

On the synthesis of industry level process models from enterprise level process models

Pieter Kwantes and Jetty Kleijn

LIACS, Leiden University, P.O.Box 9512
NL-2300 RA Leiden, The Netherlands
{p.m.kwantes,h.c.m.kleijn}@liacs.leidenuniv.nl

Abstract. In this paper we provide a framework to specify the synthesis of a comprehensive (industry level) process model from a set of lower level (enterprise) process models. This framework allows us to verify that the resulting model is compliant with a given blueprint (reference model), using local checks only. The reference model describes the intended interactions between the enterprise processes in accordance with industry standards. We propose an approach that allows to check compliance locally per enterprise without requiring information from other enterprise models. This implies in particular that there is no need for individual enterprises to disclose internal design information to the public domain. In our general set-up, we use Petri nets as a modelling framework for our process models and labelled transition systems with silent actions as their semantics. Branching bisimilarity of their transition systems is here the criterion for compliance of process models.

Keywords: business process modelling, Petri net, industry and enterprise level, synthesis, verification, (partitioned) labelled transition system, qualified branching bimilarity

1 Introduction

Industry level standardisation is a prerequisite for interoperability between business processes of the enterprises in an industry.¹ Typical areas eligible for standardisation include *communication*, eg., the syntax of messages exchanged between business processes, and the *flow of control* of this communication, i.e., who sends which messages in which order to whom.²

¹ We use the term “industry” here rather loosely to include, eg., any group of enterprises involved in the supply chain for a particular good or service.

² Industry level institutions for message syntax standardisation are quite common, eg., SWIFT for Financial services [23], HL7 for Health care [14], RosettaNet for the electronics industry [12], EDSN for the Dutch energy market [7] to name just a few.

Interoperability between business processes is therefore dependent on the compliance of these business processes with industry standards. The flow of control at the industry level emerges from the interaction of business processes at the enterprise level. However, the design (and verification) of business processes at the enterprise level is typically the exclusive domain of the enterprise in question. To align the design of (enterprise level) business processes with the required behaviour emerging at the industry level, poses a significant and costly coordination problem for the enterprises involved.

As an example, consider the settlement of a transaction on the secondary capital market (eg., buying or selling of a security). This is an industry level process that emerges from the interaction of several business processes at the enterprise level like those of *Investors*, *Brokers*, *Counterparties*, *Custodians* and *Banks*, but there could be (many) more depending on the nature of the transaction. Such interaction typically involves the bilateral asynchronous exchange of standardized messages between these enterprise level processes³. This is where the coordination problem arises. First of all because there is no central authority to ensure that (enterprise level) implementation decisions are compliant with the requirements, i.e., the intended flow of control, at the industry level. Moreover the specifications of these requirements are often informal and/or incomplete. Any approach aiming to solve this problem should take into account that, as a rule, implementation decisions are taken locally, at the enterprise level. We cannot, in general, assume the presence of an entity with the authority to align these local decisions with global requirements⁴

In this paper we demonstrate how this problem could be solved by outlining an approach to establish an enterprise's (local) compliance with a (global) industry level reference model on basis of local checks. Petri nets provide the modelling framework for our process models with labelled transition systems with silent actions as their semantics. We introduce *Enterprise nets* (E-nets) that can exchange messages with other E-nets, to represent enterprise process models and *Industry nets* (I-nets), composed of communicating E-nets, to represent process models at the industry level. A publicly available I-net reflecting the intended flow of control of communication for a particular industry level process, serves as a reference model for enterprises participating in that process, but is not concerned with how enterprises organise themselves internally. This reference model can be considered as a (minimum) set of operational requirements negotiated at the

³ eg., XML messages, meeting the ISO20022 standard, exchanged over the SWIFT-network.

⁴ Although in many cases industry level institutions, like the ones mentioned earlier, have emerged to facilitate the negotiation of certain industry level standards.

industry level. The implementation of the reference model, i.e., the synthesis of implementations provided by the individual enterprises, should be compliant with these requirements.

Compliance relates to communicating behaviour. We use labelled transition systems with silent actions to represent the behaviour of E-nets and I-nets, and branching bisimilarity to define compliance. Silent actions represent non-communicating behaviour. *Bisimilarity* is a well established approach for relating behaviours in this set up [1, 22]. *Branching bisimilarity* is the finest bisimilarity relation that ignores differences in non-communicating behaviour while respecting its branching structure [8, 9]. This appears to be a good point to start our investigation into the notion of compliance, which is defined in more detail later on. To reflect their modular structure, we introduce labelled transition systems with a partition of their labels, as an additional semantic model for I-nets. This leads to a notion of *qualified branching bisimilarity*, suitable for such transition systems. This auxiliary notion facilitates the proof that — indeed — compliance can be verified locally. Consequently, we arrive at the main result that, when combining enterprise models into an industry level process model that should be compliant with a given blueprint, local compliance checks suffice.

Our approach is an original contribution to existing approaches towards local verification of compliant global behaviour. First of all, it includes a modelling framework with a strong built in separation of concerns between global (domain specific) requirements laid down in an (industry level) reference model and local implementation decisions left to enterprises. Secondly, many existing approaches appear to be mainly concerned with the verification of general correctness properties (like deadlock freeness, soundness, or accordance) while here we are particularly concerned with the verification of domain specific requirements as laid down in the global reference model.

The paper is organised as follows. In Sections 2 and 3, we provide a precise definition of E-nets and I-nets and their implementations. In Section 4 compliance is defined and we formulate the main theorem that global compliance can be verified locally. In Section 5 we introduce qualified branching bisimilarity for partitioned labelled transition systems. In Section 6 an outline is given for the proof of our main result as presented in Section 4. Finally, related work is discussed in Section 7, while Section 8 forms the conclusion of the paper.

2 Enterprise nets and Industry nets

$\mathbb{N} = \{0, 1, 2, \dots\}$ is the set of natural numbers. Given a natural number $n \geq 1$, we denote the set $\{j \in \mathbb{N} \mid 1 \leq j \leq n\}$ by $[n]$. Let L be a set. Then L^* is the set consisting of all finite sequences $w = a_1 \cdots a_m$ of elements

$a_i \in L$ for $i \in [m]$. If $m = 0$, then w is the empty sequence, denoted by λ . If $a_i = a$ for all $i \in [m]$, then we may write $w = a^m$. In particular, for all $a \in L$, we have that a^0 is the empty sequence.

A Petri net is a triple $N = (P, T, F)$, where P is a finite set of *places*, T is a finite set of *transitions* such that $(P \cap T) = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation). We refer to the elements of $P \cup T$ as the *nodes* of N . If N' is a Petri net such that N and N' have disjoint sets of nodes then N and N' are called *disjoint*.

Let $N = (P, T, F)$ be a Petri net. Then $\bullet x = \{y \mid (y, x) \in F\}$ and $x^\bullet = \{y \mid (x, y) \in F\}$ are the *preset* and the *postset* respectively, of the node $x \in P \cup T$. A place $p \in P$ such that $\bullet p = \emptyset$ is a *source place* of N .

A *marking* of N is a function $\mu : P \rightarrow \mathbb{N}$. Let $t \in T$ and μ a marking of N . Then t is *enabled at* μ if $\mu(p) > 0$ for all $p \in \bullet t$. If t is enabled at μ , it may *occur*, leading to a new marking μ' . This we write as $\mu \xrightarrow{t} \mu'$ with $\mu'(p) = \mu(p) - 1$ whenever $p \in \bullet t \setminus t^\bullet$; $\mu'(p) = \mu(p) + 1$ whenever $p \in t^\bullet \setminus \bullet t$; and $\mu'(p) = \mu(p)$ otherwise. We extend the notation $\mu \xrightarrow{t} \mu'$ to sequences $w \in T^*$ as follows: $\mu \xrightarrow{\lambda} \mu$ for all μ and $\mu \xrightarrow{wt} \mu'$ for markings μ, μ' of N , $w \in T^*$ and $t \in T$, whenever there is a marking μ'' such that $\mu \xrightarrow{w} \mu''$ and $\mu'' \xrightarrow{t} \mu'$. If $\mu \xrightarrow{w} \mu'$, for some $w \in T^*$, we say that μ' is *reachable* from μ . $[\mu]$ is the set of markings reachable from μ . The marking μ of N such that, for all $p \in P$, $\mu(p) = 1$ if p is a source place and $\mu(p) = 0$ otherwise, is the *default initial marking* of N .

Let \mathcal{M} be the set of *message types* fixed throughout this paper. Enterprise nets, to be defined next, are Petri nets that can communicate (asynchronously) through the occurrence of output and input transitions with matching message types. They have a single source place, resembling the notion of *workflow-nets* in that respect, but without a designated final marking. Composition of these lower level nets results in a higher level net with multiple source places. We are particularly interested in the global communicating behaviour that emerges from bilateral interactions between local processes.

Definition 1 (Enterprise net). *An Enterprise net, E-net for short, is a tuple $\mathcal{E} = (P, [T_{int}, T_{inp}, T_{out}], F, M)$ such that T_{int}, T_{inp} , and T_{out} are mutually disjoint sets; T_{int} is the set of internal transitions of \mathcal{E} , T_{inp} its set of input transitions, and T_{out} its set of output transitions; moreover, $(P, T_{int} \cup T_{inp} \cup T_{out}, F)$, the underlying Petri net of \mathcal{E} , is a Petri net with exactly one source place; finally $M : T_{inp} \cup T_{out} \rightarrow \mathcal{M}$ is the communication function of \mathcal{E} .*

The triple (T_{inp}, T_{out}, M) is the interface of \mathcal{E} .

We assume that the E-nets that constitute an Industry net (defined below) are disjoint, i.e., their underlying Petri nets are disjoint. Moreover the input and output transitions of these nets can be matched.

Definition 2 (Matching). *Let $n \geq 2$. Let $V = \{\mathcal{E}_i \mid i \in [n]\}$ be a set of pairwise disjoint E-nets. Let, for each $i \in [n]$, $T_{out,i}$ be the set of output transitions of \mathcal{E}_i , $T_{inp,i}$ its set of input transitions, and M_i its communication function. A matching over V is a bijection $\varphi : \bigcup_{i \in [n]} T_{out,i} \rightarrow \bigcup_{j \in [n]} T_{inp,j}$ such that whenever $t \in T_{out,i}$ and $\varphi(t) \in T_{inp,j}$, for some i, j , then $i \neq j$ and $M_i(t) = M_j(\varphi(t))$.*

A set V of mutually disjoint E-nets is said to be *composable* if there exists a matching over V . Note that not every such V is composable; and if it is, there may exist more than one matching over this set.

Given a composable set V as specified in Definition 2, and a matching φ over V , the nets in V are connected through (new) *channel places* from the set $P(V, \varphi) = \{[t, \varphi(t)] \mid t \in T_{out,i}, i \in [n]\}$ using the *channel arcs* from $F(V, \varphi) = \{(t, [t, \varphi(t)]) \mid t \in T_{out,i}, i \in [n]\} \cup \{([t, \varphi(t)], \varphi(t)) \mid t \in T_{out,i}, i \in [n]\}$.

Definition 3 (Industry net). *Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of disjoint E-nets with $\mathcal{E}_i = (P_i, [T_{int,i}, T_{inp,i}, T_{out,i}], F_i, M_i)$ for all $i \in [n]$, and let φ be a matching over V . The Industry net, or I-net for short, over (V, φ) , denoted $\mathcal{I}(V, \varphi)$, is the Petri net (P, T, F) with $P = \bigcup_{i \in [n]} P_i \cup P(V, \varphi)$, $T = \bigcup_{i \in [n]} T_i$, and $F = \bigcup_{i \in [n]} F_i \cup F(V, \varphi)$. The Petri net $(P(V, \varphi), T \setminus \bigcup_{i \in [n]} T_{int,i}, F(V, \varphi))$ is the communication subnet of $\mathcal{I}(V, \varphi)$.*

Note that I-nets are (communication-)closed systems, in the sense that all transitions that occur as an output (input) transition in one of the E-nets are connected through a channel place with a matching input (output, respectively) transition of another E-net.

3 Implementations

We assume the existence of a publicly available I-net reflecting the intended flow of control of communication for a particular industry level process and serving as a reference model for the enterprises that will participate in that process. The idea is that each of these enterprises provides a local implementation of one of the E-nets in the reference model. The aim of the reference model however is not concerned with how they are organised internally. It only prescribes the communication requirements at a global

(industry) level and the local implementations (E-nets themselves) should obey these requirements. To ensure that all implementations “fit together” (match) as prescribed by the reference model, each should have the same interface as the E-net it implements. We assume moreover that the E-nets of different enterprises are always disjoint (eg., by proper indexing).

Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of E-nets and let φ be a matching over V . Let $\mathcal{I}(V, \varphi)$ be a reference model.

Definition 4 (Implementation). *Let $\mathcal{E} \in V$. An E-net \mathcal{E}' is an implementation of \mathcal{E} with respect to V if \mathcal{E}' is disjoint with all E-nets in $V \setminus \{\mathcal{E}\}$ and \mathcal{E} and \mathcal{E}' have the same interface.*

Note that if \mathcal{E}' is an implementation of \mathcal{E} , then $W = (V \setminus \{\mathcal{E}\}) \cup \{\mathcal{E}'\}$ is composable and φ is a matching over W . Thus $\mathcal{I}(W, \varphi)$ is an I-net.

Let $V' = \{\mathcal{E}'_i : i \in [n]\}$ be a set of E-nets such that, for all $i \in [n]$, \mathcal{E}'_i is an implementation of $\mathcal{E}_i \in V$. Since, by our assumption above, the \mathcal{E}'_i are pairwise disjoint and because φ is a matching over V' , it follows that also $\mathcal{I}(V', \varphi)$ is a well-defined Industry net. It is an (full) implementation of $\mathcal{I}(V, \varphi)$. Note that V' can inherit the matching ϕ from V , because by Definition 4 their E-nets have the same interface.

4 Compliance

To define compliance, a notion of behaviour is needed. We use labelled transition systems to represent behaviour and branching bisimilarity as a measure of behavioural equivalence.

A *labelled transition system*, *LTS* for short, is a tuple $\mathcal{T} = (Q, L, \delta)$, where Q is a set of *states*, L is a set of (*action*) *labels*, and $\delta \subseteq Q \times L \times Q$ is a *transition relation*. Rather than $(q, a, q') \in \delta$, where $q, q' \in Q$ and $a \in L$, we may also write $q \xrightarrow{a} q'$.

A *path* in \mathcal{T} from $q \in Q$ to $q' \in Q$ is a finite sequence $(q_1, a_1, q_2), (q_2, a_2, q_3), \dots, (q_m, a_m, q_{m+1})$, where $m \geq 0$, $q = q_1$, $q' = q_{m+1}$, and $(q_i, a_i, q_{i+1}) \in \delta$, for all $i \in [m]$. Such path may also be denoted as $(q_1, a_1 \dots a_m, q_{m+1})$ or as $q_1 \xrightarrow{a_1 \dots a_m} q_{m+1}$. If $m = 0$ the path is *empty* and $q = q'$. Let L' be a set of labels and $l : L \rightarrow L'$ a function. Then $l(\delta) = \{(q, l(a), q') : (q, a, q') \in \delta\}$, $l(L) = \{l(a) \mid a \in L\}$ and $l(\mathcal{T}) = (Q, l(L), l(\delta))$.

In our process model we will hide internal actions and therefore we need the concept of a *silent action*. For technical reasons we introduce *LTSs* with more than one silent action. An *LTS with silent actions* is a tuple $\mathcal{T} = (Q, L, S, \delta)$ where $S \subseteq L$ is a set of specified silent actions. All notions and notations introduced for labelled transition systems carry over to *LTSs* with silent actions in the obvious way. Note that every *LTS* $\mathcal{T} = (Q, L, \delta)$

can be interpreted as an *LTS* with an empty set of silent actions $(Q, L, \emptyset, \delta)$. Moreover, our labelled transition systems will often be equipped with a designated initial state. Thus, an *initialized LTS (with silent actions)* is a tuple $\mathcal{T} = (Q, L, \delta, q_{in})$ (or $\mathcal{T} = (Q, L, S, \delta, q_{in})$, respectively) such that $\mathcal{T} = (Q, L, \delta)$ is an *LTS* (or (Q, L, S, δ) is an *LTS* with silent actions, respectively) where $q_{in} \in Q$ is a designated initial state.

We are now ready to introduce the labelled transition systems associated with an Industry net.

Definition 5 (Transition System of an I-net). *Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of E-nets, with $\mathcal{E}_i = (P_i, [T_{int,i}, T_{inp,i}, T_{out,i}], F_i, M_i)$ for all $i \in [n]$, φ a matching over V and μ_{in} the default initial marking of $\mathcal{I}(V, \varphi) = (P, T, F)$.*

– $\mathcal{LTS}(V, \varphi) = ([\mu_{in}], T, \emptyset, \delta, \mu_{in})$, where $\delta = \{(\mu, t, \mu') \mid \mu \in [\mu_{in}], t \in T, \mu \xrightarrow{t} \mu' \text{ in } \mathcal{I}(V, \varphi)\}$, is the initialized labelled transition system of $\mathcal{I}(V, \varphi)$.

– Let $\lambda_{V,\tau} : T \rightarrow T \cup \{\tau\}$ be defined as $\lambda_{V,\tau}(t) = \begin{cases} \tau & \text{if } t \in \bigcup_{i \in [n]} T_{int,i} \\ t & \text{otherwise} \end{cases}$

Then $\lambda_{V,\tau}(\mathcal{LTS}(V, \varphi)) = ([\mu_{in}], \lambda_{V,\tau}(T), \{\tau\}, \lambda_{V,\tau}(\delta), \mu_{in})$, is the initialized labelled transition system with silent action τ of $\mathcal{I}(V, \varphi)$.

The labelled transition system associated with the Industry net $\mathcal{I}(V, \varphi)$ is its marking graph with the default initial marking as initial state. Note that since each Enterprise net has exactly one source place, the Industry net has n source places, all initially marked with one token. In $\lambda_{V,\tau}(\mathcal{LTS}(V, \varphi))$, the labels of the internal transitions of the Enterprise nets constituting V , are made silent using the special symbol τ .

Branching bisimilarity as a criterion for behavioural equivalence based on labelled transition systems with a single silent action was introduced in [9]. A detailed proof that branching bisimilarity is an equivalence relation is provided in [5]. The definitions of branching bisimilarity provided in both papers are slightly different, but as explained there, they are equivalent. For the purpose of this paper, it is more convenient to follow the definition from [5].

Definition 6 (Branching bisimilarity). *Let $\mathcal{T} = (Q, L, \{\tau\}, \delta)$ be an LTS with a unique silent action τ . Then*

1. $R \subseteq Q \times Q$ is a branching bisimulation for \mathcal{T} if - for all $(p_0, q_0) \in R$ and $a \in L$, the following holds:

(a) If $p_0 \xrightarrow{a} p_1$ in \mathcal{T} , then $a = \tau$ and $(p_1, q_0) \in R$ or

there exists a path $q_0 \xrightarrow{\tau^k} q_1 \xrightarrow{a} q_2$ in \mathcal{T} , such that $k \geq 0$, $(p_0, q_1) \in R$ and $(p_1, q_2) \in R$;

- (b) If $q_0 \xrightarrow{a} q_1$ in \mathcal{T} , then $a = \tau$ and $(p_0, q_1) \in R$ or
there exists a path $p_0 \xrightarrow{\tau^k} p_1 \xrightarrow{a} p_2$ in \mathcal{T} , such that $k \geq 0$, $(p_1, q_0) \in R$ and $(p_2, q_1) \in R$.
2. If $(p, q) \in Q \times Q$, then p and q are branching bisimilar in \mathcal{T} , denoted $p \approx_{bb} q$, whenever there is a branching bisimulation $R \subseteq Q \times Q$ for \mathcal{T} such that $(p, q) \in R$.
 3. Two initialised labelled transition systems with τ as silent action, $\mathcal{T}_1 = (Q_1, L, \{\tau\}, \delta_1, q_{1,in})$ and $\mathcal{T}_2 = (Q_2, L, \{\tau\}, \delta_2, q_{2,in})$ such that $Q_1 \cap Q_2 = \emptyset$, are branching bisimilar if there exists a branching bisimulation $R \subseteq Q_1 \times Q_2$ such that $(q_{1,in}, q_{2,in}) \in R$.

To establish whether an implementation of a reference Industry net is correct, the communicating behaviour of the implementation has to be compared with the communicating behaviour of the reference model.

Definition 7 (Global compliance). Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of E-nets and φ a matching over V . Let $\mathcal{I}(V', \varphi)$ be an implementation of $\mathcal{I}(V, \varphi)$. Then $\mathcal{I}(V', \varphi)$ is compliant with $\mathcal{I}(V, \varphi)$, denoted $\mathcal{I}(V', \varphi) \simeq \mathcal{I}(V, \varphi)$, if $\lambda_{V', \tau}(\mathcal{LTS}(V', \varphi)) \approx_{bb} \lambda_{V, \tau}(\mathcal{LTS}(V, \varphi))$.

A problem here is that we cannot investigate the behaviour of the full implementation as such. The reason being that in general enterprises are reluctant to disclose information about the implementation of their internal operations to the public domain. Each enterprise has knowledge of its own operations, but not of the operations of the other enterprises. So we must assume that the full set of implementations of the enterprise nets is not known. What can be done locally however, is to investigate the behaviour of each local E-net implementation by comparing it with the behaviour of the corresponding original E-net in the reference model. To make this comparison we should consider these behaviours *in the same context*. We can accomplish this by replacing the E-net in the reference model with its local implementation.

Definition 8 (Local compliance). Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of E-nets and φ a matching over V . Let $i \in [n]$ and \mathcal{E}'_i an implementation of \mathcal{E}_i with respect to V . Let $V^i = (V \setminus \{\mathcal{E}_i\}) \cup \{\mathcal{E}'_i\}$.

- Then $\mathcal{I}(V^i, \varphi)$ is the local replacement of \mathcal{E}_i in $\mathcal{I}(V, \varphi)$ by \mathcal{E}'_i .
- \mathcal{E}'_i is compliant with $\mathcal{I}(V, \varphi)$, denoted $\mathcal{E}'_i \sim \mathcal{I}(V, \varphi)$, if $\lambda_{V^i, \tau}(\mathcal{LTS}(V^i, \varphi)) \approx_{bb} \lambda_{V, \tau}(\mathcal{LTS}(V, \varphi))$, for all $i \in [n]$

We now claim as our main result that if local compliance is established for all local replacements of enterprise nets in the reference model, then global compliance of the full implementation follows. More formally, the following theorem holds:

Theorem 1. *Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of E-nets, φ a matching over V , and $V' = \{\mathcal{E}'_i : i \in [n]\}$ where for all $i \in [n]$, \mathcal{E}'_i an implementation of \mathcal{E}_i with respect to V . Then $\mathcal{I}(V', \phi) \simeq \mathcal{I}(V, \phi)$ if $\mathcal{E}'_i \sim \mathcal{I}(V, \phi)$ for all $i \in [n]$.*

To prove this result, we introduce in Section 5 the new, auxiliary notion of *qualified branching bisimilarity*. This facilitates our reasoning about the equivalence of modular process specifications. In general, this concept of bisimilarity, as it is based on the distinction between different silent actions, does not coincide with branching bisimilarity. For the transition systems defined by I-nets, it can be shown however that it leads to the same equivalence classes. We will use this result to give a rough outline of the proof of Theorem 1 in Section 5.

5 Qualified branching bisimilarity

Before we can introduce the new variant of branching bisimilarity, we need a modification of labelled transition systems that reflect the modular specification of I-nets. A *partitioned labelled transition system*, or *PLTS*, is a tuple $\mathcal{T}_P = (Q, [L_1, \dots, L_n], \delta)$, such that the L_i are nonempty, pairwise disjoint sets of (action) labels and $\mathcal{T} = (Q, \bigcup_{i \in [n]} L_i, \delta)$ is a labelled transition system. We refer to \mathcal{T}_P as the *partitioned labelled transition system derived from \mathcal{T} and $[L_1, \dots, L_n]$* ; and to \mathcal{T} as the *underlying labelled transition system of \mathcal{T}_P* . All terminology and notation introduced for labelled transition systems carry over to partitioned labelled transition systems. As with labelled transition systems we can add silent actions and an initial state to the specification of a partitioned labelled transition system. This leads to tuples of the form $(Q, [L_1, \dots, L_n], S, \delta, q_{in})$ with $S \subseteq \bigcup_{i \in [n]} L_i$.

Definition 9 (Partitioned transition system of a I-net). *Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of E-nets and φ a matching over V . Let μ_{in} be the default initial marking of $\mathcal{I}(V, \varphi) = (P, T, F)$.*

– *Let, for all $i \in [n]$, $N_i = (P_i, T_i, F_i)$ be the underlying Petri net of \mathcal{E}_i . Then $\mathcal{PLTS}(\mathcal{I}(V, \varphi)) = ([\mu_{in}], [T_1, \dots, T_n], \emptyset, \delta, \mu_{in})$, the PLTS derived from $\mathcal{LTS}(V, \varphi)$ and $[T_1, \dots, T_n]$, is the initialized partitioned labelled transition system of $\mathcal{I}(V, \varphi)$.*

– *Let, for each $i \in [n]$, $T_{int,i}$ be the set of internal actions of \mathcal{E}_i . Let $\mathcal{S} = \{(\tau, i) \mid i \in [n]\}$ be a set of n new (silent action) labels, where (τ, i) is associated with \mathcal{E}_i , for $i \in [n]$. Let $\ell_{V,S} : T \rightarrow (T \setminus \bigcup_{i \in [n]} T_{int,i}) \cup \mathcal{S}$ be defined as*

$$\ell_{V,S}(t) = \begin{cases} (\tau, i) & \text{if } t \in T_{int,i} \text{ for some } i \in [n] \\ t & \text{otherwise} \end{cases}$$

Then $\ell_{V,S}(\mathcal{PLTS}(\mathcal{I}(V, \varphi))) = ([\mu_{in}], [\ell(T_1), \dots, \ell(T_n)], \mathcal{S}, \ell(\delta), \mu_{in})$, is the initialized labelled transition system with silent actions \mathcal{S} of $\mathcal{I}(V, \varphi)$.

In what follows, we write $\mathcal{PLTS}(V, \varphi)$ rather than $\mathcal{PLTS}(\mathcal{I}(V, \varphi))$.

The specification of the transition system $\mathcal{PLTS}(V, \varphi)$ provides a partition of its labels according to the enterprise nets from V . Moreover, the labels of the internal transitions of each Enterprise net $\mathcal{E}_i \in V$ are made silent using a dedicated symbol (τ, i) .

The following notion of branching bisimilarity requires that in each simulation all actions (silent and non-silent ones) come from the same subset of labels.

Definition 10 (Qualified branching bisimulation). *Let $n \geq 2$. Let $\mathcal{T} = (Q, [L_1, \dots, L_n], S, \delta)$ be a PLTS with silent actions. Let, for all $i \in [n]$, $S_i = L_i \cap S$. Then*

1. $R \subseteq Q \times Q$ is a qualified branching bisimulation for \mathcal{T} if - for all $(p_0, q_0) \in R$ and $a \in \bigcup_{i \in [n]} L_i$, the following holds:
 - (a) if $p_0 \xrightarrow{a} p_1$, then $a \in S$ and $(p_1, q_0) \in R$ or $a \in L_i$ for some $i \in [n]$ and there exists a path $q_0 \xrightarrow{w} q_1 \xrightarrow{a} q_2$ in \mathcal{T} such that $w \in S_i^*$, $(p_0, q_1) \in R$ and $(p_1, q_2) \in R$;
 - (b) if $q_0 \xrightarrow{a} q_1$, then $a \in S$ and $(p_0, q_1) \in R$ or $a \in L_i$ for some $i \in [n]$ and there exists a path $p_0 \xrightarrow{w} p_1 \xrightarrow{a} p_2$ in \mathcal{T} such that $w \in S_i^*$, $(p_1, q_0) \in R$ and $(p_2, q_1) \in R$.
2. Let $p, q \in Q$. Then p and q are qualified branching bisimilar in \mathcal{T} , denoted $p \approx_{qb} q$, whenever there exists a qualified branching bisimulation $R \subseteq Q \times Q$ for \mathcal{T} such that $(p, q) \in R$.
3. Two initialised partitioned labelled transition systems with silent actions $\mathcal{T}_1 = (Q_1, [L_1, \dots, L_n], S, \delta_1, q_{1,in})$ and $\mathcal{T}_2 = (Q_2, [L_1, \dots, L_n], S, \delta_2, q_{2,in})$ such that $Q_1 \cap Q_2 = \emptyset$, are qualified branching bisimilar, denoted $\mathcal{T}_1 \approx_{qb} \mathcal{T}_2$, if $q_{1,in}$ and $q_{2,in}$ are qualified branching bisimilar in $(Q_1 \cup Q_2, [L_1, \dots, L_n], S, \delta_1 \cup \delta_2)$.

Branching bisimilarity does not imply qualified branching bisimilarity. Consider Figure 1. Let $L_1 = \{a_1\}$, $L_2 = \{a_2, \tau_2\}$, $L = L_1 \cup L_2$ and $S = \{\tau_2\}$. Let $\mathcal{T}_1 = (\{p_1, p_2\}, L, S, \{(p_1, a_1, p_2)\}, p_1)$ and $\mathcal{T}_2 = (\{q_1, q_2, q_3\}, L, S, \{(q_1, \tau_2, q_2), (q_2, a_1, q_3)\}, p_1)$ be two initialized LTSs (with silent action $\tau = \tau_2$). Then the dotted lines in Figure 1 represent a branching bisimulation between \mathcal{T}_1 and \mathcal{T}_2 .

Now let $\mathcal{T}_{1,P} = (\{p_1, p_2\}, [L_1, L_2], S, \{(p_1, a_1, p_2)\}, p_1)$ be the initialized PLTS with silent actions derived from \mathcal{T}_1 and $[L_1, L_2]$ and $\mathcal{T}_{2,P} = (\{q_1, q_2, q_3\}, [L_1, L_2], S, \{(q_1, \tau_2, q_2), (q_2, a_1, q_3)\}, p_1)$ the initialized PLTS

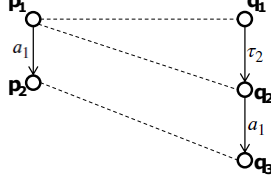


Fig. 1: \approx_{bb} does not imply \approx_{qb}

with silent actions derived from \mathcal{T}_2 and $[L_1, L_2]$. Then it is not possible to find a qualified branching bisimulation between $\mathcal{T}_{1,P}$ and $\mathcal{T}_{2,P}$ because a_1 and τ_2 are from different subsets of labels. However there is a strong relationship between branching bisimilarity and qualified branching bisimilarity for labelled transition systems of I-nets which is captured in Theorem 2.

Theorem 2 (\approx_{bb} and \approx_{qb} for I-nets). *Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ and $V' = \{\mathcal{E}'_i : i \in [n]\}$ be composable sets of E-nets. Let φ be a matching over V and φ' a matching over V' . Let $\mathcal{S} = \{(\tau, i) \mid i \in [n]\}$ be a set of n new action labels, with for all $i \in [n]$, (τ, i) associated with \mathcal{E}_i in V and with \mathcal{E}'_i in V' . Then $\lambda_{V,\tau}(\mathcal{LTS}(V, \varphi)) \approx_{bb} \lambda_{V',\tau}(\mathcal{LTS}(V', \varphi))$ if and only if $\ell_{V,\mathcal{S}}(\mathcal{PLTS}(V, \varphi)) \approx_{qb} \ell_{V',\mathcal{S}}(\mathcal{PLTS}(V', \varphi))$.*

The if direction of Theorem 2 follows directly from Definitions 5, 6, 9 and 10. To prove the only-if direction is however more involved. The basic idea is as follows (we sketch the qualified simulation of $\ell_{V,\mathcal{S}}(\mathcal{PLTS}(V, \varphi))$ by $\ell_{V',\mathcal{S}}(\mathcal{PLTS}(V', \varphi))$). From $\lambda_{V,\tau}(\mathcal{LTS}(V, \varphi)) \approx_{bb} \lambda_{V',\tau}(\mathcal{LTS}(V', \varphi))$ it follows that when a transition labelled a from an Enterprise net $\mathcal{E}_i \in V$ is simulated by a sequence wa in $\lambda_{V',\tau}(\mathcal{LTS}(V', \varphi))$, then w can be any combination of silent actions from various E-nets in V' . Because of the modularity of I-nets and the Petri net semantics it is then possible to first execute the transitions underlying the occurrences in w from \mathcal{E}'_i (labelled by (τ, i)). These form a sequence w' and it can be shown that qualified branching bisimilarity is preserved after $w'a$.

6 Outline of proof of Theorem 1

In this section we sketch a proof of Theorem 1. For the sake of convenience we repeat the statement here:

Theorem 1 Let $n \geq 2$. Let $V = \{\mathcal{E}_i : i \in [n]\}$ be a composable set of E-nets, φ a matching over V , and $V' = \{\mathcal{E}'_i : i \in [n]\}$ where for all $i \in [n]$,

\mathcal{E}'_i is an implementation of \mathcal{E}_i with respect to V . Then $\mathcal{I}(V', \phi) \simeq \mathcal{I}(V, \phi)$ if, for all $i \in [n]$, \mathcal{E}'_i is locally compliant with $\mathcal{I}(V, \varphi)$.

Let us assume that, for all $i \in [n]$, \mathcal{E}'_i is locally compliant with $\mathcal{I}(V, \varphi)$, that is $\lambda_{V^i, \tau}(\mathcal{LTS}(V^i, \varphi)) \approx_{bb} \lambda_{V, \tau}(\mathcal{LTS}(V, \varphi))$, see Definition 8. Recall that $V^i = (V \setminus \{\mathcal{E}_i\}) \cup \{\mathcal{E}'_i\}$.

From Theorem 2 we have $\ell_{V, \mathcal{S}}(\mathcal{PLTS}(V, \varphi)) \approx_{qb} \ell_{V^i, \mathcal{S}}(\mathcal{PLTS}(V^i, \varphi))$ for all $i \in [n]$. We will argue next how from this we can deduce that $\ell_{V, \mathcal{S}}(\mathcal{PLTS}(V, \varphi)) \approx_{qb} \ell_{V', \mathcal{S}}(\mathcal{PLTS}(V', \varphi))$. The theorem then follows from Theorem 2.

Let μ_{in} and μ'_{in} be the default initial markings of $\mathcal{I}(V, \varphi)$ and $\mathcal{I}(V', \varphi)$, respectively. For all $i \in [n]$, μ^i_{in} denotes the default initial marking of $\mathcal{I}(V^i, \varphi)$ and $R_i \subseteq [\mu_{in}] \times [\mu'_{in}]$ is a qualified branching bisimulation between $\ell_{V, \mathcal{S}}(\mathcal{PLTS}(V, \varphi))$ and $\ell_{V^i, \mathcal{S}}(\mathcal{PLTS}(V^i, \varphi))$.

In addition, we define relations Z_i for all $i \in [n]$, consisting of all pairs of markings (μ^i, μ') in $[\mu^i_{in}] \times [\mu'_{in}]$ such that $\mu^i(p) = \mu'(p)$ for all places p shared by $\mathcal{I}(V^i, \phi)$ and $\mathcal{I}(V', \phi)$. Note that $\mathcal{I}(V^i, \phi)$ and $\mathcal{I}(V', \phi)$ share all and only the places of \mathcal{E}'_i and the channel places $P(V', \varphi)$.

As a consequence, if $(\mu^i, \mu') \in Z_i$ for some $i \in [n]$, then every transition t of \mathcal{E}'_i enabled at μ^i in $\mathcal{I}(V^i, \phi)$ is also enabled at μ' in $\mathcal{I}(V', \phi)$ and vice versa; moreover, if $\mu^i \xrightarrow{t} \mu^i_1$ in $\mathcal{I}(V^i, \phi)$ and $\mu' \xrightarrow{t} \mu'_1$ in $\mathcal{I}(V', \phi)$, then also $(\mu^i_1, \mu'_1) \in Z_i$ (property A)⁵.

Finally, we define the relation R by $R = \bigcap_{i \in [n]} R_i \circ Z_i$. We briefly consider the steps followed to establish that R is a qualified branching bisimulation between $\ell_{V, \mathcal{S}}(\mathcal{PLTS}(V, \varphi))$ and $\ell_{V', \mathcal{S}}(\mathcal{PLTS}(V', \varphi))$, as desired.

See Figure 2 for an illustration. The large squares in Figure 2 represent the transition systems associated with the I-nets considered. The smaller squares represent E-nets.

It is easy to see that $(\mu_{in}, \mu'_{in}) \in R$. Assume $(\mu, \mu') \in R$. Let, for all $i \in [n]$, μ^i be such that $(\mu, \mu^i) \in R_i$ and $(\mu^i, \mu') \in Z_i$.

Let $\mu \xrightarrow{a} \mu_1$ in $\ell_{V, \mathcal{S}}(\mathcal{PLTS}(V, \varphi))$. Then there is a unique \mathcal{E}_j such that $a = \ell_{V, \mathcal{S}}(t)$ for some transition t of \mathcal{E}_j .

Since R_j is a qualified branching bisimulation between $\ell_{V, \mathcal{S}}(\mathcal{PLTS}(V, \varphi))$ and $\ell_{V^j, \mathcal{S}}(\mathcal{PLTS}(V^j, \varphi))$, it follows that $\mu \xrightarrow{a} \mu_1$ can be simulated from μ^j in $\ell_{V^j, \mathcal{S}}(\mathcal{PLTS}(V^j, \varphi))$, as described in Definition 10, by $\mu^j \xrightarrow{wa} \mu^j_1 \xrightarrow{a} \mu^j_2$ where wa is a sequence of zero or more transitions from \mathcal{E}'_j such that $(\mu, \mu^j_1) \in R_j$ and $(\mu_1, \mu^j_2) \in R_j$. Then, by property A, we can conclude that

⁵ This holds for $t \in T'_{int, i}$ because \mathcal{E}'_i is shared between $\mathcal{I}(V^i, \phi)$ and $\mathcal{I}(V', \phi)$ and for $t \in T'_i \setminus T'_{int, i}$ because $\mathcal{I}(V^i, \phi)$ and $\mathcal{I}(V', \phi)$ have the same communication subnet.

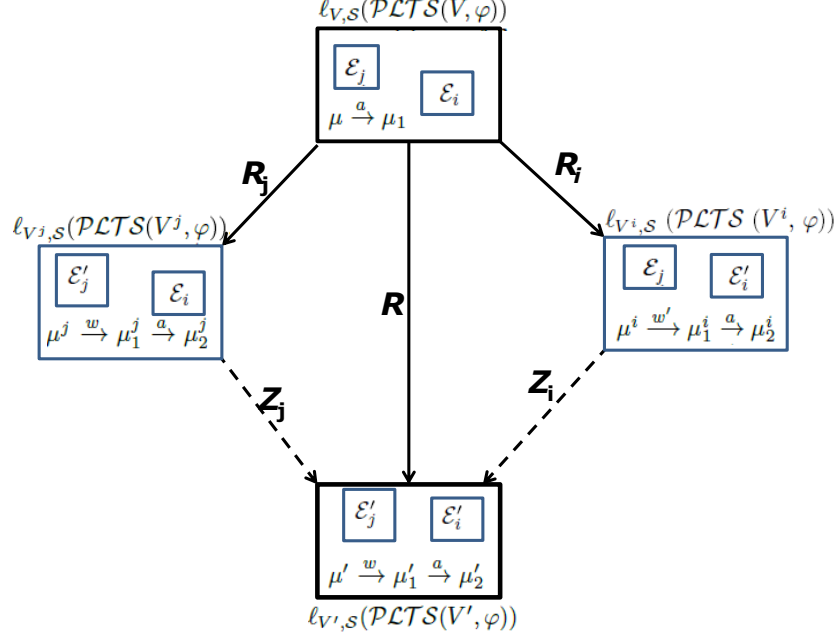


Fig. 2: R is a qualified branching bisimulation

$\mu' \xrightarrow{w} \mu'_1 \xrightarrow{a} \mu'_2$ in $\ell_{V',S}(\mathcal{P}\mathcal{L}\mathcal{T}\mathcal{S}(V', \varphi))$ with $(\mu_1^j, \mu'_1) \in Z_j$ and $(\mu_2^j, \mu'_2) \in Z_j$. Hence $(\mu, \mu'_1) \in R_j \circ Z_j$ and $(\mu_1, \mu'_2) \in R_j \circ Z_j$.

For all $i \neq j$, we can apply a similar reasoning and conclude that a sequence $\mu^i \xrightarrow{w'} \mu_1^i \xrightarrow{a} \mu_2^i$ (with $w'a$ a sequence from $\mathcal{E}_j \in V^i$) in $\ell_{V^i,S}(\mathcal{P}\mathcal{L}\mathcal{T}\mathcal{S}(V^i, \varphi))$ such that $(\mu, \mu_1^i) \in R_i$ and $(\mu_1, \mu_2^i) \in R_i$. Since the transitions involved in the simulation are not from $\mathcal{E}'_i \in V^i$, it follows that $(\mu_1^i, \mu'_1) \in Z_i$ and $(\mu_2^i, \mu'_2) \in Z_i$ and so $(\mu_1, \mu'_1) \in R_i \circ Z_i$ and $(\mu_2, \mu'_2) \in R_i \circ Z_i$. Consequently $\mu' \xrightarrow{w} \mu'_1 \xrightarrow{a} \mu'_2$ in $\ell_{V',S}(\mathcal{P}\mathcal{L}\mathcal{T}\mathcal{S}(V', \varphi))$ with $(\mu, \mu'_1) \in R$ and $(\mu_1, \mu'_2) \in R$.

To prove the other direction, assume that $(\mu, \mu') \in R$ and $\mu' \xrightarrow{a} \mu'_1$ in $\ell_{V',S}(\mathcal{P}\mathcal{L}\mathcal{T}\mathcal{S}(V', \varphi))$. Again, we can identify the unique E-net with the transition t such that $\ell_{V',S}(t) = a$ and then follow a similar pattern as before.

Note that qualified branching bisimulation is crucial for the conclusion that, for all $i \in [n]$, the relation Z_i is respected by each step in the simulation.

7 Related work

The ideas underlying the concepts of E-nets, I-nets, and compliance belong to a well-established branch of research concerned with composing systems into larger correctly functioning (concurrent) systems (see, eg., [10] for an overview).

In [13] Petri nets are studied equipped with distinguished input and output labels (I/O Petri nets) indicating their possibilities to communicate with other I/O Petri nets. An asynchronous composition operator introduces new channels for the communication between the composed nets. The transition system semantics of these nets is compositional and the composition of transition systems preserves channel properties. Composition of local workflows is considered in [20]. The open workflow nets of [21] are Petri nets with interface places enabling the communication with other open workflow nets.

The composition of Petri nets as considered in [20, 21] and also in [2] (open workflow nets) all combine local Petri nets extended with interface places which enables them to communicate with other Petri nets. In our approach, similarly to the approach in [13], local Petri nets do not have interface places but they can communicate via distinguished input and output labels. As mentioned in [13], the advantage of this approach is that it leads to a better separation of concerns: eg., the designer of a local Petri net does not have to consider whether it will be used in a synchronous or in an asynchronous environment later on; this should be a concern for the designer of the global Petri net.

An overview of research on inter-organisational workflows is given in [4]. Much of this work is concerned with the construction of inter-organisational workflows. The *Public-To-Private*(P2P) approach in [3] is concerned with the construction of local workflows as subclasses of global workflows. Inheritance preserving transformations should then guarantee compliance of a local implementation (“compliance by construction”). In [6] refinement of local states of a Petri net is supported by behaviour preserving morphisms between the Petri net and its refinement. Weak bisimilarity is used as the criterion for the preservation of behaviour.

The approach in [2] assumes a collaborative workflow, a *multi-party contract* between the participants, represented as an open workflow net and serving as a reference model. Then, a local implementation (called *private view*) is in *accordance* with the open net included in the reference model (the *public view*) if they both operate correctly (without deadlocks and terminating correctly) in the same context. Accordance can be checked locally and is compositional, i.e., if all private views are in accordance with the public view, then the overall collaborative workflow is guaranteed to

be deadlock-free and terminating correctly. It should be noted here that accordance is a weaker notion than compliance as considered in this paper: compliance implies accordance, but the converse does not always hold.

In [18, 16, 17] Symbolic Observation Graphs (SOG) are used to represent the behaviour of enterprise level business processes such that critical design information is kept private. It is shown that deadlock-freeness [18] and soundness [16] of inter-enterprise level business processes can then be verified using the composition of these enterprise level SOG's. In [17] this approach is extended to include sharing of resources, asynchronous communication between enterprise level business processes and the verification of specific workflow properties using Linear Temporal Logic.

8 Conclusion

We have introduced I-nets as a means to capture global requirements concerning communication while leaving design decisions concerning internal operations to the local level. Compliance is then defined as branching bisimilarity between the labelled transition system of the original (global) reference model and the labelled transition system of its implementation (emerging from the local implementations). The use of qualified branching bisimilarity simplified the proof of our main theorem, Theorem 1, that local compliance for all local implementations implies global compliance for the global implementation. This provides a formal underpinning for an approach that allows verification of global compliance by checking local compliance only.

To our knowledge this is a new result in the current approaches towards local verification of compliant global behaviour (see also the previous section on related work). In particular, our result is not comparable to the congruence results for branching bisimilarity in process algebras by which branching bisimilar processes when inserted in the same context, yield again branching bisimilar processes [11]. First of all, we did not formally introduce an algebra, but — more importantly — compliance does not assume or guarantee that the local processes (from the reference model and an implementation) are branching bisimilar.

In [19], the industry level requirements are captured in a reference model, then projected onto the enterprise level, and subsequently translated into an LTL formula. To verify that the transition system of the local replacement satisfies the LTL-formula, model checking is used. In this approach, the context information in the reference model is filtered out by the projection step. Furthermore, it requires the extra step to translate the requirements laid down in the (projected) reference model into an LTL formula, which is not required in the approach presented in this paper.

Our approach relies essentially on well-established concepts and results in the field of Petri nets, labelled transition systems and branching bisimilarity. This may allow implementation of automated support for our approach using already existing high performance tools incorporating proven technology like *LTSmin* [15] with only minor changes, for checking local compliance of E-nets. This will be the subject of future work.

References

1. van der Aalst, W.M.P., Basten, T.: Inheritance of workflows: an approach to tackling problems related to change. *Theor. Comput. Sci.* 270(1-2), 125–203 (2002), [https://doi.org/10.1016/S0304-3975\(00\)00321-2](https://doi.org/10.1016/S0304-3975(00)00321-2)
2. van der Aalst, W.M.P., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K.: Multiparty contracts: Agreeing and implementing interorganizational processes. *Comput. J.* 53(1), 90–106 (2010), <https://doi.org/10.1093/comjnl/bxn064>
3. van der Aalst, W.M.P., Weske, M.: The P2P approach to interorganizational workflows. In: *Advanced Information Systems Engineering, 13th International Conference, CAiSE 2001, Interlaken, Switzerland, June 4-8, 2001, Proceedings.* pp. 140–156 (2001), http://dx.doi.org/10.1007/3-540-45341-5_10
4. van der Aalst, W.M.P., Weske, M.: Reflections on a decade of interorganizational workflow research. In: *Seminal Contributions to Information Systems Engineering, 25 Years of CAiSE*, pp. 307–313 (2013), http://dx.doi.org/10.1007/978-3-642-36926-1_24
5. Basten, T.: Branching bisimilarity is an equivalence indeed! *Inf. Process. Lett.* 58(3), 141–147 (1996), [http://dx.doi.org/10.1016/0020-0190\(96\)00034-8](http://dx.doi.org/10.1016/0020-0190(96)00034-8)
6. Bernardinello, L., Mangioni, E., Pomello, L.: Local state refinement and composition of elementary net systems: An approach based on morphisms. *Trans. Petri Nets and Other Models of Concurrency* 8, 48–70 (2013), https://doi.org/10.1007/978-3-642-40465-8_3
7. EDSN: Marktfacilitering. <https://www.edsn.nl/> (Mar 2018)
8. van Glabbeek, R.J.: The linear time-branching time spectrum (extended abstract). In: *CONCUR '90, Theories of Concurrency: Unification and Extension*, Amsterdam, The Netherlands, August 27-30, 1990, *Proceedings.* pp. 278–297 (1990), <http://dx.doi.org/10.1007/BFb0039066>
9. van Glabbeek, R.J., Weijland, W.P.: Branching time and abstraction in bisimulation semantics. *J. ACM* 43(3), 555–600 (1996), <http://doi.acm.org/10.1145/233551.233556>
10. Gomes, L., Barros, J.P.: Structuring and composability issues in petri nets modeling. *IEEE Trans. Industrial Informatics* 1(2), 112–123 (2005), <https://doi.org/10.1109/TII.2005.844433>
11. Gorrieri, R., Versari, C.: *Introduction to Concurrency Theory: Transition Systems and CCS.* Springer International Publishing, Cham (2015), https://doi.org/10.1007/978-3-319-21491-7_1

12. GS1US: Rosettanet. <http://www.rosettanet.org/> (Feb 2018)
13. Haddad, S., Hennicker, R., Møller, M.H.: Channel properties of asynchronously composed petri nets. In: Application and Theory of Petri Nets and Concurrency - 34th International Conference, PETRI NETS 2013, Milan, Italy, June 24-28, 2013. Proceedings. pp. 369–388 (2013), http://dx.doi.org/10.1007/978-3-642-38697-8_20
14. HL7: Health level seven international. <http://www.hl7.org/> (Feb 2018)
15. Kant, G., Laarman, A., Meijer, J., van de Pol, J., Blom, S., van Dijk, T.: Ltsmin: High-performance language-independent model checking. In: Baier, C., Tinelli, C. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 692–707. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
16. Klai, K., Desel, J.: Checking soundness of business processes compositionally using symbolic observation graphs. In: Giese, H., Rosu, G. (eds.) Formal Techniques for Distributed Systems - Joint 14th IFIP WG 6.1 International Conference, FMOODS 2012 and 32nd IFIP WG 6.1 International Conference, FORTE 2012, Stockholm, Sweden, June 13-16, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7273, pp. 67–83. Springer (2012), https://doi.org/10.1007/978-3-642-30793-5_5
17. Klai, K., Ochi, H.: A bottom-up approach to check the correctness of interorganisational workflows. In: 2015 International Symposium on Theoretical Aspects of Software Engineering, TASE 2015, Nanjing, China, September 12-14, 2015. pp. 7–14. IEEE Computer Society (2015), <https://doi.org/10.1109/TASE.2015.18>
18. Klai, K., Tata, S., Desel, J.: Symbolic abstraction and deadlock-freeness verification of inter-enterprise processes. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) Business Process Management, 7th International Conference, BPM 2009, Ulm, Germany, September 8-10, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5701, pp. 294–309. Springer (2009), https://doi.org/10.1007/978-3-642-03848-8_20
19. Kwantes, P.M., Gorp, P.V., Kleijn, J., Rensink, A.: Towards compliance verification between global and local process models. In: Graph Transformation - 8th International Conference, ICGT 2015, Held as Part of STAF 2015, L'Aquila, Italy, July 21-23, 2015. Proceedings. pp. 221–236 (2015), http://dx.doi.org/10.1007/978-3-319-21145-9_14
20. Martens, A.: On compatibility of web services. Petri Net Newsletter pp. 12–20
21. Massuthe, P., Reisig, W., Schmidt, K.: An Operating Guideline Approach to the SOA. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Informatik (2005)
22. Milner, R.: Communication and concurrency. PHI Series in computer science, Prentice Hall (1989)
23. S.W.I.F.T.: Iso20022 universal financial industry message scheme. <http://www.iso20022.org> (Mar 2015)