrence sequence is weakly fair w.r.t. to all transitions.

There are many different fairness notions for Petri nets (and previously for other models). Our notion - often also called progress assumption - was first mentioned in [12]. It is particularly obvious for partially ordered behavior notions such as occurrence nets and can now be viewed as a standard notion.

4 Observable Liveness

In order to give the definition of observable liveness, we first stick to observable liveness of a single transition, which apparently has to be observable, and later define observable liveness of observable place/transition nets as observable liveness of all observable transitions.

So consider a single observable transition t which might be moreover controllable or not. If the net reaches from the initial marking m_0 a marking m by the occurrence of an arbitrary occurrence sequence σ_0, an agent wants to enforce transition t by selecting appropriate controllable, enabled transitions. If this is always (for each reachable marking m) possible, then we call t observably live.

From the marking m, the net first proceeds arbitrarily and autonomously, i.e., some occurrence sequence σ_1 without controllable transitions occur. This sequence can be

a) finite and lead to a deadlock,
b) finite and lead to a marking that enables controllable and uncontrollable transitions,
c) finite and lead to a marking that enables only controllable transitions,
d) or infinite.

For the infinite case we demand weakly fair behavior w.r.t. all uncontrollable transitions, i.e. there is progress in all concurrent parts of the net.

For cases b) and c), the agent fires a controllable transition and then proceeds as before with a next autonomous sequence σ_2, and so on. This will lead to either an infinite sequence σ_i, or eventually to case a) or case d).

Our liveness notion should express that – in case of observable liveness – there always is (at least one) controllable transition after any sequence σ_i in case c). To formalize this, (and to avoid an infinite alternation of $\forall$ and $\exists$) we introduce a response function φ, which delivers a set of possible controllable transitions as a response of the agent to the sequence observed so far. Notice that an observed sequence does not determine the reached marking because unobservable transitions might occur, changing the marking but not effecting the observed sequence. In turn, different observed sequences might lead to the same marking.

We call the transition t observably live if, for some such response function, we eventually observe t in the sequence created this way.

More formally, the definition reads as follows:

Definition 1. *Let $\varphi\colon O^* \to 2^C$ be a response function and let $m_0 \xrightarrow{\sigma_0} m$ be an occurrence sequence. We call an occurrence sequence σ, enabled at marking m, φ-maximal if it is either an infinite composition $\sigma = \sigma_1\, t_1\, \sigma_2\, t_2\, \sigma_3\, t_3\, \dots$ or a finite composition $\sigma = \sigma_1\, t_1\, \sigma_2\, t_2 \dots \sigma_k\, t_k\, \mu$, where $k \geq 0$, satisfying the following:*

a) All σ_i are finite and can be empty, μ is finite or infinite.
b) For each t_i we have $t_i \in \varphi(\overline{\sigma}_0\ \overline{\sigma}_1\ t_1\ \overline{\sigma}_2\ t_2 \dots \overline{\sigma}_i)$, i.e., t_i is a response to the sequence observed so far.
c) No σ_i contains a controllable transition ($i \geq 1$), and the same holds for μ.

Only for the second variant:
d) μ is weakly fair w.r.t. all non-controllable transitions. μ is moreover weakly fair w.r.t. all controllable transitions t satisfying $t \notin \varphi(\overline{\sigma}_0 \overline{\sigma'})$ for only finitely many prefixes σ' of σ.
e) If μ is finite then all transitions enabled after σ are controllable and do not belong to $\varphi(\overline{\sigma_0}\ \overline{\sigma})$ (this includes deadlocks).

Lemma 1. *Assume that σ_0 leads from m_0 to a marking m and σ is a φ-maximal occurrence sequence enabled at m. If $\sigma = \sigma_1\, \sigma_2$ and $m \xrightarrow{\sigma_1} m_1$, then σ_2 is a φ-maximal occurrence sequence enabled at m_1.*

Proof. The claim follows immediately from the definition of φ-maximal occurrence sequence. □

Some comments: All σ_i in Definition 1 are finite and succeeded by a controllable transition, chosen by the response function. If we get stuck in a deadlock, this is the case of a finite μ. We do not expect that after some σ_i only controllable transitions are enabled. Therefore, there might be situations where the user can fire a controllable transition but also the net can proceed autonomously. If liveness can only be enforced by passivity of the user in this case, the response function yields the empty set for the observed sequence.

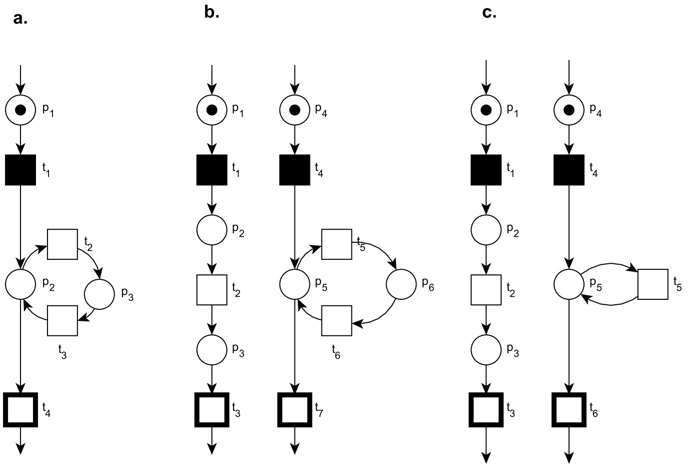

Fig. 2. Some example nets.

Figures 2.a, 2.b, and 2.c illustrate the weak fairness notion employed in our definition of φ-maximal occurrence sequence.

In the net shown in Fig. 2.a., after the controlled occurrence of t_1 the system can choose between t_2 and t_4. It can even always prefer t_2, and t_4 never occurs. Only strong fairness would imply that eventually t_4 can be observed, but our chosen notion of weak fairness does not. So t_4 is not observably live.

In Fig. 2.b., the net of Fig. 2.a. is extended by a concurrent sequence. Our weak fairness assumption implies that the left branch proceeds even if the right stays in an infinite loop. So transition t_3 is observably live.

Figure 2.c. illustrates the difference between our weak fairness and the one usually used in the literature, e.g. [13]. We do not expect that t_6 eventually occurs although it remains enabled at each marking reached after the occurrence of t_4.

However, since t_5 and t_6 share the input place p_5 we do have a conflict here. So again, t_3 is observably live and t_6 is not.

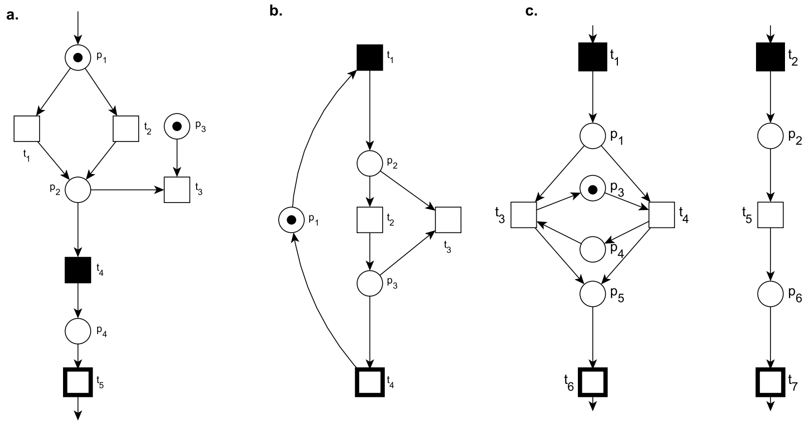

Fig. 3. Example nets.

In the net shown in Fig. 3.a, there is a conflict between t_3 and t_4. In this situation, even if the response function φ tells us to fire t_4 after t_1, we cannot be sure that t_4 will stay enabled since the unobservable transition t_3 might also fire. Since we cannot force t_4 to fire, t_5 is not observably live.

Now we define observable liveness as follows:

Definition 2. *An observable transition t of an observable place/transition net is* observably live *if there is a response function $\varphi_t \colon O^* \to 2^C$ such that, for each $m_0 \xrightarrow{\sigma_0} m$, each φ_t-maximal occurrence sequence enabled at m contains an occurrence of t. An observable place/transition net is observably live if all its observable transitions are observably live.*

In this definition, "an occurrence of t" can be replaced by "infinitely many occurrences of t", as in the definition of traditional liveness.

Theorem 1. *An observable transition t of an observable place/transition net is* observably live *if and only if there is a response function $\varphi_t \colon O^* \to 2^C$ such that, for each $m_0 \xrightarrow{\sigma_0} m$, each φ_t-maximal occurrence sequence enabled at m contains infinitely many occurrences of t.*

Proof. Clearly we only have to prove ⇒, because each occurrence sequence with infinitely many occurrences of t has at least one t-occurrence.

So assume observable liveness of t, i.e., a response function $\varphi_t\colon O^* \to 2^C$ such that, for each $m_0 \xrightarrow{\sigma_0'} m'$, each φ_t-maximal occurrence sequence enabled at m' contains an occurrence of t (notice that we replaced σ_0 by σ_0' and m by m').

Let $m_0 \xrightarrow{\sigma_0} m$ and let σ be a φ_t-maximal occurrence sequence enabled at m. We have to show that σ contains infinitely many occurrences of t. By assumption we know that σ contains at least one occurrence of t. Let σ_1 be the prefix of σ that ends after the first occurrence of t and let $\sigma = \sigma_1\, \sigma_2$. Then $m_0 \xrightarrow{\sigma_0\sigma_1} m_1$ for some marking m_1. This marking m_1 enables the φ_t-maximal occurrence sequence σ_2 by Lemma 1. Again using the assumption, σ_2 contains an occurrence of t.

The arbitrary repetition of this argument yields arbitrarily many occurrences of t in σ, whence this sequence must have infinitely many t-occurrences. □

5 Properties and Relations with Traditional Liveness

In this section, we provide some properties of observable liveness and relations to traditional liveness.

Lemma 2. *For each response function φ and each $m_0 \xrightarrow{\sigma_0} m$, there is a φ-maximal occurrence sequence enabled at m.*

Proof. In order to construct a φ-maximal occurrence sequence, we proceed iteratively. Assume that we constructed a finite sequence σ', enabled at m, in accordance with a), b) and c) of Def. 1 and let $m \xrightarrow{\sigma'} m'$. If m' enables an uncontrollable transition t or a controllable one which is in the current response set $\varphi(\overline{\sigma}_0\overline{\sigma'})$, then we append t to σ'. If there is more than one such candidate, we choose the least recently chosen such transition in order to ensure weak fairness.

If this is not possible then all transitions enabled after σ' are controllable and do not belong to $\varphi(\overline{\sigma}_0\overline{\sigma'})$, whence then σ' is a φ-maximal occurrence sequence by e) of Def. 1. □

Proposition 1. *Each observably live transition t is live.*

Proof. Since t is an observably live transition there is a response function φ_t such that for each $m_0 \xrightarrow{\sigma_0} m$, each φ_t-maximal occurrence sequence enabled at m includes t. By Lemma 2 there exists a φ_t-maximal occurrence sequence. This implies that, for each reachable marking m, there exists an occurrence sequence which enables t, and so t is live. □

Corollary 1. *An observably live net is live if all transitions are observable.* □

Notice that Cor. 1 does not hold without the assumption that all transitions are observable. The net shown in Fig. 3.b is not live since t_3 can never occur, but it is observably live.

The converse of Prop. 1 does not hold in general. Figure 2.a, if t_4 is assumed to be connected to t_1, shows a live net which is not observably live. However, if all transitions are controllable then liveness of t implies its observable liveness, as shown next:

Proposition 2. *If $O = C = T$ then observable liveness of a transition t coincides with its liveness.*

Proof. By Prop. 1, we only have to show the implication $\Leftarrow$.

Assume that t is live. We have to show that there is a response function $\varphi_t \colon O^* \to 2^C$ such that, for each $m_0 \xrightarrow{\sigma_0} m$, each φ_t-maximal occurrence sequence enabled at m contains an occurrence of t. Since t is live, there exists an occurrence sequence σ' enabled at m such that t is enabled after σ'.

Let $\sigma_0\, \sigma'\, t = \overline{\sigma_0 \sigma' t} = t_1 t_2 t_3 \ldots t_k$ and $m_0 \xrightarrow{\sigma_0 \sigma' t}$. We choose any response function with $\varphi_t(t_1 t_2 \ldots t_i) = \{t_{i+1}\}$ for $i = 0, 1, \ldots, k-1$. Since all transitions are controllable, the unique φ_t-maximal occurrence sequence consists of only controllable transitions. The σ_i (for $i = 1, 2, 3, \ldots$) given in Def. 1 are thus empty sequences, and so there is only one φ_t-maximal occurrence sequence for each m. □

Corollary 2. *If $O = C = T$, then observable liveness of a net coincides with liveness of the net.* □

Proposition 3. *Assume that in an observable net there is an infinite and weakly fair occurrence sequence σ without controllable transitions. Then each observable transition which does not appear in σ infinitely often is not observably live.*

Proof. Let $m_0 \xrightarrow{\sigma_0} m$ and assume that t is an observably live transition. There is a response function φ_t such that each φ_t-maximal occurrence sequence enabled at m contains an occurrence of t. So an infinite weakly fair occurrence sequence without controllable transitions σ which is enabled at some marking m' such that $m_0 \xrightarrow{\sigma_0} m \xrightarrow{\sigma'} m' \xrightarrow{\sigma}$ has to include t to be observably live. Since the sequence σ does not include any instance of t, t cannot be observably live. □

Corollary 3. *If an observable net without controllable transitions has an infinite and weakly fair occurrence sequence which does not include all the observable transitions then the net is not observably live.* □

6 Deterministic Uncontrollable Behavior

As seen before, a live net is not necessarily observably live. The main reason is that, for proving liveness, we can always choose an appropriate occurrence sequence enabling some transition t whereas for observable liveness this choice is only possible for controllable transitions (which are not in conflict with unobservable ones) and the net behaves arbitrarily elsewhere.

In this section, we show that the situation is different if the only choices to be made are among controllable transitions. This is not an unrealistic setting; the automated part of a system often behaves deterministically (but still concurrently), whereas the user model might allow for alternatives.

Formally, deterministic behavior is given in terms of the conflict-free property, to be defined next. Intuitively, a transition is conflict-free if it is never in conflict with any other transition; if both are enabled then they are enabled concurrently. Since “never" refers to reachable markings, the definition applies to a net with an initial marking and its state space and not only to its structure. However, each two transitions that are ever in conflict necessarily share an input place which is thus forward branching. With concurrent behavior we mean that two transitions do not compete for tokens. If a place carries more than one token, one could argue that two transitions in its post-set still can occur concurrently (see [17]). We take the stricter view that every two enabled transitions with a common input place (which can carry one or more tokens) are considered in conflict and not concurrent.

Definition 3. *A Petri net is conflict-free w.r.t. a transition u if, for each reachable marking m enabling u, every other transition v enabled at m is concurrent to u, i.e., ${}^\bullet u \cap {}^\bullet v = \emptyset$.*

Figure 3.c shows a net fragment which is conflict-free w.r.t. all its unobservable transitions. Notice that there is concurrency between these transitions. Notice also that forward branching places are possible, provided every reachable marking enables at most one output transition of a branching place. The following lemma will be used frequently in the sequel. It follows immediately from the occurrence rule.

Lemma 3. *Assume two transitions u and v of a net, both enabled at some marking m, such that ${}^\bullet u \cap {}^\bullet v = \emptyset$. Then m enables $u\,v$ as well as $v\,u$, and both sequences lead to the same marking.* □

A well-known result for conflict-free nets [10] is given by the following lemma. We provide a proof for the sake of self-containment, and since our lemma refers to a single conflict-free transition only.

Lemma 4. *If a Petri net is conflict-free w.r.t. a transition u, and some reachable marking m enables u as well as a sequence $\sigma\,u$ where u does not appear in σ, then m also enables the sequence $u\,\sigma$, and the occurrences of $\sigma\,u$ and of $u\,\sigma$ lead to the same marking.*

Proof. By induction on the length of σ.

Base: If σ is the empty sequence then nothing has to be shown.

Step: Assume $\sigma = v\,\sigma'$. We have $u \neq v$ because u does not appear in σ. By conflict-freeness w.r.t. u and since m enables both u and v, these transitions are concurrent. Therefore, and by Lemma 3, m also enables the sequences $v\,u$ and $v\,\sigma'\,u$. Let $m \xrightarrow{v} m'$.

The induction hypothesis can be applied to the marking m', enabling u and $\sigma'\,u$, yielding the sequence $u\,\sigma'$ enabled at m'. So $v\,u\,\sigma'$ is enabled at m. Again since u and v are concurrent and by Lemma 3, m also enables $u\,v\,\sigma'$, which is identical with $u\,\sigma$.

Since each transition occurs in $\sigma\, u$ and in $u\, \sigma$ the same number of times, and by the occurrence rule, the occurrences of these sequences lead to the same marking. □

Lemma 5. *If a Petri net is conflict-free w.r.t. a transition u, and some reachable marking m enables u as well as a sequence σ where u does not appear in σ, then m also enables the sequence $\sigma\, u$.*

Proof. By induction on the length of σ.

Base: If σ is the empty sequence then nothing has to be shown.

Step: Assume $\sigma = v\, \sigma'$. We have $u \neq v$ because u does not appear in σ. By conflict-freeness w.r.t. u and since m enables both u and v, these transitions are concurrent. Therefore, and by Lemma 3, m also enables the sequence $v\, u$. Let $m \xrightarrow{v} m'$.

The induction hypothesis can be applied to the marking m', enabling u and σ', yielding the sequence $\sigma'\, u$ enabled at m'. So $v\, \sigma'\, u$ is enabled at m. We have $v\, \sigma' = \sigma$, which finishes the proof. □

The following theorem constitutes the main result of this paper. It applies only to nets where the only possible conflicts occur between controllable transitions, i.e., to nets which are conflict-free w.r.t. all uncontrollable transitions. This rules out conflicts between two uncontrollable transitions as well as conflicts between controllable and uncontrollable transitions.

As a preparation, we need a couple of definitions and lemmas.

Definition 4. *An occurrence sequence σ enabled at a marking m is called* minimal towards *t, where t is a transition, if σ ends with t, contains no other occurrence of t, and no transition in σ can be postponed, i.e., $\sigma = \sigma'\, t$, t does not occur in σ', and σ cannot be divided as $\sigma = \mu'\, u\, \mu''$ for some transition u, $u \neq t$, such that $\mu'\, \mu''$ is enabled at m, too.*

A transition u can only occur if its input places carry tokens, and another transition v might have to occur before because it produces the token consumed by u. We then call the occurrence of v a causal predecessor of the occurrence of u. A minimal occurrence sequence towards a transition t contains one occurrence of t, its causal predecessors, the predecessors of these predecessors etc., and nothing else. In partially ordered runs, where causal dependence between transition occurrences is explicitly modeled by means of a partial order, this corresponds to a run containing the occurrence of t and all transition occurrences that precede t.

Definition 5. *Given a sequence σ, any deletion (i.e, replacement by the empty sequence) of elements in σ yields a* subsequence *of σ. Its* complementary sequence *is the sequence obtained from σ by deleting all elements that appear in the subsequence.*

This definition captures the case $\sigma = \sigma'\, \sigma''$ where σ' is a subsequence and σ'' is its complementary sequence (and vice versa), but is more general. For example, if $\sigma = t_1, t_2, \ldots, t_{2n}$, the sequence $t_1, t_3, \ldots, t_{2n-1}$ is a subsequence, and $t_2, t_4, \ldots t_{2n}$ its complementary sequence.

Lemma 6. *Assume a conflict-free net with a reachable marking m, a transition t and an occurrence sequence σ enabled at m that contains an occurrence of t. Then there exists a subsequence σ' of σ, enabled at m, which is minimal towards t. Moreover, if σ'' is the complementary subsequence, m enables $\sigma'\,\sigma''$.*

Proof. Define μ as the prefix of σ which ends with the first occurrence of t, and let $\overline{\mu}$ be the rest of σ. Clearly, μ is finite.

Assume that μ can be divided as $\mu = \mu'\, u\, \mu''$ such that $\mu'\, \mu''$ is enabled at m and u does not occur in μ''. By Lemma 5, we can shift u behind μ'' and thus obtain the sequence $\mu'\, \mu''\, u$. Still t occurs only once, being the last transition in μ''.

If u_1 is the rightmost transition (transition occurrence, respectively) in μ for which such a division is possible, we obtain from $\mu\, \overline{\mu}$ the sequence $\mu_1'\, \mu_1''\, u_1\, \overline{\mu}$. Let $\mu_2 = \mu_1'\, \mu_1''$. Now let u_2 be the rightmost transition with the same property for the sequence μ_2 and let $\mu_2 = \mu_2'\, u_2\, \mu_2''$. The same argument as above yields the sequence $\mu_2'\, \mu_2''\, u_2\, u_1 \overline{\mu}$. Exhaustive repetition of this procedure yields smaller and smaller sequences μ_i to be considered and eventually the sequence

$$\mu_k'\, \mu_k''\, u_k\, u_{k-i} \dots u_1\, \overline{\mu}$$

such that no further transition to be postponed can be found in $\mu_k'\, \mu_k''$. So this sequence is minimal towards t. By construction, it is a subsequence of σ, and $u_k\, u_{k-i} \dots u_1\, \overline{\mu}$ is the complementary subsequence. □

Starting with the next lemma, we additionally require 1-boundedness, i.e., we assume that no reachable marking assigns more than one token to a place.

Lemma 7. *Consider a 1-bounded and conflict-free Petri net with an arbitrary transition t. All initially enabled occurrence sequences which are minimal towards t lead to the same marking.*

Proof. Consider two occurrence sequences μ_1 and μ_2, both enabled at the initial marking, and both minimal towards t. We proceed by induction on the length of μ_1.

Base: The sequence μ_1 has only one element if and only if $\mu_1 = t$. So then t is initially enabled, and hence $\mu_1 = \mu_2 = t$.

Step: Assume that t is not initially enabled. We claim that there is an initially enabled transition u which appears in μ_1 as well as in μ_2, i.e., $\mu_1 = \mu_1'\, u\, \mu_1''$ and $\mu_2 = \mu_2'\, u\, \mu_2''$. When this claim is proven, we know by conflict-freeness that there are also initially enabled occurrence sequences $u\, \mu_1'\, \mu_1''$ and $u\, \mu_2'\, \mu_2''$. By the induction hypothesis applied to the (new initial) marking obtained by firing u and to the sequences $\mu_1'\, \mu_1''$ and $\mu_2'\, \mu_2''$, both sequences lead to the same marking, and we are finished.

So it remains to prove the claim, that some initially enabled transition occurs in μ_1 and in μ_2. We proceed indirectly and assume the contrary.

We again divide μ_2 as $\mu_2'\, \mu_2''$, now such that no transition of μ_2' occurs in μ_1 and the first transition in μ_2'', say v, occurs in μ_1. By assumption, v is not initially enabled. The sequence μ_2'' is not empty because both μ_1 and μ_2 contain t. We divide μ_1 as $\mu_1' \, \mu_1''$ such that μ_1'' begins with the first occurrence of v in μ_1.

Since v is not enabled initially, some place $s \in {}^\bullet v$ is initially unmarked. Since v is enabled after μ_1' and after μ_2', s carries a token after the occurrence of μ_1' and after the occurrence of μ_2'. By conflict-freeness and since the sets of occurring transitions in μ_1' and μ_2' are disjoint, we can also fire both, i.e. $\mu_1' \, \mu_2'$, from the initial marking. This yields a marking with two tokens on the place s, contradicting 1-boundedness. □

The proof of the above lemma also shows that all minimal sequences towards t have the same length, whence these sequences are exactly the sequences with minimal length containing an occurrence of t.

Now we are ready for the main result: liveness of a 1-bounded net implies observable liveness, provided the only conflict that can appear are between controllable transitions. Although this result might seem obvious at first sight, its proof is surprisingly involved. The core argument of the proof is that, in a live Petri net, for each transition t, every reachable marking m enables an occurrence sequence σ_m that includes an occurrence of t. If t is observable, then observable liveness requires that we can force t to occur by only providing a suitable response function φ_t which controls the behavior whenever there is a conflict. So an obvious idea is to define φ_t in such a way that always the next transition in σ_m is responded, if this transition is controllable. However, φ_t depends not on markings, but on observed sequences. That means, instead of t the user only knows the sequence of observable transitions of the initially enabled occurrence sequence σ_0 that leads to m. For this observed sequence, there might exist many sequences including unobservable transitions, and hence many different reached markings m, and so also many different occurrence sequences σ_m. Instead of the unknown occurrence sequence σ_0 we consider the set of all occurrence sequences μ_0 satisfying $\overline{\mu_0} = \overline{\sigma_0}$. Among these sequences we concentrate on the minimal ones. We will show that, if the net is 1-bounded, all these minimal occurrence sequences lead to the same marking which we call $m_{\overline{\sigma_0}}$. We will moreover show that m, the marking reached by the occurrence of σ_0 is reachable from $m_{\overline{\sigma_0}}$. However, these results only hold for conflict-free nets, and our considered net is not necessarily conflict-free. Since until now we only consider the behavior given by the observed transitions of σ_0, since all controllable transitions are observable and since conflicts only appear among controllable transitions, we can transform the considered net into a conflict-free one, without spoiling the relevant behavior. By liveness (of the original net), $m_{\overline{\sigma_0}}$ enables an occurrence sequence σ containing t. First, we look at the first observable transition in σ. Since there are no conflicts, every occurrence sequence starting at $m_{\overline{\sigma_0}}$ possessing a weak fairness assumption eventually has to enable u. If u is controllable, it might be in conflict with some other transition. In this case we set $\varphi_t(\overline{\sigma_0} = \{u\})$ so that, if u is controllable or not, also u eventually occurs. Fortunately, the distance between this marking and a marking enabling t is smaller than the distance between m and a marking enabling t, where distance is defined in terms of the number of needed observable transitions to reach one marking from the other. So we can repeat the above considerations, this way defining φ_t on the fly, until we eventually force t to occur.

Theorem 2. *If a 1-bounded observable Petri net, which is conflict-free w.r.t. all uncontrollable transitions, is live, then it is observably live.*

Proof. Consider a 1-bounded live observable Petri net which is conflict-free w.r.t. all uncontrollable transitions. We have to prove observable liveness, i.e., observable liveness of each observable transition t. So let t be an observable transition. To show observable liveness of t, we have to provide a response function φ_t such that, for each $m_0 \xrightarrow{\sigma_0} m$, each φ_t-maximal occurrence sequence σ enabled at m eventually contains t.

The considered net is only partially conflict-free, because there might be conflicts between controllable transitions. To be able to apply the previous lemmas, we make the net conflict-free for a given initially enabled sequence μ_0:

For each observable transition v we add a fresh place s_v, and an arc from s_v to v. Then v can only occur when s_v is marked. Now consider the sequence $\overline{\mu_0} = v_1 v_2 \dots v_k$. For each transition v_i in this sequence except the last (v_k) we add an arc from v_i to $s_{v_{i+1}}$. The place s_{v_1} gets an initial token, the other new places remain unmarked initially.

By construction, every reachable marking of this extended net marks at most one of the new places. Since each observable transition has such a place in its preset, always at most one observable transition is enabled. Since conflicts are only possible between controllable transitions and since each controllable transition is observable, thus no conflict can appear. Therefore, this extended net is conflict-free. By construction, the new initial marking enables μ_0 in the extended net.

The following claim also refers to an arbitrary initially enabled occurrence sequence μ_0 and to the net extended with the places as mentioned above. It generalizes Lemma 7:

Claim: All minimal occurrence sequences μ enabled at m_0 which satisfy $\overline{\mu} = \overline{\mu_0}$ lead to the same marking.

Proof of Claim: by induction on the length of $\overline{\mu_0}$.

Base: If $\overline{\mu_0}$ is empty then the only minimal sequence μ satisfying $\overline{\mu} = \overline{\mu_0}$ is the empty sequence.

Step: Let μ_1, μ_2 be minimal occurrence sequences enabled at m_0 which satisfy $\overline{\mu_1} = \overline{\mu_2} = \overline{\sigma_0}$.

Let $\mu_1 = u_1\, u_2\, \dots\, u_k$ and let u_i be the first observable transition in μ_1. Similarly, let $\mu_2 = v_1\, v_2\, \dots\, v_l$. Then the first observable transition v_j in μ_2 satisfies $u_i = v_j$.

We apply Lemma 6 to both sequences and thus obtain minimal subsequences towards u_i (v_j, respectively). By Lemma 7, both subsequences lead to the same marking. The induction hypothesis applies to the two complementary sequences. This ends the proof of the claim.

The unique (for a given μ_0) marking reached by a minimal sequence μ satisfying $\overline{\mu} = \overline{\mu_0}$ will be called m_{μ_0} in the sequel. Abusing notation, we call the same marking of the original net also m_{μ_0}, ignoring the additional places.

In the following, it will be useful to assume an arbitrary fixed total order $\prec$ on the set of observable transitions, i.e., if u and v are distinct observable transitions then either $u \prec v$ or $v \prec u$.

By liveness of the original net, for each initially enabled occurrence sequence μ_0 there exists (at least one) occurrence sequence μ_0' ending with t which is enabled by m_{μ_0} (in the original net). We assume that μ_0' has a minimal number of observable transitions among all sequences with the above property, i.e., $\overline{\mu_0'}$ has minimal length. Among these minimal sequences we assume moreover that the first observable transition in μ_0' is minimal w.r.t. $\prec$.

Now we define φ_t as follows: For each initially enabled occurrence sequence μ, we set $\varphi_t(\overline{\mu}) = \{u\}$ if $\overline{\mu'}$ begins with u and u is controllable, and $\varphi_t(\overline{\mu}) = \emptyset$ if $\overline{\mu'}$ begins with u and u is not controllable. Notice that $\overline{\mu'}$ contains t as its last transition and is hence not empty.

We now come back to the core of this proof and consider an arbitrary initially enabled occurrence sequence σ_0 which leads to a marking m. We have to show that each φ_t-maximal occurrence sequence enabled at m eventually contains t.

We consider a conflict-free variant of the net as before, but instead of considering only the sequence σ_0 we add places according to the sequence $\sigma_0\ \varphi_t(\sigma_0)$, i.e., we allow to fire the observable transition $\varphi_t(\sigma_0)$ after σ_0.

We proceed by induction on the number of observable transitions in σ_0' (which is defined above as an occurrence sequence ending with t enabled at m_{σ_0} with a minimal number of observable transitions).

Base: Assume that $\overline{\sigma_0'} = t$. Then there is an occurrence sequence σ_0', enabled at m_{σ_0} which eventually contains t (and no other observable transition). Since m is reachable from m_{σ_0} by Lemma 6, for each φ_t-maximal occurrence sequence enabled at m there is a suitable prefix yielding a φ_t-maximal occurrence sequence from m_{σ_0}. By conflict-freeness of the extended net and by weak fairness, each φ_t-maximal occurrence sequence enabled at m_{σ_0} eventually contains t. Hence this holds in particular for those passing through m.

Step: Assume that $\overline{\sigma_0'} = u_1\ u_2 \dots u_k\ t,\ k \geq 1$. Arguing as in the Base case, there is an occurrence sequence σ_0', enabled at m_{σ_0} which eventually contains u_1 (and no other observable transition). Since m is reachable from m_{σ_0} by Lemma 6, for each φ_t-maximal occurrence sequence enabled at m there is a suitable prefix yielding a φ_t-maximal occurrence sequence from m_{σ_0}. By conflict-freeness of the extended net and by weak fairness, each φ_t-maximal occurrence sequence enabled at m_{σ_0} eventually contains u_1. Hence this holds in particular for those passing m. So each φ_t-maximal occurrence sequence σ enabled at m can be divided as $\sigma_1 u_1 \sigma_2$ where σ_2 is again φ_t-maximal, and $\overline{\sigma_2}$ is shorter than $\overline{\sigma}$. By the induction hypothesis, σ_2 contains t, and therefore so does σ. □

In Fig. 4, we see one net with a conflict and a conflict-free net. The net shown in Fig. 4.a includes a conflict between a controllable transition and an uncontrollable transition (which is also unobservable). Although the net is live, since we cannot force t_1 to fire, both t_1 and t_3 are not observably live and so the net is not observably live. When the conflict in Fig. 4.a is resolved, we get the net shown in Fig. 4.b which is both live and observably live.

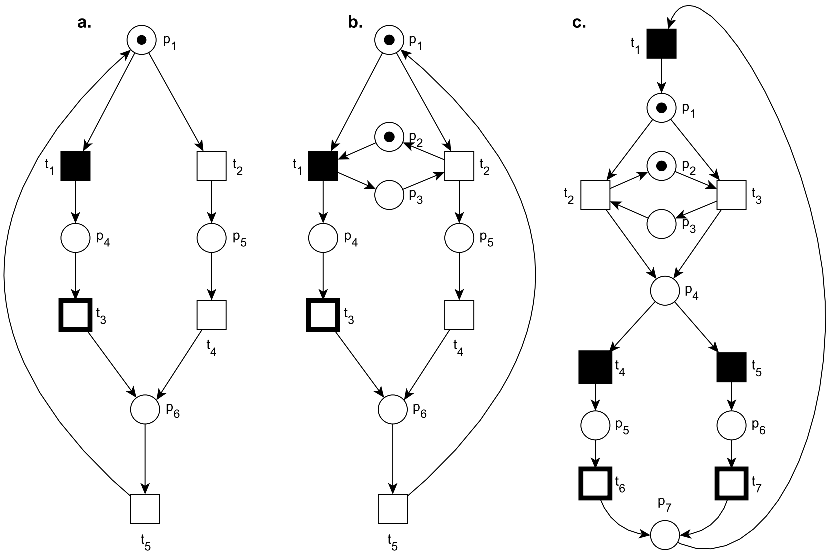

Fig. 4. **a**: a net with a conflict, **b**: a conflict-free net, **c**: a net which is conflict-free w.r.t. its uncontrollable transitions.

The net shown in Fig. 4.c is conflict-free w.r.t. all its uncontrollable transitions. Notice that there is a conflict between two controllable transitions t_4 and t_5. We can choose the related controllable transition in order to observe the occurrence of any observable transitions. The only choice is ours to make, the uncontrollable part of the machine behaves deterministically. This net is both live and observably live.

7 Conclusion and Related Work

Petri nets are widely used in software engineering for modeling and verifying software systems [3]. In this work, we provide a novel liveness notion which expresses the serviceability of a software system via an interface.

We considered a variant of Petri nets with observable transitions, where an observable transition can also be controllable. For further information about controllability and observability in Petri nets and using Petri nets in control theory, see [2, 15].

In analogy to the usual definition of liveness of a Petri net, we provided a notion for observable liveness, which roughly means that a user can always enforce the occurrence of any observable transition, only by stimulating the net by choosing appropriate enabled controllable transition. Therefore it is necessary to assume that also the uncontrollable part of a net proceeds, i.e., we assume that the net behaves weakly fair. A similar notion, *T-liveness*, yet for different motivations, is represented in [9]. One of the main differences is that only the fully controllable and observable nets are considered.

In general, liveness does not imply observable liveness and neither the opposite direction holds. This paper proves that for 1-bounded Petri nets with transitions that can be observable or additionally controllable, liveness implies observable liveness, where the latter means that control can force every transition to fire eventually from an arbitrary reachable marking – provided the net model behaves deterministically in its uncontrollable part. This control can only select enabled controllable transitions and is based only on the sequence of transitions observed so far. This way the result generalizes the obvious observation, that in a fully deterministic net a transition is live if and only if it eventually fires.

A future consideration refers to possible generalizations of our result. It clearly still holds when there is some limited nondeterminism in the uncontrolled part. For example, if two alternative uncontrollable transitions cause the same marking transformation, the result is not spoiled. More generally, we aim at defining an equivalence notion on nets, based on the respective observed behavior, which preserves observable liveness. Reduction rules, as defined e.g. in [1], [6] and [4] but also in many other papers, could be applied to the uncontrollable part leading to simpler but equivalent nets. However, there are obvious additional rules. For example, a rule that deletes a dead transition is sound w.r.t. the equivalence because dead uncontrollable transitions do not contribute to the observable liveness or non-liveness of the considered net.

As a future work, we plan to consider an automata approach for the implementation of the response function. The domain of the response function is defined infinite. In order to decide which controllable transitions can be fired next, an arbitrary history of observed transitions has to be considered. Often, a finite amount of the history is enough for this decision. If this is the case, an automata based approach can be used for the realization of the response function: the response then only depends on a state (of finitely many) of this automaton.

Concerning behavior, each run has an alternation between free choices of the machine (where in analysis all possibilities must be considered) and particular choices of the user. Therefore, describing the behavior with AND/OR-trees seems promising, maybe in combination with unfolding approaches. The partial order view would have obvious advantages to capture the progress assumption (that we called weak fairness) in a natural way [5, 14].

A final remark concerns the relation to Temporal Logics. Since liveness and all reachability questions in traditional Petri nets use existential quantification on paths (of the reachability graph), and therefore require Branching Time concepts, our approach explicates *reasons* for desired activities, i.e., transition occurrences. More precisely, as in the discussion of liveness in this paper, we distinguish uncontrollable alternatives and controllable choices, to be able to express that a certain activity (of a user) leads to the eventual occurrence of an event, no matter how the uncontrollable activities behave (but assuming they do not refuse work at all). This is clearly a Linear Time property. So, very roughly speaking, we translate Branching Time properties to Linear Time properties, and at the same time add details about controllability and observability to the system model. Future work aims at these transformations not only in the context of liveness properties but for arbitrary properties expressed by logical formulae. A related work has been done by Haddad et al. in [7].

Acknowledgements

The authors thank to Lucia Pomello and Luca Bernardinello for their valuable comments. This work was partially supported by MIUR and by MIUR - PRIN 2010/2011 grant 'Automi e Linguaggi Formali: Aspetti Matematici e Applicativi', code H41J12000190001.

References

1. Gérard Berthelot. Transformations and decompositions of nets. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Advances in Petri Nets*, volume 254 of *Lecture Notes in Computer Science*, pages 359–376. Springer, 1986.
2. Christos G. Cassandras and Stephane Lafortune. *Introduction to Discrete Event Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
3. Giovanni Denaro and Mauro Pezzè. Petri nets and software engineering. In Jörg Desel, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Lectures on Concurrency and Petri Nets*, volume 3098 of *Lecture Notes in Computer Science*, pages 439–466. Springer Berlin Heidelberg, 2004.
4. J. Desel and J. Esparza. *Free Choice Petri Nets*. Cambridge tracts in theoretical computer science. Cambridge University Press, 1995.
5. Jörg Desel, Hans-Michael Hanisch, Gabriel Juhás, Robert Lorenz, and Christian Neumair. A guide to modelling and control with modules of signal nets. In Hartmut Ehrig, Werner Damm, Jörg Desel, Martin Große-Rhode, Wolfgang Reif, Eckehard Schnieder, and Engelbert Westkämper, editors, *SoftSpez Final Report*, volume 3147 of *Lecture Notes in Computer Science*, pages 270–300. Springer, 2004.
6. Serge Haddad. A reduction theory for coloured nets. In Grzegorz Rozenberg, editor, *European Workshop on Applications and Theory in Petri Nets*, volume 424 of *Lecture Notes in Computer Science*, pages 209–235. Springer, 1988.
7. Serge Haddad, Rolf Hennicker, and MikaelH. Møller. Specification of asynchronous component systems with modal i/o-petri nets. In Martín Abadi and Alberto Lluch Lafuente, editors, *Trustworthy Global Computing*, Lecture Notes in Computer Science, pages 219–234. Springer International Publishing, 2014.
8. Lawrence E. Holloway, Bruce H. Krogh, and Alessandro Giua. A survey of petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems*, 7(2):151–190, 1997.
9. Marian V. Iordache and Panos J. Antsaklis. Design of t-liveness enforcing supervisors in petri nets. *IEEE Trans. Automat. Contr.*, 48(11):1962–1974, 2003.
10. L. H. Landweber and E. L. Robertson. Properties of conflict-free and persistent petri nets. *J. ACM*, 25(3):352–364, July 1978.
11. T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, volume 77, pages 541–580, April 1989.

12. Wolfgang Reisig. Partial order semantics versus interleaving semantics for csp-like languages and its impact on fairness. In *Proceedings of the 11th Colloquium on Automata, Languages and Programming*, pages 403–413, London, UK, UK, 1984. Springer-Verlag.
13. Wolfgang Reisig. *Elements of distributed algorithms: modeling and analysis with Petri nets*. Springer, 1998.
14. Wolfgang Reisig. *Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies*. Springer, 2013.
15. Manuel Silva. Half a century after carl adam petri's ph.d. thesis: A perspective on the field. *Annual Reviews in Control*, 37(2):191 – 219, 2013.
16. Wil M. P. van der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.
17. Rob J. van Glabbeek, Ursula Goltz, and Jens-Wolfhard Schicke. On causal semantics of petri nets. In Joost-Pieter Katoen and Barbara König, editors, *CONCUR*, volume 6901 of *Lecture Notes in Computer Science*, pages 43–59. Springer, 2011.