Newcastle University

# COMPUTING SCIENCE

## The Mutex Paradigm of Concurrency

Jetty Kleijn and Maciej Koutny

**TECHNICAL REPORT SERIES**

**No. CS-TR-1289** **November 2011**

TECHNICAL REPORT SERIES

**No. CS-TR-1289 November, 2011**

# The Mutex Paradigm of Concurrency

**J. Kleijn, M. Koutny**

## Abstract

Concurrency can be studied at different yet consistent levels of abstraction: from individual behavioural observations, to more abstract concurrent histories which can be represented by causality structures capturing intrinsic, invariant dependencies between executed actions, to system level devices such as Petri nets or process algebra expressions. Histories can then be understood as sets of closely related observations (here step sequences of executed actions). Depending on the nature of the observed relationships between executed actions involved in a single concurrent history, one may identify different concurrency paradigms underpinned by different kinds of causality structures (e.g. the true concurrency paradigm is underpinned by causal partial orders with each history comprising all step sequences consistent with some causal partial order). For some paradigms there exist closely matching system models such as elementary net systems (EN-systems) for the true concurrency paradigm, or elementary net systems with inhibitor arcs (ENI-systems) for a paradigm where simultaneity of executed actions does not imply their unorderedness. In this paper, we develop a system model fitting the least restrictive concurrency paradigm and its associated causality structures. To this end, we introduce ENI-systems with mutex arcs (ENIM-systems). Each mutex arc relates two transitions which cannot be executed simultaneously, but can be executed in any order. To link ENIM-systems with causality structures we develop a notion of process following a generic approach (semantical framework) which includes a method to generate causality structures from the new class of processes.


Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

# Bibliographical details


KLEIJN, J., KOUTNY, M.

The Mutex Paradigm of Concurrency
[By] J. Kleijn, M. Koutny
Newcastle upon Tyne: Newcastle University: Computing Science, 2011.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1289)


### Added entries


NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1289


### Abstract


Concurrency can be studied at different yet consistent levels of abstraction: from individual behavioural observations, to more abstract concurrent histories which can be represented by causality structures capturing intrinsic, invariant dependencies between executed actions, to system level devices such as Petri nets or process algebra expressions. Histories can then be understood as sets of closely related observations (here step sequences of executed actions). Depending on the nature of the observed relationships between executed actions involved in a single concurrent history, one may identify different concurrency paradigms underpinned by different kinds of causality structures (e.g. the true concurrency paradigm is underpinned by causal partial orders with each history comprising all step sequences consistent with some causal partial order). For some paradigms there exist closely matching system models such as elementary net systems (EN-systems) for the true concurrency paradigm, or elementary net systems with inhibitor arcs (ENI-systems) for a paradigm where simultaneity of executed actions does not imply their unorderedness.

In this paper, we develop a system model fitting the least restrictive concurrency paradigm and its associated causality structures. To this end, we introduce ENI-systems with mutex arcs (ENIM-systems). Each mutex arc relates two transitions which cannot be executed simultaneously, but can be executed in any order. To link ENIM-systems with causality structures we develop a notion of process following a generic approach (semantical framework) which includes a method to generate causality structures from the new class of processes.


### About the authors


Jetty Kleijn is a visiting fellow within the School of Computing Science, Newcastle University.

Maciej Koutny obtained his MSc (1982) and PhD (1984) from the Warsaw University of Technology. In 1985 he joined the then Computing Laboratory of the University of Newcastle upon Tyne to work as a Research Associate. In 1986 he became a Lecturer in Computing Science at Newcastle, and in 1994 was promoted to an established Readership at Newcastle. In 2000 he became a Professor of Computing Science.


### Suggested keywords


CONCURRENCY PARADIGMS
ELEMENTARY NET SYSTEMS
INHIBITOR ARCS
MUTEX ARCS
SEMANTICAL FRAMEWORK
STEP SEQUENCES
PROCESS AND CAUSALITY SEMANTICS

# The Mutex Paradigm of Concurrency

Jetty Kleijn[1] and Maciej Koutny[2]

[1] LIACS, Leiden University
P.O.Box 9512, NL-2300 RA Leiden, The Netherlands
kleijn@liacs.nl
[2] School of Computing Science, Newcastle University
Newcastle upon Tyne, NE1 7RU, United Kingdom
maciej.koutny@ncl.ac.uk

**Abstract.** Concurrency can be studied at different yet consistent levels of abstraction: from individual behavioural observations, to more abstract concurrent histories which can be represented by causality structures capturing intrinsic, invariant dependencies between executed actions, to system level devices such as Petri nets or process algebra expressions. Histories can then be understood as sets of closely related observations (here step sequences of executed actions). Depending on the nature of the observed relationships between executed actions involved in a single concurrent history, one may identify different *concurrency paradigms* underpinned by different kinds of causality structures (e.g., the true concurrency paradigm is underpinned by causal partial orders with each history comprising all step sequences consistent with some causal partial order). For some paradigms there exist closely matching system models such as elementary net systems (EN-systems) for the true concurrency paradigm, or elementary net systems with inhibitor arcs (ENI-systems) for a paradigm where simultaneity of executed actions does not imply their unorderedness.
In this paper, we develop a system model fitting the least restrictive concurrency paradigm and its associated causality structures. To this end, we introduce ENI-systems with *mutex* arcs (ENIM-systems). Each mutex arc relates two transitions which cannot be executed simultaneously, but can be executed in any order. To link ENIM-systems with causality structures we develop a notion of process following a generic approach (semantical framework) which includes a method to generate causality structures from the new class of processes.
**Keywords:** concurrency paradigms, elementary net systems, inhibitor arcs, mutex arcs, semantical framework, step sequences, process and causality semantics.

## 1 Introduction

Concurrency can be studied at different levels of abstraction, from the lowest level dealing with individual behavioural runs (observations), to the intermediate level of more abstract concurrent histories which can be represented by causality structures (or order structures) capturing intrinsic (invariant) dependencies between executed actions, to the highest system level dealing with devices such as Petri nets or process algebra expressions. Clearly, different descriptions of concurrent systems and their behaviours at these distinct levels of abstractions must be consistent and their mutual relationships well understood.

Abstract concurrent histories can be understood as sets of closely related observations. In this paper, each observation will be a *step sequence* (or stratified poset) of executed actions. For example, Figure 1($a$) depicts an EN-system generating three step sequences involving the executions of transitions $a$, $b$ and $c$, viz. $\sigma_1 = \{a,b\}\{c\}$, $\sigma_2 = \{a\}\{b\}\{c\}$ and $\sigma_3 = \{b\}\{a\}\{c\}$. They can be seen as belonging to a single abstract history $\Delta_1 = \{\sigma_1, \sigma_2, \sigma_3\}$ underpinned by a causal partial order in which $a$ and $b$ are unordered and they both precede $c$. From our point of view it is also important to note that $\Delta_1$ adheres to the *true concurrency paradigm* captured by the following general statement:

> Given two executed actions (e.g., $a$ and $b$ in $\Delta_1$), they can be observed as simultaneous (e.g., in $\sigma_1$) $\Longleftrightarrow$ they can be observed in both orders (e.g., $a$ before $b$ in $\sigma_2$, and $b$ before $a$ in $\sigma_3$). (TRUECON)

Concurrent histories adhering to such a paradigm are underpinned by *causal partial orders*, in the sense that each history comprises *all* step sequences consistent with some causal partial order on executed actions. Elementary net systems [18] (EN-systems) provide a fundamental and natural system level model for the true concurrency paradigm. A suitable link between an EN-system and histories like $\Delta_1$ can be formalised using the notion of a process or occurrence net [1, 18]. Full consistency between the three levels of abstraction can then be established within a generic approach (the *semantical framework* of [14]) aimed at fitting together systems (nets from a certain class of Petri nets), abstract histories and individual observations.

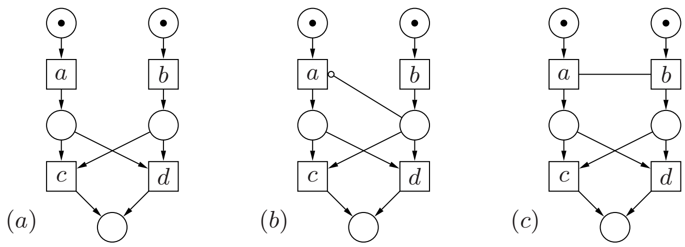


**Fig. 1.** EN-system ($a$); ENI-system with an inhibitor arc joining the output place of transition $b$ with transition $a$ implying that $a$ cannot be fired if the output place of $b$ is not empty ($b$); and ENIM-system with a mutex arc between transitions $a$ and $b$ implying that the two transitions cannot be fired in the same step ($c$).

Depending on the exact nature of relationships holding for actions executed in a single concurrent history, similar to (TRUECON) recalled above, [9] identified eight general concurrency paradigms, $\pi_1$–$\pi_8$, with true concurrency being another name for $\pi_8$. Another paradigm is $\pi_3$ characterised by (TRUECON) with $\Longleftrightarrow$ replaced by $\Longleftarrow$. This paradigm has a natural system level counterpart provided by elementary net systems with inhibitor arcs (ENI-systems). Note that inhibitor arcs (as well as activator arcs used later in this paper) are well suited to model situations involving testing for a specific condition, rather than producing and consuming resources, and proved to be useful in areas such as communication protocols [2], performance analysis [4] and concurrent programming [5].

For example, Figure 1($b$) depicts an ENI-system generating two step sequences involving transitions $a$, $b$ and $c$, viz. $\sigma_1 = \{a, b\}\{c\}$ and $\sigma_2 = \{a\}\{b\}\{c\}$. The two step sequences can be seen as belonging to the abstract history $\Delta_2 = \{\sigma_1, \sigma_2\}$ adhering to paradigm $\pi_3$, but *not* adhering to paradigm $\pi_8$ as there is no step sequence in $\Delta_2$ in which $b$ is observed before $a$ (even though $a$ and $b$ are observed in $\sigma_1$ as simultaneous). Another consequence of the latter fact is that paradigm $\pi_3$ histories are underpinned *not* by causal partial orders but rather by causality structures introduced in [10] — called *stratified order structures* — based on causal partial orders and, in addition, weak causal partial orders. Again, full consistency between the three levels of abstraction can then be established within the semantical framework of [14].

In this paper, we focus on $\pi_1$ which simply admits all concurrent histories and is the least restrictive of the eight general paradigms of concurrency investigated in [9]. Concurrent histories conforming to paradigm $\pi_1$ are underpinned by yet another kind of causality structures introduced in [9] — called *generalised stratified order structures* — based on weak causal partial orders and *commutativity*. Intuitively, two executed actions commute if they may be observed in any order in step sequences belonging to a history, but they are never observed as simultaneous.

The aim of this paper is to develop the hitherto missing system level net model matching paradigm $\pi_1$. The proposed solution consists in extending ENI-systems with *mutex* arcs, where each mutex arc relates two transitions which cannot be executed simultaneously, even when they can be executed in any order. Mutex arcs are therefore a system level device implementing commutativity (for an early attempt aimed at capturing such a feature see [16]). The resulting ENIM-systems provide a natural match for histories conforming to paradigm $\pi_1$, in the same way as EN-systems and ENI-systems provided a natural match for histories conforming to paradigms $\pi_8$ and $\pi_3$, respectively.

For example, Figure 1($c$) depicts an ENIM-system generating two step sequences involving transitions $a$, $b$ and $c$, viz. $\sigma_2 = \{a\}\{b\}\{c\}$ and $\sigma_3 = \{b\}\{a\}\{c\}$. They belong to an abstract history $\Delta_3 = \{\sigma_2, \sigma_3\}$ adhering to paradigm $\pi_1$, in which the executions of $a$ and $b$ commute. Clearly, $\Delta_3$ does *not* conform to paradigms $\pi_8$ and $\pi_3$ as there is no step sequence in $\Delta_3$ in which $a$ and $b$ are observed as simultaneous.

We prove full consistency between the three levels of abstraction for paradigm $\pi_1$. To this end, we once more use the semantical framework of [14]. In doing so, we define processes of ENIM-systems and demonstrate that these new processes provide the desired link with the generalised stratified order structures of paradigm $\pi_1$. To achieve this we introduce a notion of gso-closure making it possible to construct generalised stratified order structures from more basic relationships between executed actions involved in processes of ENIM-systems. Note that ENIM-systems were first sketched in [12] however this preliminary presentation was still incomplete. In this paper, we provide the missing details and harmonise the treatment of paradigm $\pi_1$ with those of paradigms $\pi_8$ and $\pi_3$.

The paper is organised in the following way. To motivate our subsequent study of causality in nets with mutex arcs, we first briefly recall the approach of [9] which investigates general concurrency paradigms and the associated causality structures. We then recall the semantical framework of [14]. After that we formally introduce ENIM-systems and develop their process semantics. The paper concludes with the proofs of various results which collectively justify our claim that ENIM-systems provide a fully satisfactory system model for paradigm $\pi_1$.

**Basic notions and notations**

Composing functions $f : X \to 2^Y$ and $g : Y \to 2^Z$ is defined by $g \circ f(x) \stackrel{\text{df}}{=} \bigcup_{y\in f(x)} g(y)$, for all $x \in X$. Restricting function $f$ to a sub-domain $Z$ is denoted by $f|_Z$. Relation $P \subseteq X \times X$ is irreflexive if $(x, x) \notin P$ for all $x \in X$; transitive if $P \circ P \subseteq P$; its transitive and reflexive closure is denoted by $P^*$; and its symmetric closure by $P^{\mathsf{sym}} \stackrel{\text{df}}{=} P \cup P^{-1}$.

A *relational structure* is a tuple $R \stackrel{\text{df}}{=} (X, Q_1, \ldots, Q_n)$ where $X$ is a finite *domain*, and the $Q_i$'s are binary relations on $X$ (we can select its components using the subscript $R$, e.g., $X_R$). Relational tuples, $R$ and $R'$, are *isomorphic* if there is a bijection $\xi$ from the domain of $R$ to the domain of $R'$ such that if we replace throughout $R$ each element $a$ by $\xi(a)$ then the result is $R'$. For relational structures with the same domain and arity, $R$ and $R'$, we write $R \subseteq R'$ if the subset inclusion holds component-wise. The intersection $\bigcap \mathcal{R}$ of a non-empty set $\mathcal{R}$ of relational structures with the same arity and domain is defined component-wise.

We *assume* that all sets in this paper are *labelled*, with the default labelling being the identity function. If the labelling is irrelevant for a definition or result, it may be omitted. If two domains are said to be the same, their labellings are identical.

A *partially ordered set* (or poset) is a relational structure $po \stackrel{\text{df}}{=} (X, \prec)$ consisting of a finite set $X$ and a transitive irreflexive relation $\prec$ on $X$. Two distinct elements $a, b$ of $X$ are *unordered*, $a \frown b$, if neither $a \prec b$ nor $b \prec a$ holds. Moreover, $a \precsim b$ if $a \prec b$ or $a \frown b$. Poset $po$ is *total* if the relation $\frown$ is empty, and *stratified* if $\simeq$ is an equivalence relation, where $a \simeq b$ if $a \frown b$ or $a = b$. Note that if a poset is interpreted as an observation of concurrent system behaviour, then $a \prec b$ means that $a$ was observed before $b$, while $a \simeq b$ means that $a$ and $b$ were observed as simultaneous.

A *step sequence* is a sequence of non-empty sets $\sigma \stackrel{\text{df}}{=} X_1 \ldots X_k$ $(k \geq 0)$. We will call $\sigma$ *singular* if the steps $X_i$ are mutually disjoint. In such a case, we have that $\mathsf{spo}(\sigma) \stackrel{\text{df}}{=} (\bigcup_i X_i, \bigcup_{i<j} X_i \times X_j)$ is a stratified poset. Conversely, each stratified poset $spo$ induces a unique singular step sequence $\mathsf{steps}(spo) = X_1 \ldots X_k$, with each $X_i$ being an equivalence class of $\simeq$ and $(X_i \times X_j) \subseteq\prec$ for all $i < j$, satisfying $spo = \mathsf{spo}(\mathsf{steps}(spo))$. We will identify each stratified poset $spo$ with $\mathsf{steps}(spo)$ or, equivalently, each singular step sequence $\sigma$ with $\mathsf{spo}(\sigma)$.

## 2 Paradigms of concurrency and order structures

Let $\Delta$ be a non-empty set of *stratified posets* (or, equivalently, singular step sequences) with the same domain $X$ (or $X_\Delta$).[3] Intuitively, each poset in $\Delta$ is an observation of an abstract history of a hypothetical concurrent system. Following the true concurrency

[3] Note that [9] also considered total and interval poset observations.

approach, [9] attempted to represent $\Delta$ using relational invariants on $X$. The basic idea was to capture situations where knowing some (or all) invariant relationships between executed actions involved in $\Delta$ would be sufficient to reconstruct the entire set of observations $\Delta$.

The approach of [9] identified a number of *fundamental invariants* which can be attributed to the observations in $\Delta$, each invariant describing a relationship between pairs of executed actions which is repeated in all the observations of $\Delta$. In particular, $\prec_\Delta$ comprises all pairs $(a, b)$ such that $a$ precedes $b$ in every poset belonging to $\Delta$; in other words, $\prec_\Delta$ represents *causality*. Other fundamental invariants are: $\rightleftharpoons_\Delta$ (*commutativity*, where $a \rightleftharpoons_\Delta b$ means that $a$ and $b$ are never simultaneous), $\sqsubset_\Delta$ (*weak causality*, where $a \sqsubset_\Delta b$ means that $a$ is never observed after $b$) and $\bowtie_\Delta$ (*synchronisation*, where $a \bowtie_\Delta b$ means that $a$ and $b$ are always simultaneous). One can show that knowing $\rightleftharpoons_\Delta$ and $\sqsubset_\Delta$ is always sufficient to reconstruct $\Delta$. This is done assuming that $\Delta$ is *invariant-closed* in the sense that $\Delta$ comprises all stratified posets $spo$ with the domain $X$ which respect all the fundamental invariants generated by $\Delta$, e.g., $a \prec_\Delta b$ implies $a \prec_{spo} b$, and $a \sqsubset_\Delta b$ implies $a \not\succ_{spo} b$. We then call each invariant-closed set of observations a *(concurrent) history*. Being invariant-closed is a natural assumption when constructing an abstract view of a possibly large set of individual observations, and has always been tacitly assumed in the causal partial order view of concurrent computation.

Depending on the underlying system model of concurrent computation, some additional constraints on histories $\Delta$ may be added. In particular, each design may adhere to the 'diagonal rule' — or 'diamond property' — by which simultaneity is the same as the possibility of occurring in any order, i.e., for all $a, b \in X$:

$$(\exists spo \in \Delta :\ a \frown_{spo} b) \iff (\exists spo \in \Delta :\ a \prec_{spo} b) \wedge (\exists spo \in \Delta :\ b \prec_{spo} a)\ . \quad (\pi_8)$$

For example, $\pi_8$ is satisfied by concurrent histories generated by EN-systems.

Constraints like $\pi_8$ — called *paradigms* in [8, 9] — are essentially suppositions or statements about the intended treatment of simultaneity and, moreover, allow one to simplify the invariant representation of a history $\Delta$. In particular, if $\Delta$ satisfies $\pi_8$ then one can reconstruct $\Delta$ using just causality $\prec_\Delta$ (which is always equal to the intersection of $\rightleftharpoons_\Delta$ and $\sqsubset_\Delta$). This is the essence of the true concurrency paradigm based on causal partial order.

In general, knowing $\prec_\Delta$ is insufficient to reconstruct $\Delta$. For example, if we weaken $\pi_8$ to the paradigm:

$$(\exists spo \in \Delta :\ a \prec_{spo} b) \wedge (\exists spo \in \Delta :\ b \prec_{spo} a) \Longrightarrow (\exists spo \in \Delta :\ a \frown_{spo} b) \quad (\pi_3)$$

then one needs to enhance causality with weak causality $\sqsubset_\Delta$ to provide an invariant representation of $\Delta$. The resulting relational structure $(X, \prec_\Delta, \sqsubset_\Delta)$ is an instance of the following notion.

**Definition 1 (stratified order structure [6, 11, 13, 14]).** *A* stratified order structure *(or* SO*-structure) is a relational structure* $sos \stackrel{\text{df}}{=} (X, \prec, \sqsubset)$ *where* $\prec$ *and* $\sqsubset$ *are binary relations on* $X$ *such that, for all* $a, b, c \in X$*:*

*S1:* $a \not\sqsubset a$

*S2:* $a \prec b \implies a \sqsubset b$

*S3:* $a \sqsubset b \sqsubset c \ \wedge\ a \neq c \implies a \sqsubset c$

*S4:* $a \sqsubset b \prec c \ \vee\ a \prec b \sqsubset c \implies a \prec c\ .$

The axioms imply that $\prec$ is a partial order relation, and that $a \prec b$ implies $b \not\sqsubset a$. The relation $\prec$ represents the 'earlier than' relationship on the domain of $so$, and the relation $\sqsubset$ the 'not later than' relationship. The four axioms capture the mutual relationship between the 'earlier than' and 'not later than' relations between executed actions.

For every stratified poset $spo$, $\mathsf{sos}(spo) \stackrel{\text{df}}{=} (X_{spo}, \prec_{spo}, \gtrsim_{spo})$ is an SO-structure. Moreover, $spo$ is a *stratified poset extension* of an SO-structure $sos$ whenever $sos \subseteq \mathsf{sos}(spo)$. We denote this by $spo \in \mathsf{ext}(sos)$. Following Szpilrajn's Theorem [19] that any poset can be reconstructed by intersecting its total extensions, we have that any SO-structure can be reconstructed from its stratified poset extensions.

**Theorem 1 ([11]).** *If $sos$ is an SO-structure then $\mathsf{ext}(sos) \neq \varnothing$ and:*

$$sos = \bigcap \{\mathsf{sos}(spo) \mid spo \in \mathsf{ext}(sos)\} \ .$$

*Moreover, if $\mathcal{SPO}$ is a non-empty set of stratified posets with the same domain, then $\bigcap\{\mathsf{sos}(spo) \mid spo \in \mathcal{SPO}\}$ is an SO-structure.* □

The set of stratified poset extensions of an SO-structure is a concurrent history satisfying paradigm $\pi_3$ [9]. Moreover, if a concurrent history $\Delta$ satisfies $\pi_3$, then $\Delta = \mathsf{ext}(X_\Delta, \prec_\Delta, \sqsubset_\Delta)$. Hence each abstract history $\Delta$ adhering to paradigm $\pi_3$ can be represented by the SO-structure $(X_\Delta, \prec_\Delta, \sqsubset_\Delta)$ [8].

If $\Delta$ fails to satisfy $\pi_3$, knowing $(X_\Delta, \prec_\Delta, \sqsubset_\Delta)$ may be insufficient to reconstruct $\Delta$. In the case of paradigm $\pi_1$ which places no restrictions of the kind captured by $\pi_8$ or $\pi_3$ (i.e., $\Delta$ is only assumed to be invariant-closed), one needs to use *general* SO*-structures (*GSO*-structures).*

**Definition 2 (GSO-structure [7,8]).** *A relational structure $gsos \stackrel{\text{df}}{=} (X, \rightleftharpoons, \sqsubset)$ is a* GSO-structure *if $\mathsf{sos}(gsos) \stackrel{\text{df}}{=} (X, \rightleftharpoons \cap \sqsubset, \sqsubset)$ is an* SO*-structure and the relation $\rightleftharpoons$ is symmetric and irreflexive.* ◇

In the above, $\rightleftharpoons$ represents the 'earlier than or later than, but never simultaneous' relationship, while $\sqsubset$ again represents the 'not later than' relationship.

For a stratified poset $spo$, $\mathsf{gsos}(spo) \stackrel{\text{df}}{=} (X_{spo}, \prec_{spo}^{\mathsf{sym}}, \gtrsim_{spo})$ is a GSO-structure. Also, $spo$ is a *stratified poset extension* of a GSO-structure $gsos$ if $gsos \subseteq \mathsf{gsos}(spo)$. We denote this by $spo \in \mathsf{ext}(gsos)$.

Each GSO-structure can be reconstructed from its stratified poset extensions, leading to another generalisation of Szpilrajn's Theorem.

**Theorem 2 ([7,8]).** *If $gsos$ is a* GSO*-structure then $\mathsf{ext}(gsos) \neq \varnothing$ and:*

$$gsos = \bigcap \{\mathsf{gsos}(spo) \mid spo \in \mathsf{ext}(gsos)\} \ .$$

*Moreover, if $\mathcal{SPO}$ is a non-empty set of stratified posets with the same domain, then $\bigcap\{\mathsf{gsos}(spo) \mid spo \in \mathcal{SPO}\}$ is a* GSO*-structure.* □

The set of stratified poset extensions of a GSO-structure is a concurrent history. Moreover, if $\Delta$ is a concurrent history, then $\Delta = \mathsf{ext}(X_\Delta, \rightleftharpoons_\Delta, \sqsubset_\Delta)$. Hence each abstract history $\Delta$ can be represented by the GSO-structure $(X_\Delta, \rightleftharpoons_\Delta, \sqsubset_\Delta)$ [8].

As already mentioned, paradigm $\pi_8$ and its associated causal posets $(X, \prec_\Delta)$ provide a match for concurrent histories generated by EN-systems. Similarly, one can show that paradigm $\pi_3$ and its associated SO-structures $(X, \prec_\Delta, \sqsubset_\Delta)$ provide a match for concurrent histories generated by ENI-systems. In this paper, we will extend ENI-systems with mutex arcs. The resulting ENIM-systems will provide a match for the most general paradigm $\pi_1$, and the notion of an abstract history of an ENIM-system will be captured through GSO-structures.

### Constructing order structures

We end this section describing ways of constructing SO-structures and GSO-structures from more basic, or direct, relationships. The idea is to proceed similarly as when constructing posets from acyclic relations through the operation of transitive closure. The definitions and results in this section are a new contribution to the theory of GSO-structures. Moreover, they are central for proving our subsequent results concerning nets with mutex arcs.

We first recall how the notion of transitive closure was lifted to the level of SO-structures. Let $\mu = (X, \prec, \sqsubset)$ be a relational structure (not necessarily an SO-structure). Intuitively, $\prec$ indicates which of the executed actions in $X$ are directly causally related, and $\sqsubset$ which are directly weakly causally related. The *so-closure* of $\mu$ is defined as:

$$\mu^{\mathsf{so}} \stackrel{\text{df}}{=} (X, \alpha, \gamma \setminus id_X)$$

where $\gamma \stackrel{\text{df}}{=} (\prec \cup \sqsubset)^*$, $\alpha \stackrel{\text{df}}{=} \gamma \circ \prec \circ \gamma$ and $id_X$ is the identity on $X$. Moreover, $\mu$ is *so-acyclic* if $\alpha$ is irreflexive. In such a case, $\mu^{\mathsf{so}}$ is an SO-structure [10].

We will now show how to construct GSO-structures. Let $\rho = (X, \prec, \sqsubset, \rightleftharpoons)$ be a relational structure. In addition to the two relations appearing also in the $\mu$ above, $\rightleftharpoons$ indicates which of the executed actions may be observed in any order, but not simultaneously. The *gso-closure* of $\rho$ is defined as:

$$\rho^{\mathsf{gso}} \stackrel{\text{df}}{=} (X, \psi, \gamma \setminus id_X)$$

where $\psi \stackrel{\text{df}}{=} \alpha^{\mathsf{sym}} \cup \beta^{\mathsf{sym}} \cup \rightleftharpoons$ with $\beta \stackrel{\text{df}}{=} \sqsubset^* \circ (\rightleftharpoons \cap \sqsubset^*) \circ \sqsubset^*$, in addition to $\alpha$ and $\gamma$ being defined as for $\mu^{\mathsf{so}}$. Moreover, $\rho$ is *gso-acyclic* if $\psi$ is irreflexive and symmetric.

**Proposition 1.** *If $\rho$ is gso-acyclic then $\rho^{\mathsf{gso}}$ is a* GSO-*structure.*

*Proof.* We first observe that $(i)$ $\gamma = (\gamma \circ \prec \circ \gamma) \cup \sqsubset^*$, $(ii)$ $\gamma = \gamma \circ \gamma$, and $(iii)$ $\alpha \cup \beta \subseteq \gamma$. Moreover, $(iv)$ $\alpha^{-1} \cap \gamma = \varnothing$ and $(v)$ $\beta^{-1} \cap \gamma = \varnothing$. The two latter properties follow from $(ii)$ and irreflexivity of $\alpha$ and $\beta$ (which in turn follows from irreflexivity of $\psi$ and $\alpha \cup \beta \subseteq \psi$).

Clearly, $\gamma \setminus id_X$ is irreflexive, and $\psi$ is symmetric and irreflexive (by gso-acyclicity). Hence it suffices to show that $sos \stackrel{\text{df}}{=} (X, \psi \cap (\gamma \setminus id_X), \gamma \setminus id_X)$ is an SO-structure.

*S1&S2:* Clearly, $\gamma \setminus id_X$ is irreflexive, and $\psi \cap (\gamma \setminus id_X) \subseteq \gamma \setminus id_X$.

*S3:* $(\gamma \setminus id_X) \circ (\gamma \setminus id_X) \subseteq \gamma$ holds by $(ii)$.

*S4:* We will show that $(\psi\cap(\gamma\setminus id_X))\circ(\gamma\setminus id_X) \subseteq \psi\cap(\gamma\setminus id_X)$. Our first observation is that $\psi\cap(\gamma\setminus id_X) = \psi\cap\gamma$ as $\psi$ is irreflexive. Hence it suffices to show that $(\psi\cap\gamma)\circ\gamma \subseteq \psi \cap \gamma$. We have that:

$$\psi \cap \gamma = (\alpha^{\mathsf{sym}} \cup \beta^{\mathsf{sym}} \cup \rightleftharpoons) \cap \gamma =_{(iv,v)} (\alpha \cup \beta \cup \rightleftharpoons) \cap \gamma =_{(iii)} \alpha \cup \beta \cup (\rightleftharpoons \cap \gamma)$$

which in turn implies that:

$$\begin{aligned}
&(\psi \cap \gamma) \circ \gamma = (\alpha \cup \beta \cup (\rightleftharpoons \cap \gamma)) \circ \gamma = \\
&(\alpha \circ \gamma) \cup (\beta \circ \gamma) \cup ((\rightleftharpoons \cap \gamma) \circ \gamma) =_{(i,ii)} \\
&\alpha \cup (\beta \circ ((\gamma \circ \prec \circ \gamma) \cup \sqsubset^*)) \cup ((\rightleftharpoons \cap \gamma) \circ ((\gamma \circ \prec \circ \gamma) \cup \sqsubset^*)) = \\
&\alpha \cup (\beta \circ \gamma \circ \prec \circ \gamma) \cup (\beta \circ \sqsubset^*) \cup ((\rightleftharpoons \cap \gamma) \circ \gamma \circ \prec \circ \gamma) \cup ((\rightleftharpoons \cap \gamma) \circ \sqsubset^*) \subseteq \\
&\alpha \cup \alpha \cup \beta \cup \alpha \cup ((\rightleftharpoons \cap \gamma) \circ \sqsubset^*) = \\
&\alpha \cup \beta \cup ((\rightleftharpoons \cap ((\gamma \circ \prec \circ \gamma) \cup \sqsubset^*)) \circ \sqsubset^*) = \\
&\alpha \cup \beta \cup ((\rightleftharpoons \cap (\gamma \circ \prec \circ \gamma)) \circ \sqsubset^*) \cup ((\rightleftharpoons \cap \sqsubset^*) \circ \sqsubset^*) \subseteq \\
&\alpha \cup \beta \cup \alpha \cup \beta = \alpha \cup \beta \subseteq_{(iii)} \psi \cap \gamma\ .
\end{aligned}$$

As a result, *S4* holds as its other part is symmetric. □

**Proposition 2.** *If $\rho$ is gso-acyclic then $(X, \prec, \sqsubset)$ is an so-acyclic relational structure and:*

$$\mathsf{ext}(\rho^{\mathsf{gso}}) = \{spo \in \mathsf{ext}((X, \prec, \sqsubset)^{\mathsf{so}}) \mid \frown_{spo} \cap \rightleftharpoons = \varnothing\}\ .$$

*Proof.* That $(X, \prec, \sqsubset)$ is so-acyclic follows immediately from irreflexivity of $\psi$ and $\alpha \subseteq \psi$.

$(\subseteq)$ Let $spo \in \mathsf{ext}(\rho^{\mathsf{gso}})$. Then $\psi \subseteq \prec_{spo}^{\mathsf{sym}}$ and $\gamma \setminus id_X \subseteq \sqsubset_{spo}$. Thus $\alpha^{\mathsf{sym}} \subseteq \prec_{spo}^{\mathsf{sym}}$ and $\gamma \setminus id_X \subseteq \sqsubset_{spo}$ which together with $\alpha \subseteq \gamma$ and irreflexivity of $\alpha$ imply $\alpha \subseteq \prec_{spo}$. Hence $spo \in \mathsf{ext}((X, \prec, \sqsubset)^{\mathsf{so}})$. Moreover, we have $\rightleftharpoons \subseteq \prec_{spo}^{\mathsf{sym}}$ implying $\frown_{spo} \cap \rightleftharpoons = \varnothing$.

$(\supseteq)$ Let $spo \in \mathsf{ext}((X, \prec, \sqsubset)^{\mathsf{so}})$ and $\frown_{spo} \cap \rightleftharpoons = \varnothing$. Then $\alpha \subseteq \prec_{spo}$ and $\gamma \setminus id_X \subseteq \sqsubset_{spo}$ and $\rightleftharpoons \subseteq \prec_{spo}^{\mathsf{sym}}$ (by $\frown_{spo} \cap \rightleftharpoons = \varnothing$ and irreflexivity of $\psi$ and $\rightleftharpoons \subseteq \psi$). Therefore, it suffices to show that $\beta \subseteq \prec_{spo}$.

Suppose that $(a, b) \in \beta$. Then there are $x, y$ such that $a \sqsubset^* x$ and $(x, y) \in \rightleftharpoons \cap \sqsubset^*$ and $y \sqsubset^* b$. By $(x, y) \in \rightleftharpoons$ and irreflexivity of $\psi$ and $\rightleftharpoons \subseteq \psi$, we have that $x \neq y$. Thus, by the fact that $\sqsubset^* \setminus id_X \subseteq \gamma \setminus id_X \subseteq \sqsubset_{spo}$ and $\frown_{spo} \cap \rightleftharpoons = \varnothing$, we have that $x \prec_{spo} y$. Moreover, again by $\sqsubset^* \setminus id_X \subseteq \gamma \setminus id_X \subseteq \sqsubset_{spo}$, we have $a = x \vee a \sqsubset_{spo} x$ and $y = b \vee y \sqsubset_{spo} b$. Hence, since $spo$ is a stratified poset, $a \prec_{spo} b$. □

The above result is similar to the following general characterisation of stratified poset extensions of GSO-structures.

**Proposition 3.** *If $gsos = (X, \rightleftharpoons, \sqsubset)$ is a* GSO*-structure then*

$$\mathsf{ext}(gsos) = \{spo \in \mathsf{ext}(\mathsf{sos}(gsos)) \mid \frown_{spo} \cap \rightleftharpoons = \varnothing\}\ .$$

*Proof.* $(\subseteq)$ Let $spo \in \mathsf{ext}(gsos)$. Then $\rightleftharpoons\ \subseteq \prec^{\mathsf{sym}}_{spo}$ and $\sqsubset\ \subseteq\ \precsim_{spo}$. Hence:

$$\rightleftharpoons \cap \sqsubset\ \subseteq\ \prec^{\mathsf{sym}}_{spo} \cap \precsim_{spo} = \prec_{spo}\ ,$$

yielding $spo \in \mathsf{ext}(\mathsf{sos}(gsos))$. Moreover, $\frown_{spo} \cap \rightleftharpoons\ = \varnothing$, by $\rightleftharpoons\ \subseteq \prec^{\mathsf{sym}}_{spo}$.

$(\supseteq)$ Let $spo \in \mathsf{ext}(\mathsf{sos}(gsos))$ and $\frown_{spo} \cap \rightleftharpoons\ = \varnothing$. The latter and irreflexivity of $\rightleftharpoons$ implies $\rightleftharpoons\ \subseteq \prec^{\mathsf{sym}}_{spo}$. Moreover, $\sqsubset\ \subseteq\ \precsim_{spo}$, by $spo \in \mathsf{ext}(\mathsf{sos}(gsos))$. Hence $spo \in \mathsf{ext}(gsos)$. □

## 3 Fitting nets and order structures

The operational and causality semantics of a class of Petri nets $\mathbb{PN}$ can be related within a common scheme introduced in [14]. It is reproduced here as Figure 2 where $N$ is a net from $\mathbb{PN}$ and:

- $\mathbb{EX}$ are executions (or observations) of nets in $\mathbb{PN}$.
- $\mathbb{LAN}$ are labelled acyclic nets, each representing a history.
- $\mathbb{LEX}$ are labelled executions of nets in $\mathbb{LAN}$.
- $\mathbb{LCS}$ are labelled causal structures (order structures) capturing the abstract causal relationships between executed actions.

In this paper, the executions in $\mathbb{EX}$ step sequences, and the labelled executions in $\mathbb{LEX}$ are labelled singular step sequences.

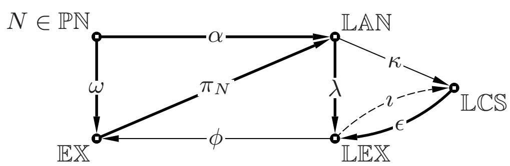


**Fig. 2.** Semantical framework for a class of Petri nets $\mathbb{PN}$. The bold arcs indicate mappings to powersets and the dashed arc indicates a partial function.

The maps in Figure 2 relate the semantical views in $\mathbb{EX}$, $\mathbb{LAN}$, $\mathbb{LEX}$, and $\mathbb{LCS}$:

- $\omega$ returns a set of executions, defining the *operational* semantics of $N$.
- $\alpha$ returns a set of labelled acyclic nets, defining an *axiomatic process* semantics of $N$.
- $\pi_N$ returns, for each execution of $N$, a non-empty set of labelled acyclic nets, defining an *operational process* semantics of $N$.
- $\lambda$ returns a set of *labelled* executions for each process of $N$, and after applying $\phi$ to such labelled executions one should obtain executions of $N$.
- $\kappa$ associates a labelled *causal* structure with each process of $N$.
- $\epsilon$ and $\imath$ allow one to go back and forth between labelled causal structures and sets of labelled executions associated with them.

The semantical framework captured by the above schema indicates how the different semantical views should agree. According to the rectangle on the left, the operational semantics of the Petri net defines processes satisfying certain axioms and moreover all labelled acyclic nets satisfying these axioms can be derived from the executions of the Petri net. Also, the labelled executions of the processes correspond with the executions of the original Petri net. The triangle on the right relates the labelled acyclic nets from $\mathbb{LAN}$ with the causal structures from $\mathbb{LCS}$ and the labelled executions from $\mathbb{LEX}$. The order structure defined by a labelled acyclic net can be obtained by combining executions of that net and, conversely, the stratified extensions of the order structure defined by a labelled acyclic net are its (labeled) executions. Thus the abstract relations between the actions in the labelled causal structures associated with the Petri net will be consistent with its chosen operational semantics.

To demonstrate that these different semantical views agree as captured through this semantical framework, it is sufficient to establish a series of results called *aims*. As there exist four simple requirements (called *properties*) guaranteeing these aims, one can concentrate on defining the semantical domains and maps appearing in Figure 2 and proving these properties.

**Property 1 (soundness of mappings)** *The maps* $\omega$, $\alpha$, $\lambda$, $\phi$, $\pi_N|_{\omega(N)}$, $\kappa$, $\epsilon$ *and* $\imath|_{\lambda(\mathbb{LAN})}$ *are total. Moreover,* $\omega$, $\alpha$, $\lambda$, $\pi_N|_{\omega(N)}$ *and* $\epsilon$ *always return non-empty sets.* ◇

**Property 2 (consistency)** *For all* $\xi \in \mathbb{EX}$ *and* $LN \in \mathbb{LAN}$,

$$\left.\begin{array}{r}\xi \in \omega(N)\\ LN \in \pi_N(\xi)\end{array}\right\} \quad \text{iff} \quad \left\{\begin{array}{l}LN \in \alpha(N)\\ \xi \in \phi(\lambda(LN))\ .\end{array}\right.$$

◇

**Property 3 (representation)** $\imath \circ \epsilon = id_{\mathbb{LCS}}$. ◇

**Property 4 (fitting)** $\lambda = \epsilon \circ \kappa$. ◇

The above four properties imply that the axiomatic (defined through $\alpha$) and operational (defined through $\pi_N \circ \omega$) process semantics of nets in $\mathbb{PN}$ are in full agreement. Also, the operational semantics of $N$ (defined through $\omega$) coincides with the operational semantics of the processes of $N$ (defined through $\phi \circ \lambda \circ \alpha$). Moreover, the causality in a process of $N$ (defined through $\kappa$) coincides with the causality structure implied by its operational semantics (through $\imath \circ \lambda$). That is, we have the following.

**Aim 1** $\alpha = \pi_N \circ \omega$. ◇

**Aim 2** $\omega = \phi \circ \lambda \circ \alpha$. ◇

**Aim 3** $\kappa = \imath \circ \lambda$. ◇

Thus, the operational semantics of the Petri net $N$ and the set of labelled causal structures associated with it are related by $\omega = \phi \circ \epsilon \circ \kappa \circ \alpha$.

**EN-systems with inhibitor arcs**

Usually, the fundamental net class for which processes and causality are introduced are EN-systems [18]. Here, however, we take elementary net systems with inhibitor arcs (ENI-systems) and use them to show how the semantical framework can be instantiated.

An ENI-system is a tuple $ENI \stackrel{\text{df}}{=} (P, T, F, Inh, M_{init})$ with $P$ and $T$ finite and disjoint sets of *places* — drawn as circles — and *transitions* — drawn as rectangles —, respectively; $F \subseteq (P \times T) \cup (T \times P)$ the flow relation of $ENI$ — the directed arcs in the diagrams; $Inh \subseteq P \times T$ its set of *inhibitor* arcs — with small circles as arrowheads; and $M_{init} \subseteq P$ its initial marking. (In general, any subset of places is a *marking*, in diagrams indicated by small black dots.) If $ENI$ has no inhibitor arcs, $Inh = \varnothing$, then it is an EN-system.

As usual, for every transition or place $x$ we define its inputs ${}^\bullet x \stackrel{\text{df}}{=} \{y \mid (y, x) \in F\}$ and outputs $x^\bullet \stackrel{\text{df}}{=} \{y \mid (x, y) \in F\}$. Moreover, ${}^\circ t \stackrel{\text{df}}{=} \{p \mid (p, t) \in Inh\}$ are the inhibitor places of transition $t$. We also define for any subset $U$ of $T$:

$$^\bullet U \stackrel{\text{df}}{=} \bigcup_{t \in U} {}^\bullet t \text{ and } U^\bullet \stackrel{\text{df}}{=} \bigcup_{t \in U} t^\bullet \text{ and } {}^\circ U \stackrel{\text{df}}{=} \bigcup_{t \in U} {}^\circ t \,.$$

A *step of* $ENI$ is a non-empty set $U$ of transitions such that $({}^\bullet t \cup t^\bullet) \cap ({}^\bullet u \cup u^\bullet) = \varnothing$, for all distinct $t, u \in U$. A step $U$ of $ENI$ is *enabled* at a marking $M$ of $ENI$ if ${}^\bullet U \subseteq M$ and $(U^\bullet \cup {}^\circ U) \cap M = \varnothing$. Such a step can then be *executed* leading to the marking $M' \stackrel{\text{df}}{=} (M \setminus {}^\bullet U) \cup U^\bullet$. We denote this by $M[U\rangle_{ENI} M'$ or by $M[U\rangle M'$ if $ENI$ is clear.

Thus the operational semantics of $ENI$ is defined: $\omega(ENI)$ comprises all step sequences $\xi = U_1 \dots U_k$ ($k \geq 0$) such that there are markings $M_{init} = M_0, \dots, M_k$ with $M_{i-1}[U_i\rangle M_i$, for $i = 0, \dots, k-1$. We call $M_k$ a *reachable* marking of $ENI$.

In what follows we will assume that each inhibitor place $p$ of an ENI-system $ENI$ has a *complement place* $\widetilde{p}$ such that ${}^\bullet p = \widetilde{p}^\bullet$ and ${}^\bullet \widetilde{p} = p^\bullet$; moreover $|\{p, \widetilde{p}\} \cap M_{init}| = 1$. It is immediate that $|\{p, \widetilde{p}\} \cap M| = 1$, for all reachable markings $M$ and all places $p$. Note that complement places can always be added to $ENI$ as this does not affect its operational semantics.

Thus, for ENI-systems $\mathbb{EX}$ are step sequences. In addition, the labelled causal structures $\mathbb{LCS}$ are SO-structures, and the labelled executions $\mathbb{LEX}$ will be labelled singular step sequences. Next we introduce the labelled acyclic nets that will form the semantical domain $\mathbb{LAN}$ for the process semantics of ENI-systems. These nets will have activator rather than inhibitor arcs.

**Definition 3 (activator occurrence nets).** *An* activator occurrence net *(or* AO*-net) is a tuple* $AON \stackrel{\text{df}}{=} (P', T', F', Act, \ell)$ *such that:*

- $P'$, $T'$ *and* $F'$ *are places, transitions and flow relation as in* ENI*-systems.*
- $|{}^\bullet p| \leq 1$ *and* $|p^\bullet| \leq 1$*, for every place* $p$*.*
- $Act \subseteq P' \times T'$ *is a set of activator arcs (indicated by black dot arrowheads) and* $\ell$ *is a labelling for* $P' \cup T'$*.*
- *The relational structure* $\rho_{AON} \stackrel{\text{df}}{=} (T', \prec_{loc}, \sqsubset_{loc})$ *is so-acyclic, where* $\prec_{loc}$ *and* $\sqsubset_{loc}$ *are respectively given by* $(F' \circ F')|_{T' \times T'} \cup (F' \circ Act)$ *and* $Act^{-1} \circ F'$*, as illustrated in Figure 3.* ◇

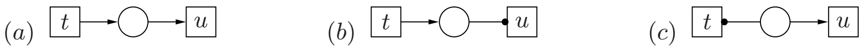


**Fig. 3.** Two cases $(a)$ and $(b)$ defining $t \prec_{loc} u$, and one case $(c)$ defining $t \sqsubset_{loc} u$.

We use ${}^\blacklozenge t \overset{\mathrm{df}}{=} \{p \mid (p,t) \in Act\}$ to denote the activator places of a transition $t$, and ${}^\blacklozenge U \overset{\mathrm{df}}{=} \bigcup_{t \in U} {}^\blacklozenge t$ for the activator places of a set $U \subseteq T'$. As for ENI-systems, a *step of* $AON$ is a non-empty set $U$ of transitions such that $({}^\bullet t \cup t^\bullet) \cap ({}^\bullet u \cup u^\bullet) = \varnothing$, for all distinct $t, u \in U$. A step $U$ of $AON$ is *enabled* at a marking $M$ of $AON$ if ${}^\bullet U \cup {}^\blacklozenge U \subseteq M$. The execution of such a $U$ is defined as for ENI-systems and leads to the marking $(M \setminus {}^\bullet U) \cup U^\bullet$.

The default *initial* and *final* markings of $AON$ are $M^{AON}_{init}$ and $M^{AON}_{fin}$ consisting respectively of all places $p$ without inputs (${}^\bullet p = \varnothing$) and all places $p$ without outputs ($p^\bullet = \varnothing$). The behaviour of $AON$ is captured by the set $\lambda(AON)$ of all step sequences from $M^{AON}_{init}$ to $M^{AON}_{fin}$. The set $\mathsf{reach}(AON)$ of markings *reachable* in $AON$ comprises all markings $M$ reachable from $M^{AON}_{init}$ such that $M^{AON}_{fin}$ is reachable from $M$. One can show that each step sequence $\sigma \in \lambda(AON)$ is singular, and that its set of elements is exactly the set of transitions $T'$. For such a step sequence $\sigma$, $\phi(\sigma)$ is obtained from $\sigma$ by replacing each $t$ by $\ell(t)$.

We define $\kappa(AON) \overset{\mathrm{df}}{=} \rho^{\mathsf{so}}_{AON}$ which is guaranteed to be an SO-structure by the so-acyclicity of $\rho_{AON}$ [10].

As far as the mappings $\epsilon$ and $\iota$ are concerned, $\epsilon$ is the set of stratified poset extensions (or, equivalently, singular step sequences) of an SO-structure, and $\iota$ is the intersection of the SO-structures (or, equivalently, singular step sequences) corresponding to a set of stratified posets with the same domain. Thus Theorem 1 immediately yields Property 3.

Finally, we give the axiomatic and operational process semantics of an ENI-system $ENI = (P, T, F, Inh, M_{init})$.

**Definition 4 (processes of ENI-systems).** *A* process *of $ENI$ is an* AO*-net $AON$ such that its labelling $\ell$:*

- *labels the places of $AON$ with places of $ENI$.*
- *labels the transitions of $AON$ with transitions of $ENI$.*
- *is injective on $M^{AON}_{init}$ and $\ell(M^{AON}_{init}) = M_{init}$.*
- *is injective on ${}^\bullet t$ and $t^\bullet$ and, moreover, $\ell({}^\bullet t) = {}^\bullet \ell(t)$ and $\ell(t^\bullet) = \ell(t)^\bullet$, for every transition $t$ of $AON$.*
- *$\ell$ is injective on ${}^\blacklozenge t$ and $\ell({}^\blacklozenge e) = \widetilde{{}^\circ \ell(t)}$ for every transition $t$ of $AON$.*

*We denote this by $AON \in \alpha(ENI)$.* ◇

**Definition 5 (processes construction).** *An* AO*-net* generated *by a step sequence $\sigma = U_1 \dots U_n \in \omega(ENI)$ is the last element in the sequence $AON_0, \dots, AON_n$ where each $AON_k \overset{\mathrm{df}}{=} (P_k, T_k, F_k, A_k, \ell_k)$ is an* AO*-net such that:*
*Step 0: $P_0 \overset{\mathrm{df}}{=} \{p^1 \mid p \in M_{init}\}$ and $T_0 = F_0 = A_0 \overset{\mathrm{df}}{=} \varnothing$.*

*Step $k$: Given $AON_{k-1}$ the sets of nodes and arcs are extended as follows:*

$$\begin{aligned}
P_k &\stackrel{\text{df}}{=} P_{k-1} \cup \{p^{1+\triangle p} \mid p \in U_k^\bullet\} \\
T_k &\stackrel{\text{df}}{=} T_{k-1} \cup \{t^{1+\triangle t} \mid t \in U_k\} \\
F_k &\stackrel{\text{df}}{=} F_{k-1} \cup \{(p^{\triangle p}, t^{1+\triangle t}) \mid t \in U_k \wedge p \in {}^\bullet t\} \\
&\qquad\quad \cup \{(t^{1+\triangle t}, p^{1+\triangle p}) \mid t \in U_k \wedge p \in t^\bullet\} \\
A_k &\stackrel{\text{df}}{=} A_{k-1} \cup \{(\widetilde{p}^{\,\triangle \widetilde{p}}, t^{1+\triangle t}) \mid t \in U \wedge p \in {}^\circ t\}\ .
\end{aligned}$$

*In the above, the label of each node $\ell_k(x^i)$ is set to be $x$, and $\triangle x$ denotes the number of the nodes of $AON_{k-1}$ labelled by $x$. We denote this by $AON_n \in \pi_{ENI}(\sigma)$.* ◇

Note that $\pi_{ENI}(\sigma)$ comprises exactly one net (up to isomorphism). The same holds for $\pi_{ENIM}(\sigma)$ defined later.

As one can show that the remaining properties are also satisfied, the semantical framework for ENI-systems holds [14].

## 4 Mutually exclusive transitions

We now introduce a new class of Petri nets by extending ENI-systems with mutex arcs prohibiting certain pairs of transitions from occurring simultaneously (i.e., in the same step). Consider Figure 4 which shows a variant of the producer/consumer scheme. In this case, the producer is allowed to retire (transition $r$), but never at the same time as the consumer finishes the job (transition $f$). Other than that, there are no restrictions on the executions of transitions $r$ and $f$. To model such a scenario we use a mutex arc between transitions $r$ and $f$ (depicted as an undirected edge). Note that mutex arcs are relating transitions in a direct way. This should however not be regarded as an unusual feature as, for example, Petri nets with priorities also impose direct relations between transitions.

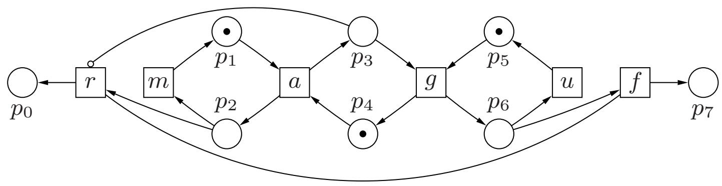


**Fig. 4.** An ENIM-system modelling a producer/consumer system with the actions: 'make item' $m$, 'add item to buffer' $a$, 'get item from buffer' $g$, 'use item' $u$, 'producer retires' $r$, and 'consumer finishes' $f$. Note: the producer can only retire if the buffer is empty (i.e., $p_3$ is empty).

An *elementary net system with inhibitor and mutex arcs* (or ENIM-system) is a tuple $ENIM \stackrel{\text{df}}{=} (P, T, F, Inh, Mtx, M_{init})$ such that $\mathsf{und}(ENIM) \stackrel{\text{df}}{=} (P, T, F, Inh, M_{init})$ is the ENI-system *underlying* $ENIM$ and $Mtx \subseteq T \times T$ is a symmetric irreflexive relation specifying the *mutex* arcs of $ENIM$. Where possible, we retain the definitions introduced for ENI-systems. The notion of a step now changes however. A *step of* $ENIM$ is a non-empty set $U$ of transitions such that $U$ is a step of $\mathsf{und}(ENIM)$ and in addition $Mtx \cap (U \times U) = \varnothing$. With this modified notion of a step, the remaining definitions pertaining to the dynamic aspects of an ENIM-system, including $\omega(ENIM)$, are the same as for the underlying ENI-system $\mathsf{und}(ENIM)$.

**Proposition 4.** $\omega(ENIM) = \{U_1 \dots U_k \in \omega(\mathsf{und}(ENIM)) \mid Mtx \cap \bigcup_i U_i \times U_i = \varnothing\}$.

*Proof.* Follows from the definitions. □

For the ENIM-system of Figure 4, we have that $M[\{r\}\rangle M''[\{f\}\rangle M'$ as well as $M[\{f\}\rangle M'''[\{r\}\rangle M'$, where $M = \{p_2, p_4, p_6\}$ and $M' = \{p_0, p_4, p_7\}$. However, $M[\{r, f\}\rangle M'$ which holds for the underlying ENI-system does not hold now as $r$ and $f$ cannot be executed in the same step.

To deal with the behaviours of ENIM-systems in the context of the semantical framework, we adapt the approach followed for ENI-system as recalled above. The labelled causal structures, $\mathbb{LCS}$, are now GSO-structures, while labelled executions, $\mathbb{LEX}$, are labelled singular step sequences, as before. The labelled acyclic nets, $\mathbb{LAN}$, used for the process semantics of ENIM-systems are introduced next.

**Definition 6 (activator mutex occurrence nets).** *An* activator mutex occurrence net *(or* AMO*-net) is a tuple* $AMON \stackrel{\text{df}}{=} (P', T', F', Act, Mtx', \ell)$ *such that:*

- $\mathsf{und}(AMON) \stackrel{\text{df}}{=} (P', T', F', Act, \ell)$ *is the* AO*-net* underlying $AMON$ *and* $Mtx' \subseteq T' \times T'$ *is a symmetric irreflexive relation specifying the* mutex *arcs of* $AMON$.
- $\rho_{AMON} \stackrel{\text{df}}{=} (T', \prec_{loc}, \sqsubset_{loc}, Mtx')$*, where* $\prec_{loc}$ *and* $\sqsubset_{loc}$ *are defined as for* AO*-nets in Definition 3, is a gso-acyclic relational structure.* ◇

The part of gso-acyclicity $\rho_{AMON}$ which deals with the mutex arcs is illustrated in Figure 5. We have there three transitions satisfying $a \sqsubset_{loc} b \sqsubset_{loc} c \sqsubset_{loc} a$. Hence, in any execution involving all these transitions, they have to belong to the same step. This, however, is inconsistent with a mutex arc between $b$ and $c$, and the gso-acyclicity fails to hold because $(a, a)$ belongs to $\sqsubset_{loc}^* \circ (Mtx' \cap \sqsubset_{loc}^*) \circ \sqsubset_{loc}^*$.

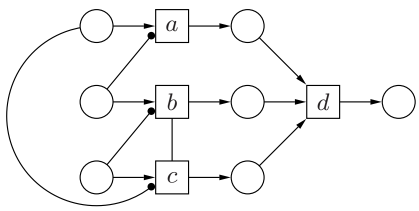


**Fig. 5.** A net which is not an AMO-net as it fails the gso-acyclicity test.

Then we let $\kappa(AMON) \stackrel{\text{df}}{=} \rho_{AMON}^{\mathsf{gso}}$ be the GSO-structure generated by $AMON$. Note that Proposition 1 guarantees the correctness of this definition. Moreover, it is consistent with the SO-structure defined by its underlying AO-net.

**Proposition 5.** $(T', \prec_{loc}, \sqsubset_{loc})$ *is an so-acyclic relational structure.*

*Proof.* Follows from Proposition 2. □

As far as the mappings $\epsilon$ and $\iota$ are concerned, $\epsilon$ is the set of stratified poset (or, equivalently, singular step sequences) extensions of a GSO-structure, and $\iota$ is the intersection of the GSO-structures corresponding to a set of stratified posets with the same domain. Thus Theorem 2 immediately yields Property 3. Other properties are dealt with later in this section.

The default initial and final markings of $AMON$, as well as its step sequence executions are defined exactly the same as for the underlying AO-net under the proviso that steps do not contain transitions joined by mutex arcs.

The following results yield more insight into the labelled executions of an activator mutex occurrence net relative to its underlying AO-net.
Let $AMON = (P', T', F', Act, Mtx', \ell)$ be an AMO-net and $AON = \mathsf{und}(AMON)$.

**Proposition 6.** $\lambda(AMON) = \{U_1 \ldots U_k \in \lambda(AON) \mid Mtx' \cap \bigcup_i U_i \times U_i = \varnothing\}$.

*Proof.* Follows from the definitions. □

**Proposition 7.** *Let $\sigma = U_1 \ldots U_k \in \lambda(AON)$ be such that there is no $i \leq k$ for which there exists a partition $U, U'$ of $U_i$ such that $U_1 \ldots U_{i-1}UU'U_{i+1} \ldots U_k \in \lambda(AON)$. Then $\sigma \in \lambda(AMON)$.*

*Proof.* By Proposition 6, it suffices to show that, for every $i \leq k$, $(U_i \times U_i) \cap Mtx' = \varnothing$. Suppose this does not hold for some $i \leq k$. Let $\kappa(AON) = (T', \prec, \sqsubset)$. From the assumption made about $\sigma$ it follows that $t \sqsubset u$, for all distinct $t, u \in U_i$. This, however, contradicts the gso-acyclicity of $\rho_{AMON}$. □

**Proposition 8.** $\mathsf{reach}(AMON) = \mathsf{reach}(AON)$.

*Proof.* $(\subseteq)$ Follows from Proposition 6.
$(\supseteq)$ Follows from Proposition 7 and the fact that each step sequence in $\lambda(AON)$ can be 'sequentialised' into the form from the formulation of Proposition 7 by splitting the steps into smaller ones. □

**Proposition 9.** *A marking $M$ belongs to* $\mathsf{reach}(AMON)$ *iff there are no places $p, p' \in M$ for which $(p, p') \in F' \circ (\prec_{loc} \cup \sqsubset_{loc})^* \circ F'$.*

*Proof.* Follows from Proposition 8 and Proposition 5.15 in [14]. □

Figure 6 depicts an AMO-net labelled with places and transitions of the ENIM-system of Figure 4. We have that both $\{a\}\{g\}\{r\}\{f\}$ and $\{a\}\{g\}\{f\}\{r\}$ belong to $\phi(\lambda(AMON_0))$, however, $\{a\}\{g\}\{f, r\}$ does not.

Now we are ready to introduce process semantics for ENIM-systems.

**Definition 7 (processes of ENIM-systems).** *A* process *of $ENIM$ is an* AMO*-net $AMON$ such that* $\mathsf{und}(AMON)$ *is a process of* $\mathsf{und}(ENIM)$ *and, for all $t, u \in T'$, we have* $(t, u) \in Mtx'$ iff $(\ell(t), \ell(u)) \in Mtx$. *We denote this by $AMON \in \alpha(ENIM)$.* ⋄

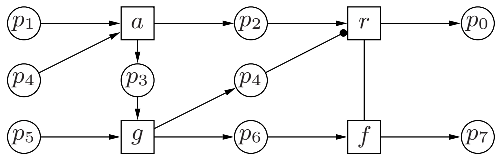


**Fig. 6.** An AMO-net $AMON_0$ with labels shown inside places and transitions.

**Definition 8 (processes construction).** *An* AMO*-net* generated *by a step sequence* $\sigma = U_1 \ldots U_n \in \omega(ENIM)$ *is the last net in the sequence* $AMON_0, \ldots, AMON_n$ *where each* $AMON_k \stackrel{\text{df}}{=} (P_k, T_k, F_k, A_k, M_k, \ell_k)$ *is as in Definition 5 except that* $M_k \stackrel{\text{df}}{=} \{(e,f) \in T_k \times T_k \mid (\ell_k(e), \ell_k(f)) \in Mtx\}$ *is an added component. We denote this by* $AMON_n \in \pi_{ENIM}(\sigma)$ ◇

The way in which mutex arcs are added in the process construction entails that some of them may be redundant when, for example, the transitions they join are causally related. However, eliminating such redundant mutex arcs (which is possible by analysing paths in the AMO-net) would go against the locality principle which is the basis of the process approach. Indeed, this approach does not remove redundant causalities as this would compromise the local causes and effects in the definition and construction of process nets.

The AMON-net shown in Figure 6 is a process of the ENIM-system of Figure 4 with $\phi(\lambda(AMON_0)) = \big\{\{a\}\{g\}\{f\}\{r\}, \{a\}\{g\}\{r\}\{f\}\big\}$. Figure 7 shows the result of applying the construction from Definition 8 to the ENIM-system of Figure 4 and one of its step sequences. Note that the resulting AMO-net is isomorphic to that shown in Figure 6.

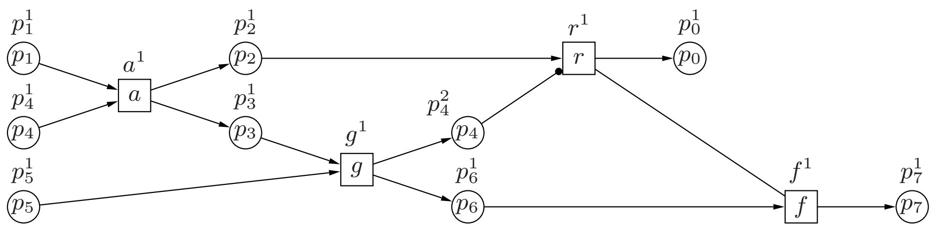


**Fig. 7.** Process generated for the ENIM-system in Figure 4 and $\sigma \stackrel{\text{df}}{=} \{a\}\{g\}\{r\}\{f\}$.

Having instantiated the semantical framework for ENIM-systems, we can now formally establish their connection with GSO-structures by proving the remaining Properties 1, 2, and 4. Below we assume that $ENIM$ is an ENIM-system.

**Proposition 10.** *Let* $\sigma$ *a step sequence of* $ENIM$, $AMON$ *an* AMO*-net,* $gsos$ *a* GSO*-structure, and* $\mathcal{SPO}$ *a set of stratified posets with the same domain.*

1. *$\omega(ENIM)$, $\alpha(ENIM)$, $\lambda(AMON)$ and $\epsilon(gsos)$ are non-empty sets.*
2. *$\kappa(AON)$ and $\iota(\mathcal{SPO})$ are* GSO*-structures.*
3. *$\pi_{ENIM}(\sigma)$ comprises an* AMO*-net.*

*Proof.* In what follows, we use the notations introduced throughout this section.

(1) We have $\omega(ENIM) \neq \varnothing$ as the empty string is a valid step sequence of $ENIM$. To show $\alpha(ENIM) \neq \varnothing$ one can take the AMO-net consisting of the initial marking of $ENIM$ with the identity labelling and no transitions. That $\epsilon(gsos) \neq \varnothing$ follows from Theorem 2. That $\lambda(AMON) \neq \varnothing$ follows from Proposition 7, $\lambda(AON) \neq \varnothing$ and the fact that each step sequence in $\lambda(AON)$ can be 'sequentialised' into the form from the formulation of Proposition 7 by splitting the steps into smaller ones.

(2) Follows from Theorem 2 and Proposition 1.

(3) We have that an element of $\pi_{ENIM}(\sigma)$ with deleted mutex arcs is an AO-net. It therefore suffices to show that the relation $\sqsubset_{loc}^{*} \circ (Mtx' \cap \sqsubset_{loc}^{*}) \circ \sqsubset_{loc}^{*}$ is irreflexive.

Suppose that $(t,t) \in \sqsubset_{loc}^{*} \circ (Mtx' \cap \sqsubset_{loc}^{*}) \circ \sqsubset_{loc}^{*}$. Then there are $t = t_1, \ldots, t_k = t$ such that $(t_i, t_{i+1}) \in \sqsubset_{loc}$ for all $i < k$, and $(t_m, t_j) \in M_n$ for some $m < j \leq k$. But this means that $t_1, \ldots, t_k$ have been generated in the same step of the construction, contradicting the definition of executability in ENIM-systems. □

**Proposition 11.** *Let $\xi \in \omega(ENIM)$ and $AMON \in \pi_{ENIM}(\xi)$.*

1. *$AMON \in \alpha(ENIM)$.*
2. *$\xi \in \phi(\lambda(AMON))$.*

*Proof.* (1) By Proposition 10(3), $AMON$ is an AMO-net. Moreover, by [14], we have that $\mathsf{und}(AMON) \in \alpha(\mathsf{und}(ENIM))$. Finally, the condition involving mutex arcs follows from the construction in Definition 8.

(2) By [14], $\xi \in \phi(\lambda(\mathsf{und}(AMON)))$. Hence $\xi = \phi(\sigma)$ for some $\sigma = U_1 \ldots U_k \in \lambda(\mathsf{und}(AMON))$. The latter, together with $\xi \in \omega(ENIM)$ and the consistency between mutex arcs in $ENIM$ and $AMON$, means that there is no mutex arc joining two elements of any $U_i$. Hence, by Proposition 6, $\sigma \in \lambda(AMON)$. Thus $\xi \in \phi(\lambda(AMON))$. □

**Proposition 12.** *Let $AMON \in \alpha(ENIM)$ and $\xi \in \phi(\lambda(AMON))$.*

1. *$\xi \in \omega(ENIM)$.*
2. *$AMON \in \pi_{ENIM}(\xi)$.*

*Proof.* (1) By [14], $\xi \in \omega(\mathsf{und}(ENIM))$. Also there is $\sigma = U_1 \ldots U_k \in \lambda(AMON)$ such that $\xi = \phi(\sigma)$. The latter, together with the consistency between mutex arcs in $ENIM$ and $AMON$, means that there is no mutex arc joining two elements of any $U_i$. Hence, by Proposition 4, $\xi \in \omega(ENIM)$.

(2) By [14], $\mathsf{und}(AMON) \in \pi_{\mathsf{und}(ENIM)}(\xi)$. Moreover, the mutex arcs are added in the same (deterministic) way to the underlying process nets, leading to $AMON \in \pi_{ENIM}(\xi)$. □

Hence Property 2 holds. We then observe that Property 3 is simply Theorem 2, and Property 4 is proved below.

**Proposition 13.** *Let* $AMON$ *be an* AMO*-net. Then* $\lambda(AMON) = \epsilon(\kappa(AMON))$.

*Proof.* We have:

$$\begin{array}{l}\epsilon(\kappa(AMON)) = \mathsf{ext}(\rho^{\mathsf{gso}}_{AMON}) = \mathsf{ext}((T', \prec_{loc}, \sqsubset_{loc}, Mtx')^{\mathsf{gso}}) =_{\text{(Prop. 2)}} \\ \{spo \in \mathsf{ext}((T', \prec_{loc}, \sqsubset_{loc})^{\mathsf{so}}) \mid \frown_{spo} \cap Mtx' = \varnothing\} = \\ \{spo \in \epsilon(\kappa(AON)) \mid \frown_{spo} \cap Mtx' = \varnothing\} = \\ \{spo \in \lambda(AON) \mid \frown_{spo} \cap Mtx' = \varnothing\} =_{\text{(Prop. 6)}} \lambda(AMON).\end{array}$$

Note that we identify stratified posets with their corresponding singular labelled step sequences. □

Finally, we can claim the semantical aims for ENIM-systems.

**Theorem 3.** *Let* $ENIM$ *be an* ENIM*-system, and* $AMON$ *be an* AMO*-net.*

$$\begin{array}{rcl}\alpha(ENIM) & = & \pi_{ENIM}(\omega(ENIM)) \\ \omega(ENI) & = & \phi(\lambda(\alpha(ENIM))) \\ \kappa(AMON) & = & \iota(\lambda(AMON))\ .\end{array}$$

## 5 Concluding remarks

We already mentioned that trying to avoid redundant mutex arcs when constructing processes would require investigation of various paths in the constructed AMO-net. In particular, it would not be sufficient to only consider the most recent transition occurrences. Consider, for example, the ENIM-system shown in Figure 8 and its step sequence $\sigma \stackrel{\text{df}}{=} \{b\}\{c\}\{b\}\{a\}$. The corresponding process, also shown in Figure 8, has two mutex arcs adjacent to the transition $a^1$. We then observe that dropping the joining of $a^1$ with $b^1$ would not be right, as the resulting AMON-net would generate a step sequence $\{a^1, b^1\}\{c^1\}\{b^2\}$, or $\{a, b\}\{c\}\{b\}$ after applying labelling, which is not a valid step sequence of the ENIM-system.

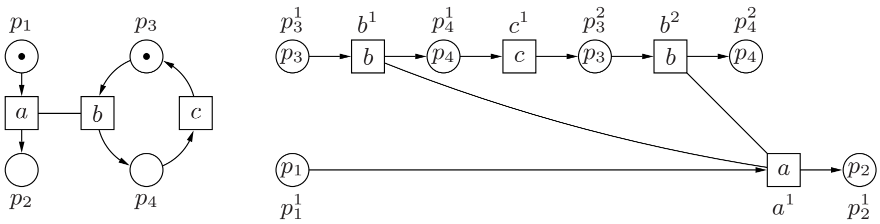


**Fig. 8.** Mutex arcs may need to connect all potential mutex transitions.

Modelling mutually exclusive transitions can be done in PT-nets using self-loops linking mutually exclusive transitions to a place marked with a single token (which has no other arcs attached to it). This is illustrated in Figure 9($a$). An alternative would be to use a mutex arc, as shown in Figure 9($b$). At a purely modelling level, there is no real difference between these two representations. However, at the semantical level, the differences can be significant. The point is that mutex arcs represent concurrent histories in a compact way. This should have a direct impact on the size of net unfolding used, in particular, for model checking. For example, the single process in Figure 9($c$) derived for the representation of Figure 9($b$) has to be replaced by two processes derived for the representation of Figure 9($a$) depicted in Figure 9($d$). It is important to observe that these two non-isomorphic processes cannot be equated using the so-called token swapping technique from [1], as the PT-net is 1-safe, suggesting that the potential state space reductions due to mutex arcs have not been considered so far. Intuitively, mutex arc stem from a different philosophy to self-loops. Whereas the latter are related to resource sharing, mutex arcs are derived from semantical considerations and so can provide a more convenient modelling tool.

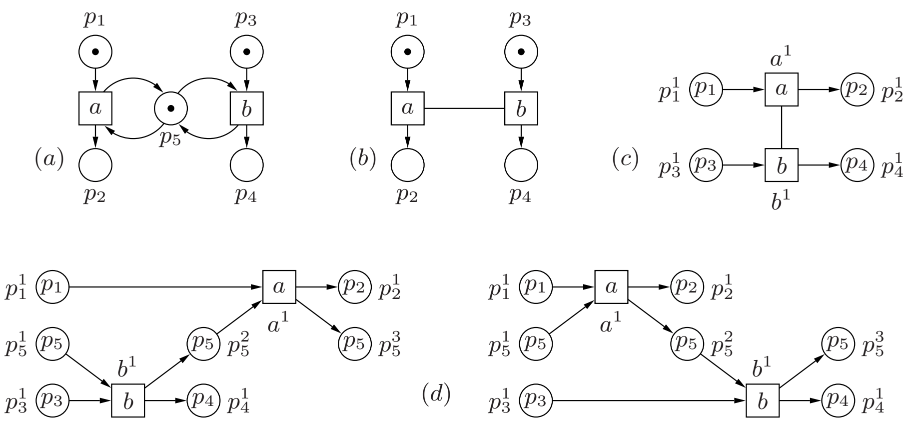


**Fig. 9.** Mutex arcs lead to more condensed process semantics than self-loops.

In our future work we plan to investigate the relationship between mutex arcs and other modelling concepts such as localities [15] and policies [3], also from the point of view of the synthesis of nets where unorderedness does not imply simultaneity of executed actions.

In this paper we did not consider ENM-systems, i.e., EN-systems extended with mutex arcs, as it was our intention to investigate a system model corresponding to the most general paradigm $\pi_1$. In future, we intend to find out where ENM-systems fit into the approach presented here.

## References

1. Best, E., Devillers, R.: Sequential and Concurrent Behaviour in Petri Net Theory. Theoretical Computer Science **55** (1987) 87–136

2. Billington, J.: Protocol Specification Using P-Graphs, a Technique Based on Coloured Petri Nets. In: Part II of [17] (1998) 331–385
3. Darondeau, P., Koutny, M., Pietkiewicz-Koutny M., Yakovlev, A.: Synthesis of Nets with Step Firing Policies. Fundamenta Informaticae **94** (2009) 275–303
4. Donatelli, S., Franceschinis, G.: Modelling and Analysis of Distributed Software Using GSPNs. In: Part II of [17], (1998) 438–476
5. Esparza, J., Bruns, G.: Trapping Mutual Exclusion in the Box Calculus. Theoretical Computer Science **153** (1996) 95–128
6. Gaifman, H., Pratt, V.R.: Partial Order Models of Concurrency and the Computation of Functions. In: LICS, IEEE Computer Society (1987) 72–85
7. Guo, G., Janicki, R.: Modelling Concurrent Behaviours by Commutativity and Weak Causality Relations. Lecture Notes in Computer Science **2422** (2002) 178–191
8. Janicki, R.: Relational Structures Model of Concurrency. Acta Informatica **45** (2008) 279–320
9. Janicki, R., Koutny, M.: Structure of Concurrency. Theoretical Computer Science **112** (1993) 5–52
10. Janicki, R., Koutny, M.: Semantics of Inhibitor Nets. Information and Computation **123** (1995) 1–16
11. Janicki, R., Koutny, M.: Order Structures and Generalisations of Szpilrajn's Theorem. Acta Informatica **34** (1997) 367–388
12. Janicki, R., Koutny, M., Kleijn, J.: Quotient Monoids and Concurrent Behaviours. In: Scientific Applications of Language Methods, Carlos Martín-Vide (eds.). Mathematics, Computing, Language, and Life: Frontiers in Mathematical Linguistics and Language Theory **2**, World Scientific (2010)
13. Juhás, G., Lorenz, R., Mauser, S.: Causal Semantics of Algebraic Petri Nets distinguishing Concurrency and Synchronicity. Fundamenta Informaticae **86** (2008) 255–298
14. Kleijn, H.C.M., Koutny, M. Process Semantics of General Inhibitor Nets. Information and Computation **190** (2004) 18–69
15. Kleijn, J., Koutny, M., Rozenberg, G.: Petri Net Semantics for Membrane Systems. Journal of Automata, Languages, and Combinatorics **11** (2006) 321–340
16. Lengauer, C., Hehner, E.C.R.: A Methodology for Programming with Concurrency: An Informal Presentation. Science of Computer Programming **2** (1982) 1–18
17. Reisig, W., Rozenberg, G. (eds.): Lectures on Petri Nets. Lecture Notes in Computer Science **1491,1492** (1998)
18. Rozenberg, G., Engelfriet, J.: Elementary Net Systems. In: Part I of [17] (1998) 12–121
19. Szpilrajn, E.: Sur l'extension de l'ordre partiel. Fundamenta Mathematicae **16** (1930) 386–389