HMSCs as partial specifications ... with PNs as completions

Benoit Caillaud, Philippe Darondeau, Loïc Hélouët, and Gilles Lesventes

IRISA, campus de Beaulieu, F35042 Rennes Cedex
Name.Surname@irisa.fr

Abstract. The paper presents ongoing work aiming at understanding the nature of specifications given by High Level Message Sequence Charts and the ways in which they can be put into effective use. Contrarily to some authors, we do not set finite state restrictions on HMSCs as we feel such restrictions do not fit in with the type of distributed systems encountered today in the field of telecommunications. The talk presents first a series of undecidability results about general HMSCs following from corresponding undecidability results on rational sets in product monoids. These negative results which with one exception do not appear yet in the literature on HMSCs do indicate that the sole way in which general HMSCs may be usefully handed as behavioural specifications is to interpret their linear extensions as minimal languages, to be approximated from above in any realization. The problem is then to investigate frameworks in which these incomplete specifications may be given a meaning by a closure operation. The second part of the paper presents a closure operation relative to Petri net languages. This closure operation is an effective procedure that relies on semilinear properties of HMSCs languages. We finally present some decidability results for the distribution and verification of HMSCs transformed into Petri nets.

1 Introduction

Message Sequence Charts (MSC) are a modern form of the old timing diagrams, adapted so as to describe scenarios in which the agents of a distributed system communicate by end-to-end message passing. High Level MSCs (HMSC) are not MSCs but finite generators of MSCs used to describe in a compact way a whole family of scenarios in quite the same way as a finite automaton generates a set of words, but using a different concatenation since MSCs are partial words rather than words. Indeed a MSC is drawn in graphical form as the Hasse diagram of a partially ordered multiset (pomset, or partial word) whose linear extensions form a language (set of words). The concatenation of MSCs is therefore the concatenation of pomsets, and it induces a somewhat complex operation of composition on the associated languages. As a result, the language formed of all linear extensions of all MSCs produced from a finite generator (HMSC) is generally not finitely recognizable. Before discussing consequences, let us introduce first precise definitions. The following is not a literal reproduction of the definitions for MSCs and HMSCs produced by the normalizing committee of the ITU [16] but we choosed to keep compatibility of our simplified definitions with this standard. The reader should be warned that we disregard the branching semantics of HMSCs (for more on this, see Mauw and Reniers's paper [18], where HMSCs are given a process algebra semantics): we are concerned here exclusively with the linear behaviours of HMSCs. Owing to this option, we feel free to ignore co-regions as they may always be expanded by interleaving without affecting linear behaviours.

1.1 Notations

A MSC describes the joint behaviour of a fixed family of agents each of which executes a process defined by a fixed finite sequence of events. Events may be autonomous (e.g., message emissions or private events) or non autonomous (e.g., message receptions). We assume a finite family of agents $[n] = \{1, \ldots, n\}$, a finite set of private events P owned by agents, and a finite set M of messages each of which determines its sender and its receiver. This yields an alphabet of events E that decomposes into a partition $P \cup S \cup R$ as follows:

- P (for private) is the set of private events,
- S (for sending) is the set of message emissions ($S = \{!m | m \in M\}$),
- R (for receiving) is the set of message receptions ($R = \{?m | m \in M\}$).

Let $\phi : E \to [n]$ be the map such that:

- for $e \in P$, $\phi(e)$ is the owner of the private event e,
- for $e = !m$, $\phi(e)$ is the emitter of the message m,
- for $e = ?m$, $\phi(e)$ is the receiver of the message m.

The alphabet of events E may be partitioned accordingly into $E_1 \cup \cdots \cup E_n$, where $E_i = \{e \in E \mid \phi(e) = i\}$. Now for any word $w \in E^*$, let $\pi_i(w)$ denote the projection of w on E_i thus $\pi_i(w) \in E_i^*$, and let $\delta(w)$ denote the distribution of w on $E_1^* \times \cdots \times E_n^*$ thus $\delta(w) = (\pi_1(w), \ldots, \pi_n(w))$. The operation of distribution will play an important role in the sequel. To end up with generalities, let us recall that a word u is a prefix of w ($u \in pref(w)$) if $w = uv$ for some v, and that $|w|_e$ is the number of occurrences of the symbol e in w. We come now to the main definitions.

1.2 Basic Message Sequence Charts

Definition 1. *A word $w \in E^*$ is said to be* admissible *if $|u|_{!m} \geq |u|_{?m}$ for every $u \in pref(w)$ and for every message $m \in M$. A vector $W = (w_1, \ldots, w_n) \in E_1^* \times \cdots \times E_n^*$ is said to be* admissible *if $W = \delta(w)$ for some admissible word $w \in E^*$. A* scenario *is an admissible vector $W \in E_1^* \times \cdots \times E_n^*$, and it is a* closed *scenario if moreover $W = \delta(w)$ entails $|w|_{!m} = |w|_{?m}$ for every $m \in M$. A* basic Message Sequence Chart *(or bMSC) is a closed scenario.*

Remark 1. When M is the empty set of messages, every word $w \in E^*$ and every vector of words $(w_1, \ldots, w_n) \in E_1^* \times \cdots \times E_n^*$ are admissible.

Definition 1 calls for a few comments. In a scenario $W = (w_1, \ldots, w_n)$, each word $w_i \in E_i^*$ defines the process of the corresponding agent $i \in [n]$. The admissibility condition guarantees there is at least one way to interleave these processes in a joint process such that all receptions of messages are preceded by matching emissions. The additional condition on bMSCs guarantees that all messages sent are received later on in this joint process. However, this prevents us from representing communication via gates (e.g., environmental communications) in bMSCs and HMSCs. Before defining the latter, let us introduce a concatenation operation on scenarios.

Definition 2. *Given scenarios $U = (u_1, \ldots, u_n)$ and $V = (v_1, \ldots, v_n)$ let their* concatenation *be defined as $U \cdot V = (u_1.v_1, \ldots, u_n.v_n)$.*

It is easily seen that $U \cdot V = \delta(uv)$ if $U = \delta(u)$ and $V = \delta(v)$; it follows from this observation that the concatenation of two scenarios (resp. bMSCs) is a scenario (resp. a bMSC). Basic Message Sequence Charts form therefore a monoid, with the distribution of the empty word as the neutral element. This allows to define families of bMSCs using finite automata interpreted in this monoid.

1.3 High Level Message Sequence Charts

Definition 3. *A High-Level Message Sequence Chart (or HMSC) is a pair $(H, \mathcal{I})$ where H is a finite automaton on a set of symbols B, with one initial state and all states final, and $\mathcal{I}$ is a map from B to the set of Basic Message Sequence Charts. This map extends to a unique morphism of monoids from B^* to the monoid of bMSCs, such that $\mathcal{I}(\epsilon) = (\epsilon, \ldots, \epsilon)$ and $\mathcal{I}(uv) = \mathcal{I}(u) \cdot \mathcal{I}(v)$ for $u, v \in B^*$. The language $L(H)$ of the automaton H in the free monoid B^* is called the* meta-language *of the HMSC. The image of $L(H)$ under $\mathcal{I}$, let $\overrightarrow{\mathcal{L}}(H) = \{\mathcal{I}(w) \mid w \in L(H)\}$, is called the* vector language *of the HMSC (it is a subset of $E_1^* \times \cdots \times E_n^*$). The admissible words $w \in E^*$ such that $\delta(w) \in \overrightarrow{\mathcal{L}}(H)$ are called* terminated sequences *of the HMSC. The set of prefixes of the terminated sequences is called the* language *of the HMSC and it is denoted $\mathcal{L}(H)$ (it is a subset of E^*).*

This definition calls for several comments. By considering all states of H as final states and all prefixes of terminated sequences as elements of $\mathcal{L}(H)$, we adopt an operational view on HMSCs as on line computing devices. One could alternatively specify an explicit subset of final states for H and restrict the definition of $\mathcal{L}(H)$ to terminated sequences. Results given in this paper do carry unchanged to this more general setting. A second simplification which is achieved here is to present bMSCs as vectors of words $(w_1, \ldots, w_n) \in E_1^* \times \cdots \times E_n^*$ rather than pomsets labelled on $E = E_1 \cup \cdots \cup E_n$. If one assumes that multiple copies of the *same* message are always received in the order they are sent, this makes no real difference since the partial order on the occurrences of events in the vector $(w_1, \ldots, w_n)$ can actually be reconstructed as soon as this vector is admissible: as all occurrences of events are already ordered in each process, it suffices to make explicit for each message m with respective emitter i and receiver j the ordering

$(w_i, k) < (w_j, l)$ for all occurences (w_i, k) and (w_j, l) of the respective events $!m$ and $?m$ such that $|w_i|_{!m} = |w_j|_{?m}$. Now it is easily seen that a word $w \in E^*$ represents a linear extension of the partial order thus obtained if and only if w is admissible and $(w_1, \ldots, w_n) \in \delta(w)$. Our presentation is therefore consistent with other presentations of HMSCs that may be found in the literature.

1.4 Using HMSCs as behavioural specifications?

The topic of this paper is to try understanding how HMSCs can be used as behavioural specifications of distributed systems to be realized. Let us briefly review a few studies where this question is addressed directly or indirectly. The *matching problem* for MSCs and HMSCs was solved in [21] by Muscholl, Peled and Su. This membership problem is as follows: given a bMSC and a HMSC, does the former belong to the set of bMSCs defined by the latter? It was shown by these authors that the matching problem is NP-complete, while the intersection problem for HMSCs (given two HMSCs, does there exist some bMSC generated by both?) is undecidable. Incidentally, there are significant differences between our HMSCs and those dealt with in [21], where no specific order can be imposed on the reception of two messages unless the emission of one depends on the reception of the other. The negative answer to the intersection problem may be reworded as follows: given a system modelled by a HMSC, one cannot decide whether this system is compatible with the specifications given by another HMSC. One may take this negative result as an indication that most problems for HMSCs are undecidable. We shall see that this intuition is right. A different approach was proposed by Damm and Harel in [5]. One of the ideas developed in that work is to interpret concatenation of bMSCs as an operation that may synchronize agents and that explicitly prevents multiple instances of a bMSC to be entered concurrently. With this interpretation, languages of HMSCs stay within rational languages; this enables the model checking of HMSCs, which is EXP-SPACE complete according to Alur and Yannakakis [1]. HMSCs with this strong form of concatenation may be realized by communicating automata with synchronous control [13]. Although we have chosen here a purely asynchronous framework, we could have set weaker constraints of *boundedness* on HMSCs so that HMSC languages would stay rational, following [1] and [14]. We did not take this option for we feel it does not suit well the field of telecommunications in which HMSCs are used for partial specifications at early design stages. We nevertheless adopt the objective of synthesizing distributed realizations of HMSCs by communicating automata, like was done for bounded HMSCs in [13] and in [14] [15]. Our approach to distribution shares little with the language-theoretical approach proposed in the last two references: we regard communicating automata as systems whose runs or partial runs take place in the real world, and hence cannot be accepted or rejected by a posteriori acceptance conditions.

To end this introduction we give a flavour of the contents of the remaining sections. Section 2 establishes a series of undecidability results for HMSC languages, following from similar results on rational sets in product monoids. We show the undecidability of inclusion and rationality of HMSC languages. More precisely, we show that both inclusion and reverse inclusion between HMSC languages and rational languages are undecidable. This leaves no hope to deal with HMSCs as complete specifications of distributed systems amenable to automated verification, without cutting down HMSCs by strong restrictions. The alternative is to consider general HMSCs as *incomplete* specifications of behaviours. The language of an HMSC should thus be seen as the minimal behaviour required from systems realizing these specifications. The meaning of specifications is now relative to a fixed class of potential realizations, and a main question is to identify classes of realizations in which each HMSC has an optimal realization, determined in a unique way. We give a selective answer to this question in section 3, where we prove that Petri Nets form such a class. We show for this purpose that HMSC languages are semilinear, and it follows from the theory of regions that they have closures in Petri net languages. Hints are given at the issues of distribution and verification, and case studies are finally dealt with before a short conclusion.

2 Undecidability results

It is shown in this section that inclusion and rationality are undecidable for HMSC languages, and similarly for the inclusion and for the reverse inclusion between HMSC languages on the one hand and rational languages on the other hand. In order to obtain these negative results, we shall focus on HMSCs with an empty set of messages, depriving them of communication between processes. We could alternatively eliminate private events and then concentrate on HMSCs with an even number of processes $n = 2k$ where every message is sent from process i ($\leq k$) to process $i + k$, such that process $i + k$ is a replica of process i up to substituting $?m$ for $!m$ for each $m \in M$. The results given in this section apply also to this case (as we shall see). Now, if we assume that M is the empty set of messages, scenarios and bMSCs are *arbitrary* elements of the product monoid $E_1^* \times \cdots \times E_n^*$. As the concatenation of scenarios (resp. bMSCs) agrees by definition with the concatenation in this monoid, vector languages of HMSCs are certainly rational subsets of $E_1^* \times \cdots \times E_n^*$, but they cannot coincide with the latter since we did not equip HMSCs with specific final states. Neither does the distribution map $\delta : E^* \to E_1^* \times \cdots \times E_n^*$ yield a bijective correspondence between languages and vector languages of HMSCs since we imposed on languages of HMSCs to be closed under prefix. Both disagreements derive from our operational view on HMSCs. Notwithstanding, it is possible to reduce undecidable problems on rational subsets of $E_1^* \times \cdots \times E_n^*$ to decision problems on languages of HMSCs, thus proving their undecidability, and this is what is achieved in this section. The organization is as follows. Basic definitions and results about recognizable and rational sets and relations are recalled in 2.1; auxiliary definitions and lemmas needed to compensate for the discrepancies between rational sets and HMSC languages (regarding final states and prefix closure) are stated in 2.2; a series of undecidable problems on rational subsets of $E_1^* \times \cdots \times E_n^*$ are reduced in 2.3 to decision problems on HMSC languages;

the consequences of these reductions on the potential use of HMSCs as system specifications are examined in 2.4.

2.1 Recognizable and rational sets and relations

Let us recall classical definitions and results that may be found in many good books on language theory, e.g. in [3].

Let $E_1, \ldots, E_n$ be finite disjoint alphabets and let $E = \bigcup_{i=1}^{n} E_i$. The free monoid (finitely) generated from E is denoted E^*. The cartesian product $E_1^* \times \cdots \times E_n^*$ of the E_i^* is a monoid with neutral element $(\varepsilon, \ldots, \varepsilon)$ and with composition as follows: $(w_1, \cdots, w_n)(w'_1, \ldots, w'_n) = (w_1 w'_1, \ldots, w_n w'_n)$. This monoid is finitely generated (its generators are vectors of words $(w_1, \cdots, w_n)$ such that $w_i \in E_i$ for some i and $w_j = \varepsilon$ for all $j \neq i$) but it is not a free monoid (the composition of generators is commutative). The subsets of $E_1^* \times \cdots \times E_n^*$ are also called *relations*.

Definition 4. *Let M be a monoid and A a subset of M. A is* recognizable *($A \in Rec(M)$) if there exists a* finite *monoid N, a morphism of monoids $\alpha : M \to N$, and a subset P of N such that $A = \alpha^{-1}(P)$.*

Definition 5. *Let M be a monoid with neutral element $1 \in M$. The family $Rat(M)$ of* rational *subsets of M is the least family of subsets X of M such that :*
a) the empty set $\emptyset$ and every singleton set $\{m\}$ are rational,
b) if A, B are rational then $A \cup B$ and AB are rational,
c) if A is rational then A^* is rational,
where $AB = \{\, \alpha\beta \mid \alpha \in A \ \wedge \ \beta \in B \,\}$ and A^* is the least subset of M such that $X = \{1\} \cup AX$ (hence $\emptyset^* = \{1\}$).

It follows from the definitions that monoid morphisms $\alpha : M \to N$ preserve rationality (if A is a rational subset of M then αA is a rational subset of N) while they reflect recognizability (if αA is a recognizable subset of N then A is a recognizable subset of M). Both notions are ideally linked as follows.

Theorem 1 (Kleene). *Let M be a free monoid. Then $Rec(M) = Rat(M)$.*

Kleene's theorem states that recognizable and rational subsets coincide in free monoids but it does not apply to $E_1^* \times \cdots \times E_n^*$ since this monoid is not free. However $E_1^* \times \cdots \times E_n^*$ is a finitely generated monoid and a weaker theorem still applies.

Theorem 2 (McKnight). *Let M be a finitely generated monoid.*
Then $Rec(M) \subset Rat(M)$.

Another crucial Kleene's theorem connects rational sets with finite automata.

Theorem 3 (Kleene). *The rational subsets of E^* coincide with the languages generated by finite automata with alphabet E.*

It follows that rational subsets of any monoid M must coincide with subsets of M generated by finite automata interpreted in M (the singleton sets $\{m\}$ used to express a rational subset of M form the alphabet of the associated automaton). In the specific case of the monoid $E_1^* \times \cdots \times E_n^*$, the set of generators is the bijective image of $E = \bigcup_{i=1}^n E_i$ by the distribution map δ, and this map is moreover a monoid morphism $\delta : E^* \to E_1^* \times \cdots \times E_n^*$. Seeing that any $m \in M$ may be finitely expressed in terms of generators, it follows clearly that the rational subsets of $E_1^* \times \cdots \times E_n^*$ coincide with the images under δ of the rational subsets of E^*. Let us add a few words about the inverse δ^{-1} of the distribution map.

Definition 6. *Given $A \subseteq E_1^* \times \cdots \times E_n^*$, the* mix *of A is the language $\delta^{-1}(A) = \{w \in E^* \mid \delta(w) \in A\}$.*

Given a HMSC $(H, \mathcal{I})$ with an empty set of messages, the words in $\mathcal{L}(H)$ are the prefixes of the words in the mix of $\overrightarrow{\mathcal{L}}(H)$; if moreover each bMSC $\mathcal{I}(b)$ contains at most one occurrence of event, then $\mathcal{L}(H)$ is equal to the mix of $\overrightarrow{\mathcal{L}}(H)$. Now the main source of problems with HMSC languages lays in that $\mathcal{L}(H)$ may not be rational even though $\overrightarrow{\mathcal{L}}(H)$ is rational. To get convinced of this fact, it suffices to consider e.g. the vector language $(e_1, e_2)^*$.

2.2 Marked sets and Prefix sets

In order to compensate for the mismatch between rational subsets of $E_1^* \times \cdots \times E_n^*$ and vector languages of HMSCs, one may envisage to represent a rational subset A of this monoid as $Pref(A_\top)$ using the following definitions.

Definition 7 (Marked subset). *Given $A \subseteq E_1^* \times \cdots \times E_n^*$ and a set of markers $\{\top_1, \ldots, \top_n\}$ disjoint from E, let $A_\top = \{W \cdot (\top_1, \ldots, \top_n) \mid W \in A\}$.*

Definition 8 (Prefix set). *Given a monoid M and a subset $A \subseteq M$ let $Pref(A) = \{m \in M \mid \exists m' \in M : mm' \in A\}$.*

Lemmas below state that the above suggested representation is faithful, that $Pref(A_\top)$ is rational if A is rational, and that $\delta^{-1}(Pref(A_\top)) = Pref(\delta^{-1}(A_\top))$. Hence, one obtains altogether a representation of rational subsets of $E_1^* \times \cdots \times E_n^*$ by HMSC languages.

Lemma 1. *Let $A, B \subseteq E_1^* \times \cdots \times E_n^*$ then the following inclusions are equivalent :*
a) $A \subseteq B$
b) $A_\top \subseteq B_\top$
c) $Pref(A_\top) \subseteq Pref(B_\top)$

Lemma 2. $A \in Rat(E_1^* \times \cdots \times E_n^*) \Rightarrow Pref(A) \in Rat(E_1^* \times \cdots \times E_n^*)$.

Proof. Let $E = \bigcup_{i=1}^n E_i$. As A is rational, $A = \delta R$ for $R \in Rat(E^*)$ accepted by some finite automaton $\mathcal{A} = (Q, E, T, q_0, Q_F)$. Let $\mathcal{A}' = (Q', E, T', q_0', Q_F')$ with $Q' = Q \times \mathcal{P}([n])$, $q_0' = (q_0, [n])$, $Q_F' = Q_F \times \mathcal{P}([n])$, and with T' ($\subseteq Q' \times E \times Q'$) defined as the least set of transitions such that, for all $i \in [n]$, $e_i \in E_i$, and

$J \subseteq [n]$:
if $q_1 \xrightarrow{e_i} q_2 \in T$, then:
$(q_1, J) \xrightarrow{e_i} (q_2, J) \ \in T'$ if $i \in J$, and
$(q_1, J) \xrightarrow{\varepsilon} (q_2, J \setminus \{i\}) \ \in T'$ in any case.

Clearly, $Pref\,(A) = \delta R'$ where R' is the rational subset accepted by $\mathcal{A}'$, hence $Pref\,(A)$ is rational. □

Lemma 3. *$Pref \circ \delta^{-1}(A) = \delta^{-1} \circ Pref(A)$ for any $A \subseteq E_1^* \times \cdots \times E_n^*$.*

Proof. Let $W' \in Pref(A)$ then by definition of Prefix sets, $W = W' \cdot W''$ for some $W \in A$ and $W'' \in E_1^* \times \cdots \times E_n^*$. Since δ is a morphism of monoids, $\delta^{-1}\{W'\} \cdot \delta^{-1}\{W''\} \subseteq \delta^{-1}\{W\}$, showing that $\delta^{-1} \circ Pref(A) \subseteq Pref \circ \delta^{-1}(A)$. Let $w' \in Pref \circ \delta^{-1}(A)$ then by definition of Prefix sets, $w = w'w''$ for some $w \in \delta^{-1}(A)$ and $w'' \in E^*$. Let $W' = \delta(w')$ and $W'' = \delta(w'')$. Since δ is a morphism of monoids, $\delta(w'w'') = W' \cdot W'' \in A$, hence $Pref \circ \delta^{-1}(A) \subseteq \delta^{-1} \circ Pref(A)$. □

The following fact is also used.

Lemma 4. *Let M be a monoid. If $A \in Rec(M)$ then $Pref(A) \in Rec(M)$.*

Proof. Let $A \in Rec(M)$, let $\alpha : M \to N$ be a morphism from M into a finite monoid N, and let P be a subset of N such that $A = \alpha^{-1}(P)$. Then $Pref(A) = \alpha^{-1}(P')$ where $P' = \{n \in N \mid \exists m \in M : n.\alpha(m) \in P\}$. □

2.3 A reduction yielding negative decision results for HMSCs

We recall first a classical theorem claiming the undecidability of several questions about rational relations (we refer the reader to [12] or to [3] p.90 for the proof of this theorem that relies on the undecidability of Post's Correspondence Problem). We establish next a reduction of these questions to similar questions on HMSC languages, showing that the latter are undecidable.

Theorem 4 (Fischer-Rosenberg). *Let X, Y be alphabets with at least two letters. Given rational subsets $A, B \subseteq X^* \times Y^*$, it is undecidable to determine whether:*

i) $A \cap B = \emptyset$;
ii) $A \subseteq B$;
iii) $A = B$;
iv) $A = X^* \times Y^*$;
v) $(X^* \times Y^*) \setminus A$ *is finite*;
vi) A *is recognizable.*

Theorem 5. *Let $E = \bigcup_{i=1}^n E_i$ be an alphabet of events partitioned into subalphabets $E_1, \ldots, E_n$ such that $n \geq 2$ and each alphabet E_i defines at least three private events for process i. Given two HMSCs H_1 and H_2 over the alphabet E, and given a rational subset $R \subseteq E^*$, it is undecidable to determine whether*

i) $\mathcal{L}(H_1) = \mathcal{L}(H_2)$;
ii) $\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$;
iii) $R \subseteq \mathcal{L}(H_2)$;
iv) $\mathcal{L}(H_1) \subseteq R$;
v) $\mathcal{L}(H_1) \subseteq \delta^{-1}\delta R$;
vi) $\mathcal{L}(H_1) = R$;
vii) $\mathcal{L}(H_1)$ *is rational* .

In order to establish Theo. 5, we will show that each problem in the above list amounts to a reduction of some undecidable problem among problems (ii,iii,iv,vi) from Theo. 4 (a different proof of the undecidability of (vii) was sketched in [14]). So, let $A, B \in Rat(X^* \times Y^*)$ where X and Y are disjoint alphabets with size at least 2. We prepare the way to the reductions by constructing first HMSCs H_1 and H_2 such that $\mathcal{L}(H_1) = \delta^{-1} \circ Pref(A_\top)$ and $\mathcal{L}(H_2) = \delta^{-1} \circ Pref(B_\top)$ (where $A_\top$ and $B_\top$ are marked sets). The alphabet of these HMSCs is $E = E_1 \cup E_2$ with $E_1 = X \cup \{\top_1\}$ and $E_2 = Y \cup \{\top_2\}$, letting $\top_1$ and $\top_2$ be distinct markers (such that $\{\top_1, \top_2\} \cap (X \cup Y) = \emptyset$). All events in E_i are private events of process i (for $i \in [2]$). The construction of H_1 is described hereafter.

As $A \in Rat(X^* \times Y^*)$, the marked set $A_\top$ is a rational subset of ${E_1}^* \times {E_2}^*$, hence $A_\top = \delta R$ for some $R \in Rat(E^*)$. Let $\mathcal{A}$ be a finite automaton on the alphabet E such that $R = L(\mathcal{A})$ (the language generated by automaton $\mathcal{A}$) and the reachability set of each state of $\mathcal{A}$ includes a nonempty subset of final states. Let H_1 be the automaton on E that derives from $\mathcal{A}$ by making all states final so that $L(H_1) = Pref(R)$. The HMSC associated with A is the pair $(H_1, \mathcal{I})$ where the interpretation map $\mathcal{I} : E \to {E_1}^* \times {E_2}^*$ is defined as the restriction on E ($\subseteq E^*$) of the distribution map $\delta : E^* \to {E_1}^* \times {E_2}^*$ (hence $\mathcal{I}(x) = (x, \varepsilon)$ for $x \in X \cup \{\top_1\}$ and $\mathcal{I}(y) = (\varepsilon, y)$ for $y \in Y \cup \{\top_2\}$). Lemmas below show that the construction works as expected.

Lemma 5. $Pref \circ \delta L(H_1) = Pref(A_\top)$

Proof. $A_\top = \delta R \Rightarrow R \subseteq \delta^{-1} A_\top \Rightarrow Pref(R) \subseteq Pref \circ \delta^{-1}(A_\top) \Rightarrow Pref(R) \subseteq \delta^{-1} \circ Pref(A_\top)$ (lemma 3) $\Rightarrow L(H_1) \subseteq \delta^{-1} \circ Pref(A_\top) \Rightarrow \delta L(H_1) \subseteq \delta\delta^{-1} \circ Pref(A_\top) \Rightarrow \delta L(H_1) \subseteq Pref(A_\top)$ ($\delta\delta^{-1}$ is the identity) $\Rightarrow Pref \circ \delta(L(H_1)) \subseteq Pref(A_\top)$. Conversely, $A_\top = \delta R \subseteq \delta L(H_1) \Rightarrow Pref(A_\top) \subseteq Pref \circ \delta(L(H_1))$.

Lemma 6. $\mathcal{L}(H_1) = Pref \circ \delta^{-1}\delta(L(H_1))$.

Proof. As the interpretation map of H_1 has been defined as the restriction of δ on E, and seeing that every word in $\delta^{-1}\delta(L(H_1))$ is admissible because H_1 has an empty set of messages, this follows directly form the definition of HMSCs. □

Lemma 7. $\mathcal{L}(H_1) = \delta^{-1} \circ Pref(A_\top)$

Proof. $\mathcal{L}(H_1) = Pref \circ \delta^{-1}\delta(L(H_1))$ (lemma 6)
$= \delta^{-1} \circ Pref \circ \delta(L(H_1))$ (lemma 3) $= \delta^{-1} \circ Pref(A_\top)$ (lemma 5). □

Lemma 8. $\delta^{-1}\delta(\mathcal{L}(H_1)) = \mathcal{L}(H_1)$.

Proof. $\delta^{-1}\delta(\mathcal{L}(H_1)) = \delta^{-1}\delta \circ Pref \circ \delta^{-1}\delta(L(H_1))$ (lemma 7) $= \delta^{-1}\delta\delta^{-1} \circ Pref \circ \delta(L(H_1))$ (lemma 3) $= \delta^{-1} \circ Pref \circ \delta(L(H_1))$ ($\delta\delta^{-1}$ is the identity) $= Pref \circ \delta^{-1}\delta(L(H_1))$ (lemma 3) $= \mathcal{L}(H_1)$ (lemma 6) □

Proof of theorem 5. Given $A, B \in Rat(X^* \times Y^*)$ let H_1, H_2 be HMSCs such that $\mathcal{L}(H_1) = \delta^{-1} \circ Pref(A_\top)$ and $\mathcal{L}(H_2) = \delta^{-1} \circ Pref(B_\top)$. We show that (i) to (vii) are reductions of undecidable problems on A or B or A and B.
$*$ *ad (i)* $A = B$ iff $Pref(A_\top) = Pref(B_\top)$ (lemma 1) iff
$\mathcal{L}(H_1) = \mathcal{L}(H_2)$ (lemma 7, seeing that $\delta\delta^{-1}$ is the identity).
The undecidability of $\mathcal{L}(H_1) = \mathcal{L}(H_2)$ follows from (iii) in Theo. 4.
$*$ *ad (ii)* this is an obvious consequence of (i).
$*$ *ad (iii)* $A \subseteq B$ iff $Pref(A_\top) \subseteq Pref(B_\top)$ (lemma 1) iff
$\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$ (lemma 7, seeing that $\delta\delta^{-1}$ is the identity).
We claim that $\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$ iff $L(H_1) \subseteq \mathcal{L}(H_2)$.
The direct implication follows from
$L(H_1) \subseteq \delta^{-1}\delta L(H_1) \subseteq Pref \circ \delta^{-1}\delta L(H_1) = \mathcal{L}(H_1)$ (lemma 6).
For the reverse implication, note that $\mathcal{L}(H_2) = Pref \circ \delta^{-1}\delta(\mathcal{L}(H_2))$
(as $\mathcal{L}(H_2) = Pref(\mathcal{L}(H_2))$ and by lemma 8 $\mathcal{L}(H_2) = \delta^{-1}\delta\mathcal{L}(H_2)$)
and $\mathcal{L}(H_1) = Pref \circ \delta^{-1}\delta(L(H_1))$ (lemma 6),
hence $L(H_1) \subseteq \mathcal{L}(H_2) \Rightarrow \mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$.
The undecidability of $R \subseteq \mathcal{L}(H_2)$ follows from (ii) in Theo. 4 with $R = L(H_1)$.
$*$ *ad (iv)* $A \cap B \neq \emptyset$ iff $A_\top \cap B_\top \neq \emptyset$ iff $Pref(A_\top) \cap B_\top \neq \emptyset$ iff
$Pref \circ \delta(\mathcal{L}(H_1)) \cap B_\top \neq \emptyset$ (lemma 7, seeing that $\delta\delta^{-1}$ is the identity).
Let $S \in Rat(E^*)$ such that $B_\top = \delta S$.
We claim that $Pref \circ \delta(\mathcal{L}(H_1)) \cap B_\top \neq \emptyset$ iff $\mathcal{L}(H_1) \cap S \neq \emptyset$.
For the direct implication, we observe:
$Pref \circ \delta(\mathcal{L}(H_1)) \cap B_\top \neq \emptyset \Rightarrow S \cap \delta^{-1} \circ Pref \circ \delta(\mathcal{L}(H_1)) \neq \emptyset \Rightarrow$
$S \cap Pref \circ \delta^{-1}\delta(\mathcal{L}(H_1)) \neq \emptyset$ (lemma 3) $\Rightarrow S \cap Pref(\mathcal{L}(H_1)) \neq \emptyset$ (lemma 8)
$\Rightarrow S \cap \mathcal{L}(H_1) \neq \emptyset$ (since $\mathcal{L}(H_1) = Pref(\mathcal{L}(H_1))$).
For the reverse implication, we note that $\mathcal{L}(H_1) = Pref(\mathcal{L}(H_1))$
and that $\delta \circ Pref(\mathcal{L}(H_1)) \subseteq Pref \circ \delta(\mathcal{L}(H_1))$ because δ is a morphism of monoids,
hence $\mathcal{L}(H_1) \cap S \neq \emptyset \Rightarrow \delta(\mathcal{L}(H_1)) \cap B_\top \neq \emptyset \Rightarrow Pref \circ \delta(\mathcal{L}(H_1)) \cap B_\top \neq \emptyset$.
Altogether $A \cap B = \emptyset$ iff $\mathcal{L}(H_1) \cap S = \emptyset$ iff $\mathcal{L}(H_1) \subseteq R$ where $R = E^* \setminus S$.
The undecidability of $\mathcal{L}(H_1) \subseteq R$ follows from (i) in Theo. 4.
$*$ *ad (v)* $A \subseteq B$ iff $Pref(A_\top) \subseteq Pref(B_\top)$ (lemma 1) iff
$\mathcal{L}(H_1) \subseteq \delta^{-1} \circ Pref(B_\top)$ (lemma 7).
As $Pref(B_\top) \in Rat({E_1}^* \times {E_2}^*)$, $Pref(B_\top) = \delta R$ for some $R \in Rat(E^*)$.
Thus $A \subseteq B$ iff $\mathcal{L}(H_1) \subseteq \delta^{-1}\delta R$ and the undecidability of $\mathcal{L}(H_1) \subseteq \delta^{-1}\delta R$ follows from (ii) in Theo. 4.
$*$ *ad (vi)* $A = X^* \times Y^*$ iff $Pref(A_\top) = Pref(X^*\top_1 \times Y^*\top_2)$
iff $\delta^{-1} \circ Pref(A_\top) = R = \mathcal{L}(H_1)$ (lemma 7) where we let
$R = Pref((X \cup Y)^* \top_1 (X \cup Y)^* \top_2 \cup (X \cup Y)^* \top_2 (X \cup Y)^* \top_1)$.
The undecidability of $\mathcal{L}(H_1) = R$ follows from (iv) in Theo. 4.
$*$ *ad (vii)* We show that $A \in Rec(X^* \times Y^*)$ if and only if $\mathcal{L}(H_1) \in Rat(E^*)$.
The direct implication may be established as follows:
$A \in Rec(X^* \times Y^*) \Rightarrow A \in Rec(E_1^* \times E_2^*) \Rightarrow A_\top \in Rec(E_1^* \times E_2^*)$

(as the recognizable sets are closed under composition ([3], p.61) and seeing that $(\top_1, \top_2) \in Rec(E_1^* \times E_2^*))$
$\Rightarrow Pref(A_\top) \in Rec(E_1^* \times E_2^*)$ (lemma 4) $\Rightarrow \delta^{-1}(Pref(A_\top) \in Rec(E^*)$
(as the recognizable sets are closed under inverse morphisms ([3], p.53) and $\delta : E^* \to E_1^* \times E_2^*$ is a monoid morphism)
$\Rightarrow \mathcal{L}(H_1) \in Rec(E^*)$ (lemma 7) $\Rightarrow \mathcal{L}(H_1) \in Rat(E^*)$ (Theo 2).
The reverse implication may be established as follows:
$\mathcal{L}(H_1) \in Rat(E^*) \Rightarrow \mathcal{L}(H_1) \in Rec(E^*)$ (Theo 1)
$\Rightarrow Pref(\mathcal{L}(H_1)) \in Rec(E^*)$ (lemma 4)
$\Rightarrow Pref \circ \delta^{-1} \circ Pref(A_\top) \in Rec(E^*)$ (lemma 7)
$\Rightarrow \delta^{-1} \circ Pref(A_\top) \in Rec(E^*)$ (lemma 3)
$\Rightarrow Pref(A_\top) \in Rec(E_1^* \times E_2^*)$
(by direct application of a proposition stated in [8] (p.33) and [9] (p.172) for morphisms from a free monoid E^* onto an arbitrary monoid, applied here to the surjective morphism $\delta : E^* \to E_1^* \times E_2^*$)
$\Rightarrow Pref(A_\top) \cap (E_1^* \times E_2^*) \cdot (\top_1, \top_2) \in Rec(E_1^* \times E_2^*)$
(the recognizable sets are closed under intersection)
$\Rightarrow A_\top \in Rec(E_1^* \times E_2^*)$ (as $A_\top = Pref(A_\top) \cap (E_1^* \times E_2^*) \cdot (\top_1, \top_2))$.
Thus there exists a finite monoid N, a morphism α from $(E_1^* \times E_2^*)$ into N, and a subset P of N such that $A_\top = \alpha^{-1}(P)$. Let α' be the restriction of α on $(X^* \times Y^*)$ and let $P' = \{s \in N \mid s.\alpha(\top_1, \top_2) \in P\}$ then clearly $A = {\alpha'}^{-1}(P')$, and therefore $A \in Rec(X^* \times Y^*)$.
We have thus shown that $A \in Rec(X^* \times Y^*)$ iff $\mathcal{L}(H_1) \in Rat(E^*)$ and the undecidability of $\mathcal{L}(H_1) \in Rat(E^*)$ follows from (vi) in Theo. 4. $\square$

We will now show that the undecidability of relations (i) to (v) in Theo. 5 extends to purely communicating HMSCs (i.e. such that all events are emissions or receptions). This may be done by reducing undecidable problems on non communicating HMSCs to decision problems on purely communicating HMSCs. Given (non communicating) HMSCs $H_1 = (H_1, \mathcal{I})$ and $H_2 = (H_2, \mathcal{I})$ on the set of events $E = E_1 \cup E_2$ with an interpretation map $\mathcal{I} : E \to E_1^* \times E_2^*$ such that $\mathcal{I}(e) = \delta(e)$ for all $e \in E$, define (communicating) HMSCs $H_1' = (H_1, \mathcal{I}')$ and $H_2' = (H_2, \mathcal{I}')$ on the same automata H_1 and H_2 by choosing a new interpretation map $\mathcal{I}'$ such that $\mathcal{I}'(e) = (!e, \varepsilon, ?e, \varepsilon)$ for $e \in E_1$ and $\mathcal{I}'(e) = (\varepsilon, !e, \varepsilon, ?e)$ for $e \in E_2$. Thus the $\mathcal{I}'(e)$ are bMSCs on the set of events $E' = R \cup S$ where $S = \{!e \mid e \in E\}$ and $R = \{?e \mid e \in E\}$. It is easily seen that for $i \in \{1, 2\}$, $\mathcal{L}(H_i') = \cup \{\, !(u) \sqcup\!\sqcup\, ?(v) \mid u \in \mathcal{L}(H_i) \;\&\; v \in pref(u) \,\}$ where $! : E^* \to S^*$ and $? : E^* \to R^*$ are the respective morphisms such that $!(e) = !e$ and $?(e) = ?e$, while $\sqcup\!\sqcup$ is the shuffle operator. As a consequence:
$\mathcal{L}(H_1) \subseteq \mathcal{L}(H_2)$ iff $\mathcal{L}(H_1') \subseteq \mathcal{L}(H_2')$,
$A \subseteq \mathcal{L}(H_2)$ iff $!(A) \subseteq \mathcal{L}(H_2')$ for $A \in Rat(E^*)$,
$\mathcal{L}(H_1) \subseteq A$ iff $\mathcal{L}(H_1') \subseteq !(A) \sqcup\!\sqcup R^*$ for $A \in Rat(E^*)$,
$\mathcal{L}(H_1) \subseteq \delta^{-1}\delta A$ iff $\mathcal{L}(H_1') \subseteq \delta^{-1}\delta(!(A) \sqcup\!\sqcup R^*)$ for $A \in Rat(E^*)$,
and the undecidability of relations (i) to (v) for purely communicating HMSCs follows from Theo. 5 (seeing that $A \sqcup\!\sqcup R^* \in Rat(E'^*)$ for $A \in Rat(S^*)$).

2.4 Should HMSCs be considered as specifications?

There are several ways to consider HMSCs as specifications of distributed systems. By interpreting HMSCs as abstract generators for languages $\mathcal{L}(H)$, we admittedly restrict the scope of our investigations in this respect, but the generated languages may be given at least three different meanings: $\mathcal{L}(H)$ may be considered as a subset of the behaviour of the specified system (specifications of service), as an exact definition of the behaviour of the specified system (complete specifications), or as a superset of the behaviour of the specified system (specifications of safety). Needless to say, all interpretations may co-exist in a logical framework for distributed system specification based on HMSCs. Sharing the common opinion that any practical specification framework should enable some decision of conformity of systems with specifications, let us examine these various interpretations under the light shed by Theo. 5.

By (ii) in Theo. 5 one cannot check HMSCs considered as representations of systems against HMSCs expressing safety conditions or service requirements. By (iii) and (iv) in Theo. 5, one cannot check bounded systems against HMSCs expressing safety conditions or service requirements; and one cannot check HMSCs considered as representations of systems against regular safety conditions or regular service requirements. As Theo. 5 is not conclusive for star-free regular languages, this leaves open the problem of checking HMSCs against linear temporal logic formulas (but we bet the situation is not better). Finally, by (vii) in Theo. 5 one cannot decide whether a given HMSC generates a rational language (in which case most difficulties vanish since we are brought back to the realm of effective boolean algebras).

The above observations leave two ways out. One is to impose constraints on HMSCs strong enough to guarantee that languages of HMSCs are kept within rational languages [1] [5] [14]. The price to pay is to give up with unbounded systems, nicely modelled with HMSCs and largely present as partial specifications of telecommunication systems. The alternative way, yet unexplored, is to consider languages of HMSCs as specifications of minimal service for a given class of (potentially) unbounded systems. The price to pay is to accept that the realized behaviours may be strictly *larger* than the specified behaviours. The difference between realized behaviours and specified behaviours is an inverse measure for the quality of realizations.

An ideal class of unbounded systems for the realization of HMSC specifications should come equipped with an effective procedure able to synthesize optimal realizations of all HMSC specifications; it should also allow for model-checking systems against safety assertions in order to verify that the added part of the realized behaviours makes no problem. These criteria are demanding but they are reasonable. Next section shows that they are met by Petri nets and more importantly by *distributable* Petri nets, a variety of Petri nets that translate easily to clusters of automata communicating by asynchronous message passing. We do *not* claim that HMSC specifications are implemented at best via Petri net synthesis. Petri nets are used only as an illustration aiming to show that general HMSCs may really play a central role in the design of distributed systems. The search for other (more) adequate classes of realizations for HMSC specifications is an open direction for further work.

3 Petri net realization of HMSC languages

It is shown in this section that HMSC languages $\mathcal{L}(H)$ may be mapped to closures $\overline{\mathcal{L}}(H)$ in the class $\mathbf{G_0}^{\mathbf{f}}$ of (free) Petri net languages [23], such that $\overline{\mathcal{L}}(H) = \mathcal{L}(N)$ for some net N effectively computed from H, thus providing an optimal realization of H. In order to get distributed realizations of HMSCs, we will specialize the construction of closures to *distributable* Petri nets, that may be compiled to distributed implementations on an asynchronous network. Petri nets may be model-checked against safety assertions, hence they are an adequate class of realizations for HMSC specifications, according to the criteria stated in section 2. Two results will complete the picture: on the negative side, the problem whether a HMSC language may be realized exactly by some Petri net is undecidable; on the positive side, Petri net realizations of HMSCs may be model-checked against (possibly non regular) safety assertions represented as Petri nets.

It is not the goal of the paper to give a presentation on the topic of Petri net synthesis, hence we shall only present here the necessary results (for more on the topic, the reader may consult [2] [4] [6] [7]). The corner-stone of the methods developed so far for deriving Petri nets from formal languages is the semilinearity of their commutative images. The main contribution of the section is to show that commutative images of HMSC languages are precisely semilinear.

The section is organized as follows. The semilinearity of commutative images of HMSC languages is established in 3.1; using this fact, Petri net closures of HMSC languages are constructed in 3.2; distributable Petri nets are considered in 3.3; the undecidability of the Petri net synthesis problem for HMSC languages is proved in 3.4; the issue of model-checking realizations of HMSCs is finally addressed in 3.5.

3.1 Commutative images of HMSC languages are semilinear

First, let us recall the definitions of linear subsets and semilinear subsets of a monoid.

Definition 9. *Let M be a monoid. A subset of M is* linear *if it may be expressed as $m \cdot P^*$ where $m \in M$ and P is a finite subset of M. A* semilinear *subset of M is a finite union of linear subsets of M.*

Let $k = |E|$ where E is the alphabet of events fixed for HMSCs. The monoid we shall consider here is $\mathbb{N}^k$ (with the all zero k-vector as the neutral element and the addition of k-vectors as the composition operation). Words $w \in E^*$ may be sent into $\mathbb{N}^k$ by counting the occurences of each letter $e_i \in E$, resulting in an k-vector $\psi(w) = (|w|_{e_1}, \ldots, |w|_{e_k})$ that represents the commutative image of w; the mapping $\psi : E^* \to \mathbb{N}^k$, known as Parikh mapping, is actually a monoid morphism. We aim at showing that for any HMSC H with set of events E, the Parikh image $\psi\mathcal{L}(H)$ of the language of H is a semilinear subset of $\mathbb{N}^k$. To that effect, we shall use the crucial fact that semilinear subsets and rational subsets coincide in any commutative monoid [10] (hence in particular in $\mathbb{N}^k$) and a series of technical lemmas about scenarios. In the sequel, notations are like those in section 1.1, except that $u \leq w$ and $U \leq W$ are used as abbreviations for $\exists v \;\; w = u \cdot v$ and $\exists V \;\; W = U \cdot V$ respectively in the (unit divisor free) monoids E^* and $E_1^* \times \cdots \times E_n^*$.

Lemma 9. *Let $U \leq W$ where U and W are scenarios. For any admissible word u such that $U = \delta(u)$ there exists an admissible word w such that $u \leq w$ and $W = \delta(w)$.*

Proof. Let $W = U \cdot V$ (V is not necessarily admissible). By induction on the size of V, with the trivial case $W = U$ as a basis, it suffices to establish: $W \neq U \Rightarrow \exists e \in E$ such that $\delta(e) \leq V$ and ue is admissible. We proceed with a proof by contradiction. Suppose $\forall e \in E \;\; \delta(e) \leq V \Rightarrow ue$ is not admissible. As u is admissible $\delta(e) \leq V$ entails $e = ?m$ for some m such that $?m$ and $!m$ occur an equal number of times in u. Let $W = (w_1, \ldots, w_n)$, $U = (u_1, \ldots, u_n)$ and $I = \{i \in [n] \,|\, w_i \neq u_i\}$. Thus $I \neq \emptyset$ and for each $i \in I$, $w_i = u_i \cdot ?m_i \cdot u'_i$ where $?m_i$ and $!m_i$ occur an equal number of times in u. Fix some admissible word w such that $W = \delta(w)$ (there must exist such words since W is admissible). Let $w' \leq w$ be the least prefix of w such that $|u|_{?m_i} < |w'|_{?m_i}$ for some $i \in I$. If we set $\delta(w') = (w'_1, \ldots, w'_n)$ then necessarily $w'_i = u_i \cdot ?m_i$ for some $i \in I$, and $w'_j \leq u_j$ for all $j \neq i$. Now let $m = m_i$ (hence $\phi(?m) = i$) and $\phi(!m) = k$. As w' is admissible, $|w'|_{?m} \leq |w'|_{!m}$. Seeing that $|w'|_{!m} = |w'_k|_{!m} \leq |u_k|_{!m} = |u|_{!m}$ and $|w'|_{?m} = |w'_i|_{?m} = 1 + |u_i|_{?m} = 1 + |u|_{?m}$ it follows that $|u|_{!m} \geq 1 + |u|_{?m}$, contradicting $m = m_i$ or the fact that $!m_i$ and $?m_i$ occur an equal number of times in u. ☐

Lemma 10. *Let $H = (H, \mathcal{I})$ be a HMSC on E whose underlying automaton has set of symbols B (hence $\mathcal{I}$ maps symbols $b \in B$ to bMSCs on E). A word $u \in E^*$ belongs to the language of the HMSC ($u \in \mathcal{L}(H)$) if and only if it is admissible and there exists some word $\beta \in B^*$ accepted by the underlying automaton ($\beta \in L(H)$) such that $\delta(u) \leq \mathcal{I}(\beta)$.*

Proof. Suppose $u \in \mathcal{L}(H)$, then by definition, $u \leq w$ for some admissible word w such that $\delta(w) = \mathcal{I}(\beta)$ for some $\beta \in L(H)$, and prefixes of admissible words are admissible. Conversely, let u be an admissible word and suppose that $\delta(u) \leq \mathcal{I}(\beta)$ for some $\beta \in L(H)$. It then follows directly from lemma 9 that $u \in \mathcal{L}(H)$. ☐

Lemma 11. *Let $W = W_1 \cdot \ldots \cdot W_m$, where for each $i \in [m]$, $W_i = (w_{i_1}, \ldots, w_{i_n})$ is a bMSC (thus W is a bMSC). A vector $U \in E_1^* \times \cdots \times E_n^*$ is an admissible prefix of W if and only if it may decomposed as $U = W'_1 \cdot \ldots \cdot W'_m$ such that: for all $i \in [m]$, $W'_i = (w'_{i_1}, \ldots, w'_{i_n})$ is an admissible prefix of W_i and for all $p \in [n]$, $w'_{i_p} \neq w_{i_p} \Rightarrow w'_{j_p} = \varepsilon$ for all $j > i$.*

Proof. The delicate part of the proof is the direct implication. The reverse relation may be established as follows. Let $U = W'_1 \cdot \ldots \cdot W'_m$ as above, hence $U \leq W$. For each $i \in [m]$ let w'_i be an admissible word such that $W'_i = \delta(w'_i)$ (such words must exist by definition) and let $u = w'_1 \cdot \ldots w'_m$. Then $U = \delta(u)$ and u is an admissible word, and U is therefore an admissible prefix of W. Let us show now the direct implication. Every prefix U of $W = W_1 \cdot \ldots \cdot W_m$ has a unique decomposition $U = W'_1 \cdot \ldots \cdot W'_m$ satisfying all requirements of the lemma but the admissibility of the W'_i. We shall prove by induction on the size of U, with the trivial case $U = \delta(\varepsilon)$ as a basis, that this requirement is met when U is an admissible prefix of W. Hence let $u = ve$ (with $e \in E$) be an admissible word such that $U = \delta(u)$, and assume by induction that $\delta(v) = W''_1 \cdot \ldots \cdot W''_m$ and the W''_i satisfy all conditions expressed in the lemma (for W'_i). Since $\delta(u) = \delta(v) \cdot \delta(e)$ and the considered decomposition of $\delta(v)$ is unique, there must exist $i \in [m]$ such that $W'_i = W''_i \cdot \delta(e)$ and $W'_j = W''_j$ for $j \neq i$ (hence W'_j is admissible for $j \neq i$). It remains to show that W'_i is admissible. For the sake of contradiction, suppose the opposite. Let $W''_i = (w''_{i_1}, \ldots, w''_{i_n})$, and let $w''_i \in E^*$ be an admissible word such that $W''_i = \delta(w''_i)$. From our supposition $w''_i e$ is not admissible. As w''_i is admissible, the only possibility is that $e = ?m$ for some m such that $!m$ and $?m$ occur an equal number of times in w''_i. Hence, assuming $\phi(?m) = p$ and $\phi(!m) = q$, we have: $|w'_{i_p}|_{?m} = 1 + |w''_{i_p}|_{?m} = 1 + |w''_{i_q}|_{!m} = 1 + |w'_{i_q}|_{!m}$. Now $W'_i \leq W_i \Rightarrow w'_{i_p} \leq w_{i_p} \Rightarrow |w'_{i_p}|_{?m} \leq |w_{i_p}|_{?m}$, and $|w_{i_p}|_{?m} = |w_{i_q}|_{!m}$ since W_i is a bMSC (closed scenario). Altogether we obtain the inequality $1 + |w'_{i_q}|_{!m} \leq |w_{i_q}|_{!m}$. Hence $w'_{i_q} \neq w_{i_q}$ and therefore $w'_{j_q} = \varepsilon$ for $j > i$. Moreover, $w'_{i_p} \neq \varepsilon \Rightarrow w'_{j_p} = w_{j_p}$ for $j < i$. Recalling that $|w_{j_p}|_{?m} = |w_{j_q}|_{!m}$ for all j and summing up one obtains the inequality $\sum_k |w'_{k_q}|_{!m} < \sum_{k \leq i} |w'_{k_p}|_{?m}$. It follows from this inequality that $|u|_{!m} < |u|_{?m}$ for any word u such that $U = \delta(u)$, hence U is not an admissible prefix of W, which contradicts the assumption. □

One can now easily derive from any HMSC H on E a finite automaton $\overline{H}$ on $E_1^* \times \cdots \times E_n^*$ whose generated language of vectors is the set of admissible prefixes of vectors in $\overrightarrow{\mathcal{L}}(H)$. The construction is sketched below.

Let $H = (A, \mathcal{I})$ where the automaton A has set of symbols B and $\mathcal{I}$ maps symbols $b \in B$ to bMSCs on E. Let $A = (S, B, T, s_0)$ where S is the set of states, $s_0 \in S$ is the initial state, and $T \subseteq S \times B \times S$ is the set of transitions. Then $\overline{H} = (\overline{S}, \overline{B}, \overline{T}, \overline{s_0})$ where $\overline{S} = S \times \mathcal{P}[n]$ (the second component represents a collection of dead agents), $\overline{B} \subseteq E_1^* \times \cdots \times E_n^*$ is the set of admissible prefixes of the bMSCs $\mathcal{I}(b)$ for b ranging over B, $\overline{s_0} = (s_0, \emptyset)$, all states are final, and $\overline{T}$

is the least set of transitions such that, for all $J, J' \subseteq [n]$ and $V \in \overline{B}$:
if $s \xrightarrow{b} s'$ in A then $(s', J) \xrightarrow{V} (s, J')$ in $\overline{H}$ whenever
V is an admissible prefix of $\mathcal{I}(b)$ and the following hold,
letting $V = (v_1, \ldots, v_n)$ and $\mathcal{I}(b) = (w_1, \ldots, w_n)$:
* $v_j = \varepsilon$ for all $j \in J$,
* $J \subseteq J'$,
* $v_j \neq w_j \Rightarrow j \in J'$.

The correctness of the construction follows directly from lemma 11.

The automaton $\overline{H}$ can easily be transformed into an automaton $\psi(\overline{H})$ on $\mathbb{N}^k$, whose set of accepted vectors is the set of Parikh images of the words in $\mathcal{L}(H)$. The transformation consists in replacing labels $V \in \overline{B}$ by corresponding vectors $\psi(V) \in \mathbb{N}^k$, where the Parikh mapping $\psi : E^* \to \mathbb{N}^k$ is extended to $E_1^* \times \cdots \times E_n^*$ by setting $\psi(v_1, \ldots, v_n) = \psi(v_1) + \ldots + \psi(v_n)$. The correctness of the construction follows directly from lemmas 9 and 10.

Recalling that rational subsets and semilinear subsets coincide in $\mathbb{N}^k$ (where this correspondence is effective in both directions), we have obtained a complete proof of the following theorem.

Theorem 6. *Let H be a HMSC, then $\psi\mathcal{L}(H)$ is effectively semilinear.*

A little more is needed if we want to compute closures of HMSC languages with respect to general Petri nets. We need for each $e \in E$ a semilinear expression of the Parikh image of the *e-terminating* sublanguage $\mathcal{L}(H) \cap E^*e$. A semilinear expression of $\psi(\mathcal{L}(H) \cap E^*e)$ may be obtained as shown for $\psi\mathcal{L}(H)$ by specializing the basic automaton $\overline{H}$ according to e. For $e \in E$, let $\overline{H}_e$ be the automaton that derives from H by carrying the following list of modifications:
* each state (s, J) is replaced with two states $(s, J, 0)$ and $(s, J, 1)$,
* $(s_0, \emptyset, 0)$ is the initial state,
* the states $(s, J, 1)$ are the final states,
* $(s, J) \xrightarrow{V} (s', J')$ splits to $(s, J, 0) \xrightarrow{V} (s', J', 0)$ and $(s, J, 1) \xrightarrow{V} (s', J', 1)$,
* $(s, J, 0) \xrightarrow{V} (s', J', 1)$ is added for each transition $(s, J) \xrightarrow{V} (s', J')$ such that $V = \delta(ue)$ for some admissible word ue with $\phi(e) \in J'$.

Verifying that the set of vectors accepted by $\psi(\overline{H}_e)$ is equal to the Parikh image of $\mathcal{L}(H) \cap E^*e$ is left to the reader (note: use the fact that bMSCs are closed scenarios).

3.2 Petri net closures of HMSC languages

This part recalls the definition of Petri nets, brings in a general theorem that connects Parikh semilinear languages with Petri nets, and applies this theorem to HMSC languages in order to define and compute their Petri net closures.

Definition 10. *A* Petri net (system) *is a quadruple* $N = (P, E, F, M_0)$ *where: P and E are finite disjoint sets of* places *and* events, $F : (P \times E) \cup (E \times P) \to \mathbb{N}$, *and* $M_0 : P \to \mathbb{N}$. *Maps* $M : P \to \mathbb{N}$ *are called* markings. M_0 *is the* initial *marking. The net is* pure *if for all* e *and* p, $F(p,e) = 0 \vee F(e,p) = 0$. *An event* e *may be* fired *at* M *if* $(\forall p \in P)\ F(p,e) \leq M(p)$. *The firing of* e *results in a* transition $M\,[\,e > M'$ *such that* $(\forall p \in P)\quad M'(P) = M(p) - F(p,e) + F(e,p)$. *A* firing sequence *of* N *is a (nonempty) sequence* $M_0\,[\,e_1 > M_1 \ldots [\,e_n > M_n$. *The* $\mathbf{G_0}$ *language of the net is the set of labels* $e_1 \ldots e_n$ *of the firing sequences, plus the empty word* (ε). *A marking* M *is* reachable *if* $M = M_0$ *or* $M = M_n$ *for some firing sequence.* N *is* bounded *if the set of reachable markings is finite.*

In the sequel, $\mathcal{L}(N)$ denotes the $\mathbf{G_0}$ language of the net N. In the basic version of nets defined above, events do not bear extra labels; for this reason, languages $\mathcal{L}(N)$ are called *free* Petri net languages; thus $\mathcal{L}(N) \in \mathbf{G_0}^{\mathbf{f}}$ (where $\mathbf{G_0}$ means that all prefixes are included and $\mathbf{f}$ means that the labelling is free). In the sequel, we say that a language $\mathcal{L} \subseteq E^*$ is Parikh semilinear if its Parikh image $\psi\mathcal{L}$ is a semilinear set.

Theorem 7. *If a language* $\mathcal{L} \subseteq E^*$ *is Parikh semilinear, one can effectively compute from* $\psi\mathcal{L}$ *a* pure *Petri net* N *such that* $\mathcal{L} \subseteq \mathcal{L}(N)$ *and* $\mathcal{L} \subseteq \mathcal{L}(N') \Rightarrow \mathcal{L}(N) \subseteq \mathcal{L}(N')$ *for every pure Petri net* N'. *This assertion remains true if one replaces pure Petri nets with pure* and bounded *Petri nets.*
If all the e*-terminating sublanguages of a language* $\mathcal{L} \subseteq E^*$ *are Parikh semilinear, one can effectively compute from the respective sets* $\psi(\mathcal{L} \cap E^*e)$ *a Petri net* N *such that* $\mathcal{L} \subseteq \mathcal{L}(N)$ *and* $\mathcal{L} \subseteq \mathcal{L}(N') \Rightarrow \mathcal{L}(N) \subseteq \mathcal{L}(N')$ *for every Petri net* N'. *This assertion remains true if one replaces Petri nets with* bounded *Petri nets.*

The reader may find a brief presentation of the construction of N in [7], with enough indications for a complete proof of the above theorem (that extends Prop. 3.9 of [6]). If we now apply this theorem to HMSC languages, we get immediately the following.

Theorem 8. *Let* H *be a HMSC on* E. *One can effectively compute from* H *a general (resp. pure resp. bounded resp. pure and bounded) Petri net* N_H *with set of events* E *such that* $\mathcal{L}(N_H)$ *is the least language of a general (resp. bounded resp. pure resp. pure and bounded) Petri net* N *satisfying* $\mathcal{L}(H) \subseteq \mathcal{L}(N)$.

Proof. By lemma 6, the conditions of application of Theo. 7 are valid. □

The net N_H is not totally determined by the theory (several nets may have an identical language even though they have no redundant places), but also by the algorithm chosen for the construction. On the contrary, the language of the net N_H does not depend on the chosen algorithm. In the sequel, the language $\mathcal{L}(N_H)$ is denoted $\overline{\mathcal{L}}(H)$ and is called the Petri net *closure* of the HMSC language $\mathcal{L}(H)$. Every net N such that $\mathcal{L}(N) = \overline{\mathcal{L}}(H)$ is called a Petri net *realization* of the HMSC H or more properly of the HMSC language $\mathcal{L}(H)$. It is important to observe that all the words in $\overline{\mathcal{L}}(H)$ are admissible (this property may be enforced on the behaviours of a net by supplying for each message m one place p_m such that $F(!m, p_m)$ and $F(p_m, ?m)$). On the contrary, semilinearity is generally not preserved by the closure operation.

3.3 An undecidability result

At this stage, a question naturally arises: if one sticks to the strict requirement of equality of the specified and realized languages, does the subset of HMSCs which may be realized in the strict sense using Petri nets form a recursive subset? The answer is negative, and we produce hereafter evidence for this. In the meantime, let us recall an important result of Petri net theory due to Elisabeth Pelz [22].

Definition 11. *A* labeled *Petri net is a Petri net N equipped with a labeling map $\ell : E \to (A \cup \{\varepsilon\})$, where ε is the empty word. The labeled net is* deterministic *if at each reachable marking at most one event can be fired for each label. The labeled net is ε-free if $\ell(e) \neq \varepsilon$ for all $e \in E$. The language of the labeled net (N, ℓ) is the set of all images under ℓ of firing sequences from the initial marking. A Petri net* generator *is a labeled Petri net equipped with a finite subset of final markings (or partial markings) $\mathcal{F}$. The language of the generator is the set of all images under ℓ of firing sequences from the initial marking to final markings (or partial markings).*

Theorem 9 (Pelz). *The complement of the language of a deterministic ε-free labeled net N is the language of a Petri net generator $\mathcal{C}N$ constructible from N.*

Corollary 1. *Let N and N' be Petri net generators. If N' is deterministic and ε-free, one can decide on the inclusion $\mathcal{L}(N) \subseteq \mathcal{L}(N')$.*

Proof. $\mathcal{L}(N) \subseteq \mathcal{L}(N')$ if and only if the language of N'' is empty, where N'' is the Petri net generator defined as the synchronized product of N and $\mathcal{C}N'$, with events defined as pairs of events with common label, and with final (partial) markings defined as inverse projections of the final (partial) markings of $\mathcal{C}N'$. Deciding on the emptyness of $\mathcal{L}(N')$ reduces to the (partial) reachability problem for Petri nets, which is decidable [19]. □

The basic Petri nets introduced in Def. 10 are a particular case of deterministic ε-free Petri net generators: their labelling map $\ell : E \to (E \cup \{\varepsilon\})$ acts as the identity on E, and their set of final partial markings $\mathcal{F}$ has the totally undefined marking as its unique element. We proceed with the proof of the announced result.

Theorem 10. *Relation $\overline{\mathcal{L}}(H) = \mathcal{L}(H)$ is undecidable (from H).*

Proof. By lemma 7, from any rational subset $A \in Rat(X^* \times Y^*)$, one can construct a HMSC H on $E = E_1 \cup E_2$ (with $E_1 = X \cup \{\top_1\}$, $E_2 = Y \cup \{\top_2\}$, $\top_1 \neq \top_2$, and $(X \cup Y) \cap \{\top_1, \top_2\} = \emptyset$) such that $\mathcal{L}(H) = \delta^{-1} \circ Pref(A_\top)$ (with $\phi(e) = i$ for $e \in E_i$). It should be clear from this relation that $\mathcal{L}(H) \subseteq Pref(X^*\top_1) \sqcup\!\sqcup Pref(Y^*\top_2)$, where $\sqcup\!\sqcup$ is the shuffle operator, and that equality is met if and only if $A = X^* \times Y^*$. Since $Pref(X^*\top_1)$ and $Pref(Y^*\top_2)$ are languages of (one-place) nets N_1 and N_2 with disjoint sets of events, their shuffle is the language of the net N obtained by putting N_1 and N_2 side by side. Hence $\mathcal{L}(H) \subseteq \mathcal{L}(N)$, and $\mathcal{L}(H) = \mathcal{L}(N)$ if and only if $A = X^* \times Y^*$.

By Theo. 8, one can construct from H another Petri net N_H such that $\mathcal{L}(N_H) = \overline{\mathcal{L}}(H)$. As $\mathcal{L}(H) \subseteq \mathcal{L}(N)$, $\mathcal{L}(N_H) = \overline{\mathcal{L}}(H) \subseteq \mathcal{L}(N)$ by definition of Petri net closures of HMSC languages.

Suppose for contradiction that one can decide on the relation $\mathcal{L}(H) = \overline{\mathcal{L}}(H)$. We derive a decision procedure for the relation $A = X^* \times Y^*$, thus contradicting (iv) in Theo. 4. The procedure is as follows. If $\mathcal{L}(H) = \overline{\mathcal{L}}(H)$ has a negative answer (thus $\mathcal{L}(H) \subset \overline{\mathcal{L}}(H)$) then $\mathcal{L}(H) \subset \mathcal{L}(N)$ (since $\overline{\mathcal{L}}(H) \subseteq \mathcal{L}(N)$) and therefore $A \neq X^* \times Y^*$ (as $\mathcal{L}(H) = \mathcal{L}(N)$ iff $A = X^* \times Y^*$). If $\mathcal{L}(H) = \overline{\mathcal{L}}(H)$ has a positive answer (thus $\mathcal{L}(H) = \mathcal{L}(N_H)$) then $A = X^* \times Y^*$ if and only if $\mathcal{L}(N) \subseteq \mathcal{L}(N_H)$ (as $A = X^* \times Y^*$ iff $\mathcal{L}(H) = \mathcal{L}(N)$ iff $\mathcal{L}(N_H) = \mathcal{L}(N)$, and $\mathcal{L}(N_H) = \overline{\mathcal{L}}(H) \subseteq \mathcal{L}(N)$). By Theo. 9 and corollary 1, the last relation can be decided. Hence we have obtained a decision of the relation $A = X^* \times Y^*$. □

3.4 Distributed net realizations of HMSCs

Of special interest for the realization of HMSCs are the *distributable* Petri nets introduced in [4]. Let us recall the definition.

Definition 12. *A* distributable *Petri net system with set of* locations $[n]$ *is a quintuple* $\mathcal{N} = (P, E, F, M_0, \phi)$ *where* (P, E, F, M_0) *is a Petri net system and* $\phi : (P \cup E) \to [n]$ *is a* placement *map such that* $F(p, e) \neq 0 \Rightarrow \phi(p) = \phi(e)$ *for every place* $p \in P$ *and for every event* $e \in E$.

The range $[n]$ of the placement map represents the collection of sites on an asynchronous communication network where no message is ever lost or duplicated. Places and events located at different sites may be connected by the flow (multi) relation F. Hence an event $e \in E$ may produce tokens for a distant place $p \in P$. As the flow of tokens must be implemented on the network by asynchronous message passing, tokens produced will be available only after some delay, but this remains compatible with the asynchronous nature of Petri nets. On the contrary, if events $e \in E$ were allowed to consume tokens from distant places, one would immediately be faced with the problem of distributed conflict that cannot be solved without building first a synchronous layer on top of the asynchronous network. The condition $F(p, e) \neq 0 \Rightarrow \phi(p) = \phi(e)$ guarantees that conflicts cannot occur between events at different sites. A straightforward procedure for the implementation of distributable nets on asynchronous networks then follows. Let us postpone the description of this procedure and come back to HMSCs.

By definition, the set of events of a HMSC comes equipped with a placement map $\phi : E \to [n]$ (we recall that $\phi(e) = i$ if e is a private event owned by process i, or $e = !m$ and i is the emitter of m, or $e = ?m$ and i is the receiver of m). Thus, it makes sense to try realizing HMSCs with distributable Petri nets such that processes $i \in [n]$ are mapped identically to sites. Next theorem shows that this special form of the realization problem may be solved with little effort.

Theorem 11. *Theorem 7 extends to distributable Petri nets with fixed placement map* $\phi : E \to [n]$*. Theorem 8 extends similarly to distributable Petri nets with the placement map* $\phi : E \to [n]$ *inherited from* H*.*

So, given a HMSC H on E with placement map $\phi : E \to [n]$, one can compute a distributable Petri net $\mathcal{N}_H = (P, E, F, M_0, \phi)$ whose generated language is the closure of $\mathcal{L}(H)$ with respect to distributable Petri nets. In order to obtain a distributed realization of H, it remains to implement the distributable net $\mathcal{N}_H$ on the asynchronous network. To this effect, we propose a two stage procedure.

In a first stage, we expand $\mathcal{N}_H = (P, E, F, M_0, \phi)$ into a distributable net $\mathcal{N}'_H = (P', E', F', M'_0, \phi')$ where both new places and new events are added in order to model the buffered mode of transmission of tokens on the asynchronous network. The idea is to let $F'(e,p) = 0$ for all $e \in E$ and $p \in P$ such that $\phi(e) \neq \phi(p)$ (as an instantaneous transmission of tokens between different sites is not possible) and to compensate for the distorsion by introducing auxiliary message emissions and receptions (new events) which implement the asynchronous transmission of the tokens produced by e and passed to p from $\phi(e)$ to $\phi(p)$. The set of auxiliary messages $\mathcal{M}$ is the set of nonempty multisets μ on P such that $\mu = \mu(i,e,j)$ for some $i, j \in [n]$ and $e \in E$, letting:

$$\mu(i,e,j)(p) = F(e,p) \;\text{ if }\; i = \phi(e) \neq \phi(p) = j \;\text{ and }\; 0 \;\text{ otherwise.}$$

The sets $E_!$ (resp. $E_?$) of auxiliary message emissions (resp. receptions) are:

$$E_! = \{\, \mu^i_! \mid \exists e\, \exists j \;\, \mu = \mu(i,e,j) \neq \emptyset \} \;\text{ and }\; E_? = \{\, \mu_? \mid \exists i\, \exists e\, \exists j \;\, \mu = \mu(i,e,j) \neq \emptyset \}.$$

Auxiliary places are introduced in order to condition the emissions resp. receptions of auxiliary messages, giving the respective sets:

$$P_! = \{\, p^i_\mu \mid \exists e\, \exists j \;\, \mu = \mu(i,e,j) \neq \emptyset \} \;\text{ and }\; P_? = \{\, p_\mu \mid \exists i\, \exists e\, \exists j \;\, \mu = \mu(i,e,j) \neq \emptyset \}.$$

The sets of places and events of $\mathcal{N}'_H$ are $P' = P \cup P_! \cup P_?$ and $E' = E \cup E_! \cup E_?$. The initial marking M_0 is extended to M'_0 by setting $M'_0(p') = 0$ for all $p' \notin P$. The localisation map ϕ is extended to ϕ' by setting $\phi'(p^i_\mu) = \phi'(\mu^i_!) = i$ and $\phi'(p_\mu) = \phi'(\mu_?) = j$ where j is the (unique) location such that $\mu = \mu(i,e,j)$ for some i and e. The definition of $\mathcal{N}'_H$ is completed by setting the flow relations as follows ($e \in E$ and $p \in P$):
$F'(p,e) = F(p,e)$,
$F'(e,p) = F(e,p)$ if $\phi(e) = \phi(p)$ and 0 otherwise,
$F'(e,p^i_\mu) = 1$ if $\mu = \mu(i,e,j)$ for some j and 0 otherwise,
$F'(p^i_\mu, \mu^i_!) = F'(\mu^i_!, p_\mu) = F'(p_\mu, \mu_?) = 1$,
$F'(\mu_?, p) = \mu(p)$,
$F' = 0$ in all cases left unspecified.

It is proved in [2] that when all auxiliary events in $P' \setminus P$ are considered unobservable, the reachable state graph of $\mathcal{N}'_H$ is *divergence free*, which means that no infinite sequence of unobservable transitions can occur, and *branching bisimilar* to the reachable state graph of $\mathcal{N}_H$ (see [24]), which entails that the observable behaviours of the two nets are identical.

In a second stage, we remove from $\mathcal{N'}_H$ the auxiliary places p_μ which were used to represent tokens in transit on the network. The effect of the removal is to disconnect $\mathcal{N'}_H$ and to produce n component nets $\mathcal{N}_i$. For each $i \in [n]$, the net $\mathcal{N}_i$ is the restriction of $\mathcal{N'}_H$ on the (remaining) places and events with location i. One is left with implementing each $\mathcal{N}_i$ on the corresponding site i such that auxiliary events are interpreted as follows:
* each auxiliary event $\mu^i_!$ is interpreted as sending message μ to the (unique) destination j on the network such that $\mu = \mu(i, e, j)$,
* each auxiliary event $\mu_?$ in $\mathcal{N}_i$ is interpreted as receiving message μ from the network.

We obtain in this way a distributed (and provably correct) realization of the (distributable) Petri net closure $\overline{\mathcal{L}}(H)$ of the language $\mathcal{L}(H)$ of the HMSC H.

In case when $\mathcal{N}_H$ is a bounded Petri net, one can go a step further by translating the component nets $\mathcal{N}_i$ to *finite* automata A_i. As the component nets are generally unbounded, even though $\mathcal{N}_H$ is bounded, the translation is not immediate. The trick is to introduce for each place p of $\mathcal{N}_i$ with bound $\overline{p}$ in $\mathcal{N}_H$ (thus $p \in P$) a new place representing $\overline{p} - p$. The effect of the complementary places is to transform each $\mathcal{N}_i$ to a bounded net by pruning away behaviours of the autonomous net $\mathcal{N}_i$ that could not occur anyway in the context of $\mathcal{N'}_H$. Indeed, $\mathcal{N'}_H$ is bounded if $\mathcal{N}_H$ is bounded, and bounds agree on common places. The finite automata A_i are finally obtained by computing the reachable state graphs of the bounded versions of the components nets $\mathcal{N}_i$. We obtain in this way distributed realizations of HMSCs by finite automata communicating with asynchronous message passing.

3.5 Model-checking Petri net realizations of HMSCs

As $\overline{\mathcal{L}}(H)$ is by definition the closure of $\mathcal{L}(H)$, realized behaviours $\overline{\mathcal{L}}(H)$ may be larger than specified behaviours $\mathcal{L}(H)$. One may want to verify that extra behaviours cause no problems, which amounts to model-check Petri net realizations N_H of HMSCs against safety assertions. In view of the results recalled hereafter, this is certainly possible. Model-checking Petri nets w.r.t. the linear time μ-calculus is decidable [11]. More generally, one can decide on the inclusion $\mathcal{L}(N) \subseteq A$ for a net N labelled on E and $A \in Rat(E^*)$ [17]. Last but not least, by Theo. 9, one can decide on the inclusion $\mathcal{L}(N) \subseteq \mathcal{L}(N')$ for two nets N and N' labelled on E provided that N' is deterministic.

The decision techniques supplied in the above references apply to arbitrary Petri nets N. In the particular case where $N = N_H$ is the net realization of a HMSC, one may try to exploit this specific fact. For instance, let H and N have the same set of events, then $\mathcal{L}(N_H) \subseteq \mathcal{L}(N)$ if and only if every place p of N coincides with a *region* of $\mathcal{L}(H)$ (see [7]). This may be checked directly and efficiently from the automaton $\psi(\overline{H})$ which was constructed in section 3.1. One may decide in a similar way on the equivalent inclusions $\mathcal{L}(H_1) \subseteq \overline{\mathcal{L}}(H_2)$ or $\overline{\mathcal{L}}(H_1) \subseteq \overline{\mathcal{L}}(H_2)$ for two different HMSCs. It is not clear that the use of regions may help to decide more efficiently on the inclusion $\overline{\mathcal{L}}(H) \subseteq A$ for $A \in Rat(E^*)$.

4 Case studies

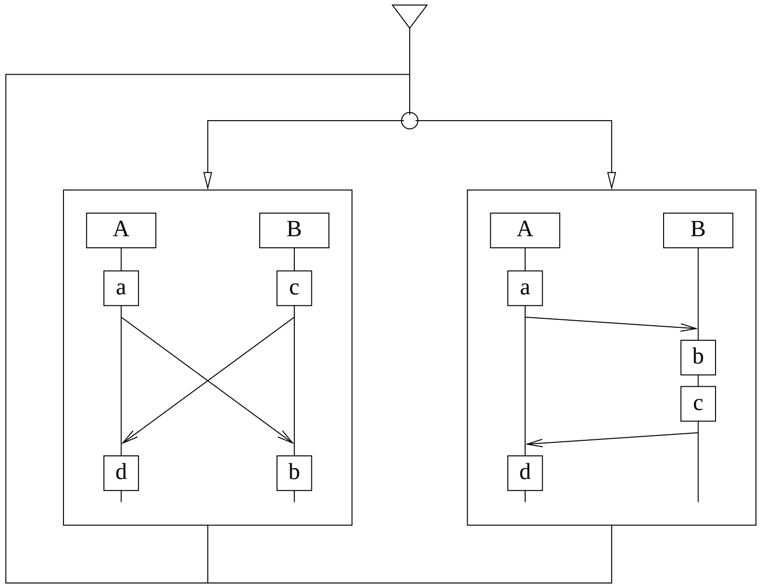

Fig. 1. A HMSC with non-local choice

Two case studies taken from telecommunication applications are now presented. They embrace typical behaviours of communication protocols. Both consist in synthesizing communicating finite state machines (CFSM) from a given HMSC (H). We apply for this purpose the method sketched in 3.4 and explained in [2]: i) synthesize a distributable (bounded) Petri net, where places and transitions have locations and transitions take tokens only from co-located places, and ii) transform the Petri net into a set of asynchronously communicating automata. However, we make here an advanced use of Petri net synthesis: we compute the Petri net closure of the language $\mathcal{L}(H)$ *projected* on the subset of private events P of E $(= P \cup S \cup R)$. In view of Theo. 7, this makes sense since the projection of a language on a sub-alphabet preserves the semi-linearity of its commutative image. Messages in a bMSC are then considered only as a means to specify the partial order on events in P; therefore we don't attribute labels to message emissions (in S) and receptions (in E). The goal of reconstructing a dialogue between agents which enforces the specified partial order on P is achieved in step (ii), where auxiliary messages are used in any case for achieving distribution.

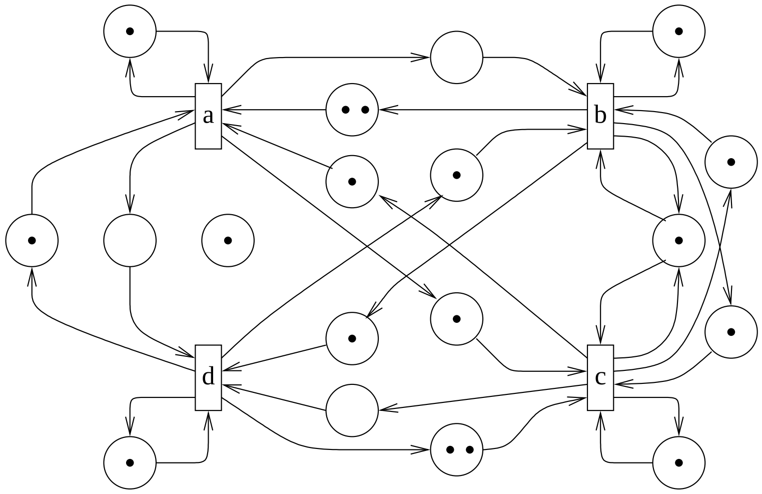

Fig. 2. Synthesized Petri net

4.1 A HMSC with non-local choice

The HMSC H in figure 1 defines two instances of agents (A and B) repeatedly performing four local actions a, b, c and d which are ordered thanks to a pair of messages. Two cases can occur: either (right-hand side case) all actions are performed in sequence, or (left-hand side case) some events can be performed concurrently. The commutative image of $\mathcal{L}(H)$ is given by the regular expression:

$$(abcd)^* \left(0 + a + c + ab + ac + abc + acd + abc^2 + a^2cd\right)$$

More precisely, for $x \in \{a, b, c, d\}$, the commutative images of the right residuals $\mathcal{L}(H)/x$ are given in this order by the respective regular expressions:

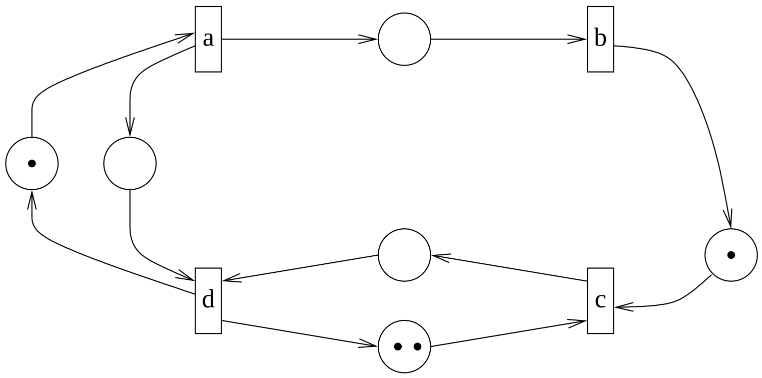

Fig. 3. Minimal Petri net

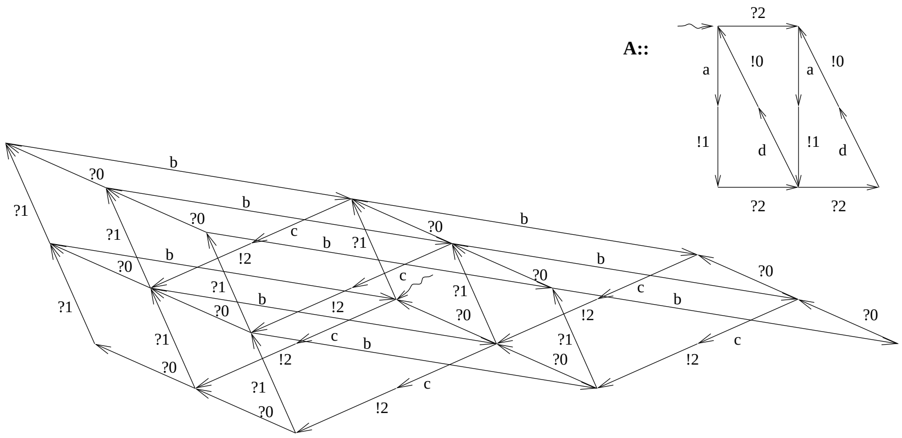

Fig. 4. Communicating finite state machines

$(abcd)^* \, (0 + c + acd)$
$(abcd)^* \, \big(a + ac + acd + a^2cd\big)$
$(abcd)^* \, (0 + a + ab + abc)$
$(abcd)^* \, (ac + abc)$

Hence each place $(m_0,\, {}^\bullet a,\, a^\bullet,\, {}^\bullet b,\, b^\bullet,\, {}^\bullet c,\, c^\bullet,\, {}^\bullet d,\, d^\bullet)$ marked with m_0 tokens in the net $N = \mathcal{N}_H$ which realizes the (bounded) Petri net closure $\overline{\mathcal{L}}(H)$ of $\mathcal{L}(H)$ should satisfy the following set of linear equations and inequations:

$$\begin{cases}
a^\bullet - {}^\bullet a + b^\bullet - {}^\bullet b + c^\bullet - {}^\bullet c + d^\bullet - d^\bullet = 0 & \textit{places are bounded} \\
m_0 \geq {}^\bullet a & \textit{Firing of } \mathrm{a} \\
m_0 + c^\bullet - {}^\bullet c \geq {}^\bullet a & \\
m_0 + a^\bullet - {}^\bullet a + c^\bullet - {}^\bullet c + d^\bullet - {}^\bullet d \geq {}^\bullet a & \\
m_0 + a^\bullet - {}^\bullet a \geq {}^\bullet b & \textit{Firing of } \mathrm{b} \\
m_0 + a^\bullet - {}^\bullet a + c^\bullet - {}^\bullet c \geq {}^\bullet b & \\
m_0 + a^\bullet - {}^\bullet a + c^\bullet - {}^\bullet c + d^\bullet - {}^\bullet d \geq {}^\bullet b & \\
m_0 + 2a^\bullet - 2{}^\bullet a + c^\bullet - {}^\bullet c + d^\bullet - {}^\bullet d \geq {}^\bullet b & \\
m_0 \geq {}^\bullet c & \textit{Firing of } \mathrm{c} \\
m_0 + a^\bullet - {}^\bullet a \geq {}^\bullet c & \\
m_0 + a^\bullet - {}^\bullet a + b^\bullet - {}^\bullet b \geq {}^\bullet c & \\
m_0 + a^\bullet - {}^\bullet a + b^\bullet - {}^\bullet b + c^\bullet - {}^\bullet c \geq {}^\bullet c & \\
m_0 + a^\bullet - {}^\bullet a + c^\bullet - {}^\bullet c \geq {}^\bullet d & \textit{Firing of } \mathrm{d} \\
m_0 + a^\bullet - {}^\bullet a + b^\bullet - {}^\bullet b + c^\bullet - {}^\bullet c \geq {}^\bullet d &
\end{cases}$$

Of course, all nine variables $(m_0,\, {}^\bullet a,\, a^\bullet,\, {}^\bullet b,\, b^\bullet,\, {}^\bullet c,\, c^\bullet,\, {}^\bullet d,\, d^\bullet)$ must be greater than or equal to zero. In addition to these linear constraints, one among two alternative sets of equations should be imposed in order to ensure that places are distributed according to the mapping of events : places located on A should satisfy ${}^\bullet b = {}^\bullet c = 0$, places located on B should statisfy ${}^\bullet a = {}^\bullet d = 0$.

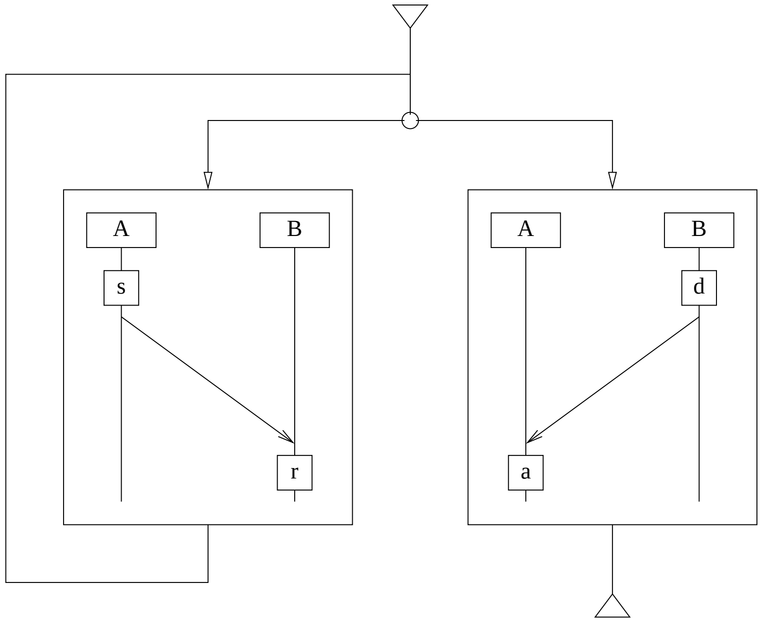

Fig. 5. An non-regular HMSC: the simple asynchronous transfer protocol

This defines two polyhedral cones. Their extremal rays are the places of the distributable (bounded) Petri net N in figure 2. It is easy to check that $\mathcal{L}(H)$ (which turns out to be regular) is equal to $\mathcal{L}(N)$. The Petri net N contains redundant places which can be safely discarded without changing its language. This results in the net in figure 3, which can be transformed into the pair of communicating automata in figure 4 (emissions resp. receptions of auxiliary messages are represented by occurences $!i$ resp. $?i$).

4.2 A non-regular HMSC language

The second case study is about the simple asynchronous transfer protocol defined by the HMSC H in figure 5. Action s stands for a message sending, r for a receiving, d for a disconnection request, and a for a disconnection acknowledgement. The commutative image of $\mathcal{L}(H)$ is given by the expression $(sr)^*(s^* + d + da)$. Each (possibly unbounded) place (m_0, ${}^\bullet a$, $a^\bullet$, ${}^\bullet b$, $b^\bullet$, ${}^\bullet c$, $c^\bullet$, ${}^\bullet d$, $d^\bullet$) of the net $N = \mathcal{N}_H$ which realizes the (general) Petri net closure $\overline{\mathcal{L}}(H)$ of $\mathcal{L}(H)$ should satisfy the following set of linear equations and inequations:

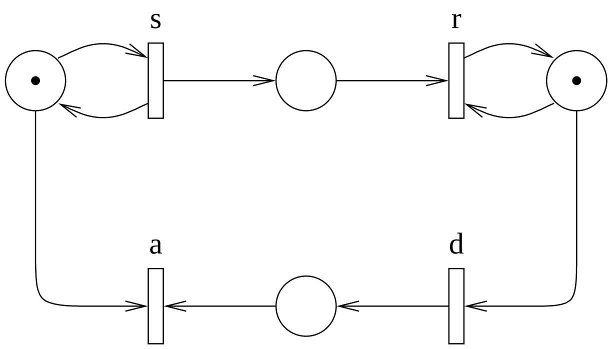

Fig. 6. Synthesized Petri net without redundant places

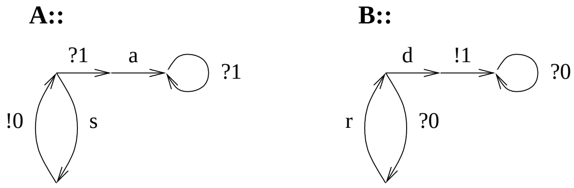

Fig. 7. Communicating automata

$$\begin{cases} m_0 \geq 0 & \textit{variables are non-negative} \\ {}^\bullet s \geq 0, s^\bullet \geq 0, {}^\bullet r \geq 0, r^\bullet \geq 0 & \\ {}^\bullet d \geq 0, d^\bullet \geq 0, {}^\bullet a \geq 0, a^\bullet \geq 0 & \\ s^\bullet - {}^\bullet s \geq 0 & \textit{inequations on cycles} \\ s^\bullet - {}^\bullet s + r^\bullet - {}^\bullet r \geq 0 & \\ m_0 \geq {}^\bullet s & \textit{Firing of } \mathrm{s} \\ m_0 + s^\bullet - {}^\bullet s \geq {}^\bullet r & \textit{Firing of } \mathrm{r} \\ m_0 \geq {}^\bullet d & \textit{Firing of } \mathrm{d} \\ m_0 + d^\bullet - {}^\bullet d \geq {}^\bullet a & \textit{Firing of } \mathrm{a} \end{cases}$$

Distribution constraints impose either ${}^\bullet r = {}^\bullet d = 0$ or ${}^\bullet s = {}^\bullet a = 0$. This defines two polyhedral cones. Their extremal rays are: $d^\bullet + {}^\bullet a$, $m_0 + {}^\bullet a$, $m_0 + s^\bullet + {}^\bullet s + {}^\bullet a$, $m_0 + s^\bullet + {}^\bullet s$, m_0, $s^\bullet$, $r^\bullet$, $d^\bullet$, $a^\bullet$, $m_0 + r^\bullet + {}^\bullet r$, $m_0 + r^\bullet + {}^\bullet r + {}^\bullet d$, $s^\bullet + {}^\bullet r$, $m_0 + {}^\bullet d$. Some rays define redundant places and can be safely discarded. The synthesized Petri net is shown in figure 6. This net may be transformed into the pair of communicating automata in figure 7 ($!i$ resp. $?i$ represent emissions resp. receptions of auxiliary messages). These communicating automata have been derived under the assumption that the communication medium consists in a pair of unbounded reliable channels. The place between s and r in the Petri net is unbounded, and since the communication medium acts as a memory, the communication channel from agent A to agent B is also unbounded.

5 Conclusion

The results which have been presented in this paper call for several comments.

First of all, the undecidability results shown in section 2 indicate that one has an alternative between two antagonistic views upon HMSCs:
1) One may regard HMSCs as complete specifications. In that case, one is led to only consider HMSCs with regular behaviour, as otherwise verification and realization would become unfeasible. The loss of expressiveness is in our opinion a serious drawback at an early stage in the design of telecommunication systems.
2) One may regard HMSCs as incomplete specifications. One must in that case define the meanings of HMSCs as closures of their normal behaviours with respect to such or such class of realizations, e.g. distributable Petri nets or communicating automata. Verification on general HMSCs' behaviours is unfeasible, but verification on their closures is effective for well chosen classes of realizations – Petri nets for instance. We feel that this pragmatic view is far more suitable.

High level message sequence charts express positive facts on system behaviour. Results shown in section 3 about verification on Petri net realizations of HMSCs suggest that it would be desirable to extend HMSCs so that both positive and negative facts could be expressed in a single formalism. This should not be confused with Harel's distinction between compulsory and optional events in HMSCs.

Finally, this paper only considered closures of HMSCs' behaviours with respect to Petri nets. Considering other classes and comparing their advantages is an open field for research. Incidentally, the Petri net closure of a HMSC shows an interesting property: the flow of tokens in the synthesized net seems to induce a minimal covering of the order on events in the HMSC. This could provide a way of minimizing communications while preserving behaviour.

References

1. Alur, R., Yannakakis, M.: Model Checking of Message Sequence Charts. Proc. Concur, LNCS **1664** (1999) 114–129
2. Badouel, E., Caillaud, B., Darondeau, Ph.: Distributing Finite Automata through Petri Net Synthesis. (draft available from the authors)
3. Berstel, J.: Transductions and Context-Free Languages. Teubner Studienbücher, Stuttgart (1979)
4. Caillaud, B.: Bounded Petri Net Synthesis Techniques and their Applications to the Distribution of Reactive Automata. JESA **9–10** no.33 (1999) 925–942
5. Damm, W., Harel, D.: LCSs: Breathing Life into Message Sequence Charts. Report CS98/09, Weizmann Institute of Technology (1998)
6. Darondeau, Ph.: Deriving Unbounded Petri Nets from Formal Languages. Proc. Concur, LNCS **1466** (1998) 533–548
7. Darondeau, Ph.: Region Based Synthesis of P/T-Nets and its Potential Applications. Proc. ICATPN, LNCS **1825** (2000) 16–23

8. Diekert, V., Métivier, Y.: Partial Commutation and Traces. Research report 1996/02, Universität Stuttgart Fakultät Informatik (1996)
9. Diekert, V., Rozenberg, G. (editors): The Book of Traces. World Scientific, Singapore (1995)
10. Eilenberg, S., Schützenberger, M.: Rational Sets in Commutative Monoids. Journal of Algebra **13** (1969) 173–191
11. Esparza, J.: On the Decidability of Model-checking for several mu-calculi and Petri Nets. Proc. Caap, LNCS **787** (1994) 115–129
12. Fischer, P.C., Rosenberg, A.L.: Multitape One-Way Nonwriting Automata. JCSS **2** (1968) 88–101
13. Harel, D., Kugler, H.: Synthesizing State-Based Object Systems from LSC Specifications. Report MCS99/20, Weizmann Institute of Technology (1999)
14. Henriksen, J.G., Mukund, M., Narayan Kumar, K., Thiagarajan, P.S.: On Message Sequence Graphs and Finitely Generated Regular MSC Languages. Proc. Icalp, LNCS **1853** (2000) 675–686
15. Henriksen, J.G., Mukund, M., Narayan Kumar, K., Thiagarajan, P.S.: Regular Collections of Message Sequence Charts. Proc. MFCS, LNCS **1893** (2000) 405–414
16. TU-TS Recommendation Z.120: Message Sequence Chart 1996 (MSC96). Technical Report, ITU-TS, Geneva (1996)
17. Jancar, P., Moeller, F.: Checking Regular Properties of Petri Nets, Proc. Concur, LNCS **962** (1995) 348–362
18. Mauw, S., Reniers, M.A., High-Level Message Sequence Charts. Proc. Eighth SDL Forum, Elsevier Science Publishers B.V. (1997) 291–306
19. Mayr, E.: An Algorithm for the General Petri Net Reachability Problem. SIAM Journal on Computing **13** (1984) 441–460
20. Mukund, M., Narayan Kumar, K., Sohoni, M.: Synthesizing distributed finite-state systems from MSCs. Proc. Concur, LNCS **1877** (2000) 521–535
21. Muscholl, A., Peled, D., Su, Z.: Deciding Properties for Message Sequence Charts. Proc. Fossacs, LNCS **1378** (1998) 226–242
22. Pelz, E.: Closure Properties of Deterministic Petri Nets. Proc. Stacs, LNCS **247** (1987) 373–382
23. Peterson, J.L.: Computation Sequence Sets. JCSS **13** (1976) 1–24
24. van Glabbeek, R.J., Weijland, W.P.: Branching Time and Abstraction in Bisimulation Semantics. Proc. IFIP Congress, North Holland / IFIP (1989) 613–618