A Compositional Approach for Verifying Hierarchical Interface-Based Supervisory Control

Ryan Leduc * **Robi Malik** **

* *Department of Computing and Software, McMaster University, Hamilton, Canada, (e-mail: leduc@mcmaster.ca)*
** *Department of Computer Science, University of Waikato, Hamilton, New Zealand, (e-mail: robi@cs.waikato.ac.nz)*

Abstract: Hierarchical Interface-based Supervisory Control (HISC) decomposes a discrete-event system into a high-level subsystem which communicates through interfaces with several low-level subsystems. The framework provides a set of local conditions that can be checked for each subsystem individually to conclude global conditions such as nonblocking and controllability. The size of HISC systems that can be verified automatically is primarily limited by the size of the largest subsystem. To overcome this limitation, this paper proposes the use of compositional verification. Most of the HISC conditions can be verified efficiently using existing methods for compositional verification, but a few are more challenging. This paper shows how these more challenging conditions can be expressed equivalently as generalized nonblocking problems, so the compositional approach for generalized nonblocking developed by the authors in (Malik and Leduc, 2009) is applicable. This makes all the HISC conditions amenable for compositional verification, considerably increasing the size of systems that can be handled using the framework.

Keywords: Discrete event systems, large-scale systems, hierarchical control, finite state machines, verification.

1. INTRODUCTION

In the area of Discrete-Event Systems (DES), two common tasks are to verify that a composite system is (i) nonblocking and (ii) controllable (Ramadge and Wonham, 1989). The main obstacle to performing these tasks is the combinatorial explosion of the synchronous product state space.

The framework of Hierarchical Interface-based Supervisory Control (HISC) was proposed in (Leduc, 2002; Leduc *et al.*, 2005*a*; Leduc *et al.*, 2005*b*; Leduc *et al.*, 2006; Leduc, 2009) to alleviate the state explosion problem. The HISC approach decomposes a system into a *high-level subsystem* which communicates with one or more parallel *low-level subsystems* through separate interfaces that restrict the interaction of the subsystems. It provides a set of local conditions that can be used to verify global conditions such as nonblocking and controllability. As each local condition can be verified considering only a single subsystem, the complete system model never needs to be stored in memory, offering potentially significant savings in computational resources.

When checking the per-subsystem HISC conditions, each subsystem is treated as a flat system. The system size that can be handled is thus limited by the size of the largest subsystem. As typically each level is defined using several automata, it is appealing to verify these conditions by compositional verification, which has been used with considerable success for safety and nonblocking verification (Brandin *et al.*, 2004; Ware and Malik, 2008; Flordal and Malik, 2009). This avoids the explicit construction of the complete state space of each subsystem and thus increases the complexity of the subsystems that can be handled.

Although many of the HISC conditions can be directly expressed as a standard controllability or nonblocking problem, the LD interface consistency points 5 and 6 cannot. However, they are tantalizingly similar to standard nonblocking. Instead of requiring that every reachable state have a path to a marked state (coreachable), they only require that a subset of states be coreachable. This inspired the authors to come up with the *generalized nonblocking* condition, introduced in (Malik and Leduc, 2008). They then extended the compositional verification results for standard nonblocking to this new setting in (Malik and Leduc, 2009).

Our approach to develop an HISC compositional verification method is to first cast the required HISC conditions for a given subsystem as a generalized nonblocking problem, and then show that this new representation is equivalent. We then use our compositional approach for generalized nonblocking (Malik and Leduc, 2009) to check these conditions. In essence, we perform a hybrid hierarchical-compositional verification. Instead of checking controllability and nonblocking on the entire system directly, we first use the HISC method to break the system down into subsystems, and then we use the compositional method to check the HISC conditions for each subsystem. This allows us to leverage the hierarchical and information hiding structure.

Sect. 2 introduces the necessary background of nondeterministic automata and defines the generalized nonblocking property, and Sect. 3 provides an introduction to Hierarchical Interface-based Supervisory Control. Next, Sect. 4 presents our new results where the HISC definitions are expressed as generalized nonblocking problems and we provide equivalence proofs. Sect. 5 adds some concluding remarks.

2. PRELIMINARIES

2.1 Events and Languages

In this paper, discrete event systems are modeled using nondeterministic automata. While most concepts required for this paper can be explained using deterministic automata, nondeterminism is needed for compositional verification.

Event sequences and languages are a simple means to describe system behaviors. Their basic building blocks are *events*, which are taken from a finite *alphabet* Σ. In addition, the *silent event* $\tau \notin \Sigma$ is used, with the notation $\Sigma_\tau = \Sigma \cup \{\tau\}$.

Σ^* denotes the set of all finite *strings* of the form $\sigma_1\sigma_2\ldots\sigma_n$ of events from Σ, including the *empty string* ε. The *catenation* of $s, t \in \Sigma^*$ is written as st. A subset $L \subseteq \Sigma^*$ is called a *language*. For $\Omega \subseteq \Sigma$, *natural projection* $P_\Omega\colon \Sigma^* \to \Omega^*$ denotes the operation that deletes all events not in Ω from strings. For language $L \subseteq \Sigma^*$ and $s \in \Sigma^*$, the set of *eligible events* is $\mathrm{Elig}_L(s) := \{\, \sigma \in \Sigma \mid s\sigma \in L \,\}$.

Definition 1. A (nondeterministic) *automaton* is a tuple $G = \langle \Sigma, X, \to, X^\circ, X^m \rangle$ where Σ is a finite set of *events*, X is a set of *states*, $\to \subseteq X \times \Sigma_\tau \times X$ is the *state transition relation*, $X^\circ \subseteq X$ is the set of *initial states*, and $X^m \subseteq X$ is the set of *marked states*.

Definition 2. An automaton $G = \langle \Sigma, X, \to, X^\circ, X^m \rangle$ is *deterministic* if X° is a singleton, $x \xrightarrow{\sigma} y_1$ and $x \xrightarrow{\sigma} y_2$ always implies $y_1 = y_2$, and $\to$ contains no τ-transitions.

The transition relation is written in infix notation $x \xrightarrow{\sigma} y$, and is extended to strings in Σ_τ^* in the standard way. For state sets $X_1, X_2 \subseteq X$, the notation $X_1 \xrightarrow{s} X_2$ denotes the existence of $x_1 \in X_1$ and $x_2 \in X_2$ such that $x_1 \xrightarrow{s} x_2$. Also, $x \to y$ denotes the existence of a string $s \in \Sigma_\tau^*$ such that $x \xrightarrow{s} y$, and $x \xrightarrow{s}$ denotes the existence of a state $y \in X$ such that $x \xrightarrow{s} y$. Finally, $G \to x$ stands for $X^\circ \to x$.

To support silent events, another transition relation $\Rightarrow \subseteq X \times \Sigma^* \times X$ is introduced. Then $x \stackrel{s}{\Rightarrow} y$, with $s = \sigma_1\sigma_2\ldots\sigma_n \in \Sigma^*$, denotes the existence of a string $t \in \tau^*\sigma_1\tau^*\sigma_2\tau^*\cdots\tau^*\sigma_n\tau^*$ such that $x \xrightarrow{t} y$. That is, $x \xrightarrow{s} y$ denotes a path with *exactly* the events in s, while $x \stackrel{s}{\Rightarrow} y$ denotes a path with an arbitrary number of τ events shuffled with the events of s. Notations such as $X_1 \stackrel{s}{\Rightarrow} X_2$, $x \Rightarrow y$, and $x \stackrel{s}{\Rightarrow}$ are defined analogously to $\to$.

We denote the *closed behavior* of automaton G to be $L(G) = \{\, s \in \Sigma^* \mid X^\circ \stackrel{s}{\Rightarrow} \,\}$, and the *marked behavior* to be $L_m(G) = \{\, s \in \Sigma^* \mid X^\circ \stackrel{s}{\Rightarrow} X^m \,\}$.

Synchronous composition models the parallel execution of two or more automata, and is done using lock-step synchronization in the style of (Hoare, 1985).

Definition 3. Let $G_1 = \langle \Sigma_1, X_1, \to_1, X_1^\circ, X_1^m \rangle$ and $G_2 = \langle \Sigma_2, X_2, \to_2, X_2^\circ, X_2^m \rangle$ be two automata. The *synchronous product* $G_1 \parallel G_2$ of G_1 and G_2 is

$$\langle \Sigma_1 \cup \Sigma_2, X_1 \times X_2, \to, X_1^\circ \times X_2^\circ, X_1^m \times X_2^m \rangle \qquad (1)$$

where

$(x_1, x_2) \xrightarrow{\sigma} (y_1, y_2)$ if $\sigma \in (\Sigma_1 \cap \Sigma_2), x_1 \xrightarrow{\sigma}_1 y_1, x_2 \xrightarrow{\sigma}_2 y_2$;
$(x_1, x_2) \xrightarrow{\sigma} (y_1, x_2)$ if $\sigma \in (\Sigma_1 \cup \{\tau\}) \setminus \Sigma_2, x_1 \xrightarrow{\sigma}_1 y_1$;
$(x_1, x_2) \xrightarrow{\sigma} (x_1, y_2)$ if $\sigma \in (\Sigma_2 \cup \{\tau\}) \setminus \Sigma_1, x_2 \xrightarrow{\sigma}_2 y_2$.

For deterministic automata, we now define controllability. We start by assuming the standard event partition $\Sigma = \Sigma_u \,\dot\cup\, \Sigma_c$, splitting the alphabet into *uncontrollable* and *controllable* events. To control a given plant $G_1 = \langle \Sigma_1, X_1, \to_1, \{x_1^\circ\}, X_1^m \rangle$, we define a *supervisor* represented as an automaton $S = \langle \Sigma_S, X_S, \to_S, \{x_S^\circ\}, X_S^m \rangle$.

Definition 4. Let $\Sigma := \Sigma_1 \cup \Sigma_S$, $P_1\colon \Sigma^* \to \Sigma_1^*$, and $P_S\colon \Sigma^* \to \Sigma_S^*$. Define $L_1 := P_1^{-1}(L(G_1))$ and $L_S := P_S^{-1}(L(S))$. Supervisor S is *controllable* for plant G_1 if $L_S\Sigma_u \cap L_1 \subseteq L_S$ or, equivalently, $(\forall s \in L_1 \cap L_S)\, \mathrm{Elig}_{L_1}(s) \cap \Sigma_u \subseteq \mathrm{Elig}_{L_S}(s)$.

2.2 Generalized Nonblocking

It is a desirable property that every execution of an automaton can be completed by reaching a marked state in X^m, otherwise *livelock* or *deadlock* may occur. The following extends the standard nonblocking definition (Ramadge and Wonham, 1989) to the case of nondeterministic automata considered in this paper.

Definition 5. An automaton $G = \langle \Sigma, X, \to, X^\circ, X^m \rangle$ is called *(standard) nonblocking*, if for all states $x \in X$ such that $G \Rightarrow x$, it also holds that $x \Rightarrow X^m$.

Standard nonblocking requires that all reachable states be coreachable, i.e., that a marked state can be reached from each reachable state (Ramadge and Wonham, 1989). The *generalized nonblocking* property introduced in (Malik and Leduc, 2008) weakens this condition by only requiring a subset of states to be coreachable. This is expressed formally using *multi-colored* automata, extending the traditional concept of *marked states* to multiple simultaneous marking conditions by labeling states with different *colors* or *propositions*. The following definition is introduced in (Malik and Leduc, 2008), and is based on similar ideas in (Clarke *et al.*, 1999; de Queiroz *et al.*, 2004).

Definition 6. A *multi-colored automaton* is a tuple $G = \langle \Sigma, \Pi, X, \to, X^\circ, \Xi \rangle$ where Σ is a finite set of *events*, Π is a finite set of *propositions* or *colors*, $\to \subseteq X \times \Sigma_\tau \times X$ is the *state transition relation*, $X^\circ \subseteq X$ is the set of *initial states*, and $\Xi\colon \Pi \to 2^X$ defines the set of marked states for each proposition in Π.

Definition 7. Let $G_1 = \langle \Sigma_1, \Pi, X_1, \to_1, X_1^\circ, \Xi_1 \rangle$ and $G_2 = \langle \Sigma_2, \Pi, X_2, \to_2, X_2^\circ, \Xi_2 \rangle$ be two multi-colored automata. The synchronous product of G_1 and G_2 is

$$G_1 \parallel G_2 = \langle \Sigma, \Pi, X \to, X^\circ, \Xi \rangle \qquad (2)$$

where Σ, X, $\to$, and X° are given as in Def. 3, and $\Xi(\pi) = \Xi_1(\pi) \times \Xi_2(\pi)$ for each $\pi \in \Pi$.

For generalized nonblocking, we first define propositions α and ω. The intended meaning is that ω represents terminal states and corresponds to the marked state set X^m, while

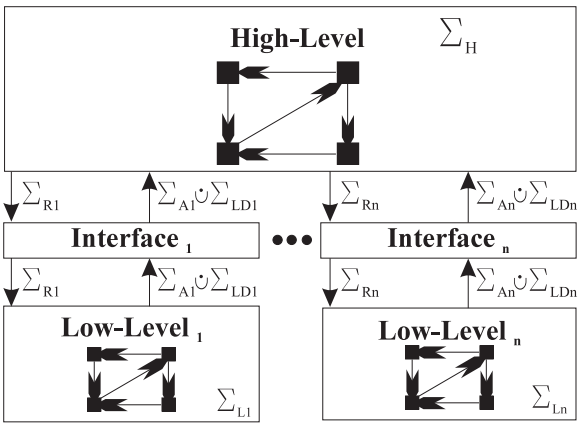

Fig. 1. Interface Block Diagram.

α specifies a set of states from which terminal states are required to be reachable.

Definition 8. Let $G = \langle \Sigma, \Pi, X, \rightarrow, X^\circ, \Xi \rangle$ with $\alpha, \omega \in \Pi$ be a multi-colored automaton. G is (α, ω)-*nonblocking*, if for all states $x \in \Xi(\alpha)$ such that $G \Rightarrow x$ it also holds that $x \Rightarrow \Xi(\omega)$.

Standard nonblocking can be expressed using generalized nonblocking simply by taking $\Xi(\alpha) = X$. The relationship between generalized nonblocking and standard nonblocking along with some applications is discussed in (Malik and Leduc, 2008).

3. HISC WITH LOW DATA EVENTS

This section gives a brief introduction to Hierarchical Interface-based Supervisory Control (HISC). For an illustrative example please see (Leduc *et al.*, 2005*a*), and for more detailed explanation and justification please refer to (Leduc *et al.*, 2005*b*; Leduc *et al.*, 2006).

An HISC system (Leduc, 2009) is a two-level system which includes one *high-level subsystem* and one or more *low-level subsystems*. The high-level subsystem communicates with each low-level subsystem through a separate *interface*. HISC systems are defined using only deterministic automata.

In HISC there is a master-slave relationship. The high-level subsystem sends a command to a particular low-level subsystem, which then performs the indicated task and returns an answer. Fig. 1 shows the conceptual structure and information flow of the system. This style of interaction is enforced by an interface that mediates communication between the two subsystems. All system components, including the interfaces, are modeled as automata.

In order to restrict information flow and decouple the subsystems, the system alphabet is partitioned into pairwise disjoint alphabets:

$$\Sigma := \Sigma_{\mathrm{H}} \ \dot\cup \dot{\bigcup_{j=1,\ldots,n}} (\Sigma_{\mathrm{L}_j} \dot\cup \Sigma_{\mathrm{R}_j} \dot\cup \Sigma_{\mathrm{A}_j} \dot\cup \Sigma_{\mathrm{LD}_j}) \qquad (3)$$

The events in Σ_{H} are called *high-level events*, and the events in Σ_{L_j} are the j^{th} *low-level events* ($j = 1, \ldots, n$), as these events appear only in the high-level and j^{th} low-level

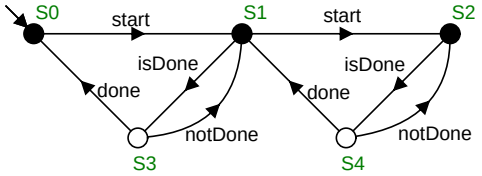

Fig. 2. Example LD Interface.

subsystem models, $\mathbf{G}_H$ and $\mathbf{G}_{L_j}$ respectively. We then have the high-level subsystem $\mathbf{G}_H$ defined over event set $\Sigma_{\mathrm{H}} \dot\cup (\dot\cup_{j\in\{1,\ldots,n\}} [\Sigma_{\mathrm{R}_j} \dot\cup \Sigma_{\mathrm{A}_j} \dot\cup \Sigma_{\mathrm{LD}_j}])$ and the j^{th} low-level subsystem $\mathbf{G}_{L_j}$ defined over event set $\Sigma_{\mathrm{L}_j} \dot\cup \Sigma_{\mathrm{R}_j} \dot\cup \Sigma_{\mathrm{A}_j} \dot\cup \Sigma_{\mathrm{LD}_j}$. We model the j^{th} *interface* by DES $\mathbf{G}_{I_j}$, which is defined over the event set $\Sigma_{\mathrm{R}_j} \dot\cup \Sigma_{\mathrm{A}_j} \dot\cup \Sigma_{\mathrm{LD}_j}$. We define the *flat system* to be $\mathbf{G} = \mathbf{G}_H \parallel \mathbf{G}_{I_1} \parallel \mathbf{G}_{L_1} \parallel \cdots \parallel \mathbf{G}_{I_n} \parallel \mathbf{G}_{L_n}$.

As the j^{th} interface $\mathbf{G}_{I_j}$ is only concerned with communication between the subsystems, it is defined only over the events that are common to both levels of the hierarchy. The events in Σ_{R_j} are called *request events* and represent commands sent from the high-level subsystem to the j^{th} low-level subsystem. The events in Σ_{A_j} are *answer events* and represent the low-level subsystem's responses to the request events. The events in Σ_{LD_j} are called *low data events*. These events were introduced in (Leduc, 2009) to produce more general and more powerful interfaces than the original HISC architecture. Low data events provide a means for a low-level to send information (data) through the interface, independently of the standard request-answer pattern. Request, answer, and low data events are collectively known as the set of *interface events*, defined as $\Sigma_{\mathrm{I}} := \dot\cup_{j=1}^{n} (\Sigma_{\mathrm{R}_j} \dot\cup \Sigma_{\mathrm{A}_j} \dot\cup \Sigma_{\mathrm{LD}_j})$.

In order to enforce the serialization of requests and answers, we restrict the interfaces to the subclass of LD interfaces defined below. Fig. 2 shows an example of an LD interface, where $\Sigma_{\mathrm{R}_j} = \{isDone, start\}$, $\Sigma_{\mathrm{A}_j} = \{done\}$, and $\Sigma_{\mathrm{LD}_j} = \{notDone\}$. It could correspond to a machine at the low-level with an effective internal buffer of two.

Definition 9. A deterministic automaton $\mathbf{G}_I = \langle \Sigma_{\mathrm{I}}, X, \rightarrow, \{x^\circ\}, X^m \rangle$ is an *LD interface* for the alphabet partition $\Sigma_{\mathrm{I}} = \Sigma_{\mathrm{R}} \dot\cup \Sigma_{\mathrm{A}} \dot\cup \Sigma_{\mathrm{LD}}$ if the following conditions are satisfied.

(i) $x^\circ \in X^m$;
(ii) If $x \xrightarrow{\sigma} y$ for some $x \in X^m$, then $\sigma \in \Sigma_{\mathrm{R}}$, or $\sigma \in \Sigma_{\mathrm{LD}}$ and $y \in X^m$.
(iii) If $x \xrightarrow{\sigma} y$ for some $x \notin X^m$, then $\sigma \in \Sigma_{\mathrm{LD}}$, or $\sigma \in \Sigma_{\mathrm{A}}$ and $y \in X^m$.

To simplify notation, we bring in the following event sets, natural projections, and languages. In particular, languages such as $\mathcal{H}$ represent the behavior of a given DES extended over Σ^*.

$$\Sigma_{\mathrm{I}_j} := \Sigma_{\mathrm{R}_j} \dot\cup \Sigma_{\mathrm{A}_j} \dot\cup \Sigma_{\mathrm{LD}_j}, \qquad P_{I_j} : \Sigma^* \to \Sigma_{\mathrm{I}_j}^*$$
$$\Sigma_{\mathrm{IL}_j} := \Sigma_{\mathrm{L}_j} \cup \Sigma_{\mathrm{I}_j}, \qquad P_{IL_j} : \Sigma^* \to \Sigma_{\mathrm{IL}_j}^*$$
$$\Sigma_{\mathrm{IH}} := \Sigma_{\mathrm{H}} \cup \bigcup_{j\in\{1,\ldots,n\}} \Sigma_{I_j}, \qquad P_{IH} : \Sigma^* \to \Sigma_{\mathrm{IH}}^*$$
$$\Sigma_{\mathrm{LD}} := \bigcup_{j\in\{1,\ldots,n\}} \Sigma_{\mathrm{LD}_j}$$

$$\mathcal{H} := P_{IH}^{-1}(L(\mathbf{G}_H)), \quad \mathcal{H}_m := P_{IH}^{-1}(L_m(\mathbf{G}_H)) \subseteq \Sigma^*$$
$$\mathcal{L}_j := P_{IL_j}^{-1}(L(\mathbf{G}_{L_j})), \quad \mathcal{L}_{m_j} := P_{IL_j}^{-1}(L_m(\mathbf{G}_{L_j})) \subseteq \Sigma^*$$
$$\mathcal{I}_j := P_{I_j}^{-1}(L(\mathbf{G}_{I_j})), \quad \mathcal{I}_{m_j} := P_{I_j}^{-1}(L_m(\mathbf{G}_{I_j})) \subseteq \Sigma^*$$
$$\mathcal{I} := \bigcap_{j\in\{1,\ldots,n\}} \mathcal{I}_j, \qquad \mathcal{I}_m := \bigcap_{j\in\{1,\ldots,n\}} \mathcal{I}_{m_j}$$

The following *interface consistency* properties are necessary to ensure that the high and low-level subsystems interact with the interfaces correctly.

Definition 10. The n^{th} degree interface system composed of DES $\mathbf{G}_H, \mathbf{G}_{I_1}, \mathbf{G}_{L_1}, \ldots, \mathbf{G}_{I_n}, \mathbf{G}_{L_n}$ is *LD interface consistent (LDIC)* with respect to the alphabet partition (3), if for all $j \in \{1,\ldots,n\}$, the following conditions are satisfied:

Multi-level Properties

(1) The event set of $\mathbf{G}_H$ is Σ_{IH}, and the event set of $\mathbf{G}_{L_j}$ is Σ_{IL_j}.
(2) $\mathbf{G}_{I_j}$ is an LD interface.

High-Level Property

(3) $(\forall s \in \mathcal{H}\cap\mathcal{I})\ \text{Elig}_{\mathcal{I}_j}(s)\ \cap\ (\Sigma_{\text{A}_j} \dot\cup \Sigma_{\text{LD}_j}) \subseteq \text{Elig}_{\mathcal{H}}(s)$

Low-Level Properties

(4) $(\forall s \in \mathcal{L}_j\cap\mathcal{I}_j)\ \text{Elig}_{\mathcal{I}_j}(s)\ \cap\ \Sigma_{\text{R}_j} \subseteq \text{Elig}_{\mathcal{L}_j}(s)$
(5) $(\forall s \in \mathcal{L}_j\cap\mathcal{I}_j)(\forall \rho\in\Sigma_{\text{R}_j})(\forall\sigma\in\Sigma_{\text{A}_j})$
if $s\rho\sigma \in \mathcal{I}_j$ then $(\exists l\in\Sigma_{L_j}^*)\ s\rho l\sigma \in \mathcal{L}_j\cap\mathcal{I}_j$
(6) $(\forall s \in \mathcal{L}_j\cap\mathcal{I}_j)$
if $s\in\mathcal{I}_{m_j}$ then $(\exists l\in\Sigma_{L_j}^*)\ sl\in\mathcal{L}_{m_j}\cap\mathcal{I}_{m_j}$.

The conditions in Def. 10 are referred to as the *LD interface consistency (LDIC) properties* in the following. They are local conditions of individual subsystems, designed to capture the way an HISC subsystem should behave to ensure correct communication with other subsystems. LDIC properties 1 and 2 are syntactic properties ensuring that all DES use the correct alphabet, and that all interfaces have the appropriate structure for HISC. LDIC property 3 is a controllability-like property that requires the high-level always to be ready to accept any answer or low data event the low-levels may produce according to the interfaces. Similarly, LDIC property 4 requires the low-levels always to be capable of accepting requests that may come from the high-level. Finally, LDIC properties 5 and 6 are nonblocking-like requirements to be satisfied by the low-level, discussed in more detail in Sect. 4 below. Essentially, the low-level is required to be able to eventually execute all answers possible according to the interface and to terminate, but only immediately after the interface has entered particular states. For more details, please refer to (Leduc, 2009; Leduc *et al.*, 2009).

3.1 Local Conditions for Global Nonblocking

If the following *level-wise nonblocking* properties are satisfied in addition to interface consistency, this is enough to conclude that the flat system is to be nonblocking.

Definition 11. The n^{th} degree interface system composed of DES $\mathbf{G}_H, \mathbf{G}_{I_1}, \mathbf{G}_{L_1}, \ldots, \mathbf{G}_{I_n}, \mathbf{G}_{L_n}$ is said to be *LD level-wise nonblocking (LDLWNB)* if the following conditions are satisfied:

(I) *LD-nonblocking at the high-level*:
$(\forall s\in\mathcal{H}\cap\mathcal{I})(\exists s'\in(\Sigma\setminus\Sigma_{\text{LD}})^*)\ ss'\in\mathcal{H}_m\cap\mathcal{I}_m$
(II) *nonblocking at the low-level*:
$\mathbf{G}_{L_j} \parallel \mathbf{G}_{I_j}$ is nonblocking for each $j = 1,\ldots,n$.

Theorem 1. (Leduc, 2009) If the n^{th} degree ($n \geq 1$) interface system composed of deterministic DES $\mathbf{G}_H, \mathbf{G}_{I_1}, \mathbf{G}_{L_1}, \ldots, \mathbf{G}_{I_n}, \mathbf{G}_{L_n}$, is LD level-wise nonblocking and LD interface consistent with respect to the alphabet partition (3), then $\mathbf{G} = \mathbf{G}_H \parallel \mathbf{G}_{L_1} \parallel \mathbf{G}_{I_1} \parallel \cdots \parallel \mathbf{G}_{L_n} \parallel \mathbf{G}_{I_n}$ is nonblocking.

3.2 Local Conditions for Global Controllability

For controllability, we need to split the subsystems into their plant and supervisor components. We define the *high-level plant* to be $\mathbf{G}_H^p$, and the *high-level supervisor* to be $\mathbf{S}_H$ (both defined over event set Σ_{IH}). Similarly, the j^{th} *low-level plant* and *supervisor* are $\mathbf{G}_{L_j}^p$ and $\mathbf{S}_{L_j}$ (defined over Σ_{IL_j}). The high-level subsystem and the j^{th} low-level subsystem are then $\mathbf{G}_H := \mathbf{G}_H^p \| \mathbf{S}_H$ and $\mathbf{G}_{L_j} := \mathbf{G}_{L_j}^p \| \mathbf{S}_{L_j}$, respectively.

We can now define our *flat supervisor* and *plant* as well as some useful languages as follows:

$$\mathbf{Plant} := \mathbf{G}_H^p \parallel \mathbf{G}_{L_1}^p \parallel \cdots \parallel \mathbf{G}_{L_n}^p$$
$$\mathbf{Sup} := \mathbf{S}_H \parallel \mathbf{S}_{L_1} \parallel \cdots \parallel \mathbf{S}_{L_n} \parallel \mathbf{G}_{I_1} \parallel \ldots \parallel \mathbf{G}_{I_n}$$
$$\mathcal{H}^p := P_{IH}^{-1}L(\mathbf{G}_H^p), \qquad \mathcal{S}_H := P_{IH}^{-1}L(\mathbf{S}_H) \subseteq \Sigma^*$$
$$\mathcal{L}_j^p := P_{IL_j}^{-1}L(\mathbf{G}_{L_j}^p), \quad \mathcal{S}_{L_j} := P_{IL_j}^{-1}L(\mathbf{S}_{L_j}) \subseteq \Sigma^*$$

We now provide the controllability requirements that each level must satisfy. For a discussion of the individual points of Def. 12, please refer to (Leduc *et al.*, 2006).

Definition 12. The n^{th} degree interface system composed of DES $\mathbf{G}_H^p, \mathbf{S}_H, \mathbf{G}_{L_1}^p, \mathbf{S}_{L_1}, \mathbf{G}_{I_1}, \ldots, \mathbf{G}_{L_n}^p, \mathbf{S}_{L_n}, \mathbf{G}_{I_n}$ is *LD level-wise controllable (LDLWC)* with respect to the alphabet partition (3), if for all $j \in \{1,\ldots,n\}$ the following conditions hold:

(I) The alphabet of $\mathbf{G}_H^p$ and $\mathbf{S}_H$ is Σ_{IH}, the alphabet of $\mathbf{G}_{L_j}^p$ and $\mathbf{S}_{L_j}$ is Σ_{IL_j}, and the alphabet of $\mathbf{G}_{I_j}$ is Σ_{I_j}
(II) $(\forall s\in\mathcal{L}_j^p\cap\mathcal{S}_{L_j}\cap\mathcal{I}_j)\ \text{Elig}_{\mathcal{L}_j^p}(s)\cap\Sigma_u \subseteq \text{Elig}_{\mathcal{S}_{L_j}\cap\mathcal{I}_j}(s)$
(III) $(\forall s\in\mathcal{H}^p\cap\mathcal{I}\cap\mathcal{S}_H)\ \text{Elig}_{\mathcal{H}^p\cap\mathcal{I}}(s)\cap\Sigma_u \subseteq \text{Elig}_{\mathcal{S}_H}(s)$

Theorem 2. (Leduc, 2009) If the n^{th} degree ($n \geq 1$) interface system composed of DES $\mathbf{G}_H^p, \mathbf{S}_H, \mathbf{G}_{L_1}^p, \mathbf{S}_{L_1}, \mathbf{G}_{I_1}, \ldots, \mathbf{G}_{L_n}^p, \mathbf{S}_{L_n}, \mathbf{G}_{I_n}$ is LD level-wise controllable with respect to the alphabet partition given by (3), then
$(\forall s\in L(\mathbf{Plant})\cap L(\mathbf{Sup}))$
$\text{Elig}_{L(\mathbf{Plant})}(s)\cap\Sigma_u \subseteq \text{Elig}_{L(\mathbf{Sup})}(s)$

4. COMPOSITIONAL VERIFICATION OF HISC

This section explains how to compositionally verify the HISC conditions for nonblocking and controllability.

To verify that an HISC system is nonblocking, it must be shown to satisfy LDIC Properties 1–6 (Def. 10) and LDLWNB Properties I–II (Def. 11). First, LDIC Properties 1 and 2 are syntactical conditions that can be checked easily by inspecting each automaton individually. LDIC Properties 3 and 4 are essentially controllability properties, once suitable definitions of plant, supervisor, and

uncontrollable events have been made. These properties can be checked efficiently using methods for compositional verification of safety properties such as (Brandin *et al.*, 2004; Ware and Malik, 2008). LDLWNB Property II is a standard nonblocking property, which can be checked directly using the compositional approach for standard nonblocking in (Flordal and Malik, 2009), or by converting the property into generalized nonblocking (Malik and Leduc, 2008) and using compositional verification of generalized nonblocking (Malik and Leduc, 2009).

To verify that an HISC system is controllable, it is enough to verify LDLWC Properties I–III (Def. 12), which are either syntactical (LDLWC Property I) or can be treated as standard controllability verification problems (LDLWC Property II and III).

We thus only need to provide compositional methods to check LDIC Properties 5 and 6, and LDLWNB Property I. In the following, each of these three remaining properties is addressed in a subsection of its own.

4.1 LDIC Property 5

LDIC Property 5 is similar to standard nonblocking if states where answer event σ is possible are considered as marked states. However, instead of requiring a path to these states from all reachable states as in standard nonblocking, such a path is required only from states *immediately* after request event ρ. Also, we require that the path to a marked state only contains low-level events. This property can be expressed using generalized nonblocking by marking precisely those states α that are entered immediately after request event ρ, and introducing a new DES to restrict the occurrence of interface events while testing whether a given answer σ can occur.

The following definition and result generalizes the solution from (Malik and Leduc, 2008) to include low data events. To verify that an n^{th} order HISC system satisfies LDIC Property 5, we have to check the property for each of the n subsystems. To keep things simple, we give definitions for a single low level $\mathbf{G}_L = \mathbf{G}_{L_j}$ and interface $\mathbf{G}_I = \mathbf{G}_{I_j}$ ($j \in \{1, \ldots, n\}$), with associated event sets $\Sigma_{\mathrm{R}} = \Sigma_{\mathrm{R}_j}$, $\Sigma_{\mathrm{A}} = \Sigma_{\mathrm{A}_j}$, $\Sigma_{\mathrm{LD}} = \Sigma_{\mathrm{LD}_j}$, and $\Sigma_{\mathrm{L}} = \Sigma_{\mathrm{L}_j}$.

Definition 13. Let $\Sigma = \Sigma_{\mathrm{R}} \,\dot\cup\, \Sigma_{\mathrm{A}} \,\dot\cup\, \Sigma_{\mathrm{LD}} \,\dot\cup\, \Sigma_{\mathrm{L}}$, $\Sigma_{\mathrm{I}} = \Sigma_{\mathrm{R}} \cup \Sigma_{\mathrm{A}} \cup \Sigma_{\mathrm{LD}}$ and let $\mathbf{G}_I = \langle \Sigma_{\mathrm{I}}, X_I, \rightarrow_I, \{x_I^\circ\}, X_I^m \rangle$ and $\mathbf{G}_L = \langle \Sigma, X_L, \rightarrow_L, \{x_L^\circ\}, X_L^m \rangle$ be two deterministic automata. $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy *LDIC Property 5* if, for all strings $s \in \Sigma^*$ and all events $\rho \in \Sigma_{\mathrm{R}}$ such that $s\rho \in L(\mathbf{G}_I \parallel \mathbf{G}_L)$, and for all events $\sigma \in \Sigma_{\mathrm{A}}$ such that $P_{\Sigma_{\mathrm{I}}}(s)\rho\sigma \in L(\mathbf{G}_I)$, there exists $t \in \Sigma_L^*$ such that $s\rho t\sigma \in L(\mathbf{G}_I \parallel \mathbf{G}_L)$.

For automata $\mathbf{G}_I$ and $\mathbf{G}_L$ and answer event $\sigma \in \Sigma_{\mathrm{A}}$, we construct the multi-colored automata $\mathbf{G}_I^\sigma$, $\mathbf{G}_L^\sigma$, and $\mathbf{T}^\sigma$ such that $\mathbf{G}_I^\sigma \parallel \mathbf{G}_L^\sigma \parallel \mathbf{T}^\sigma$ is (α, ω)-nonblocking if and only if LDIC Property 5 is satisfied for the given answer σ. The construction is as follows.

- $\mathbf{G}_I^\sigma$ is obtained from $\mathbf{G}_I$ by adding propositions α and ω such that all states with σ enabled are marked α, and all states are marked ω. If $\mathbf{G}_I = \langle \Sigma_{\mathrm{I}}, X_I, \rightarrow_I, \{x_I^\circ\}, X_I^m \rangle$, then $\mathbf{G}_I^\sigma = \langle \Sigma_{\mathrm{I}}, \{\alpha, \omega\}, X_I, \rightarrow_I, \{x_I^\circ\}, \Xi_I \rangle$ with $\Xi_I(\alpha) = \{\, x \in X_I \mid x \xrightarrow{\sigma} \,\}$ and $\Xi_I(\omega) = X_I$.

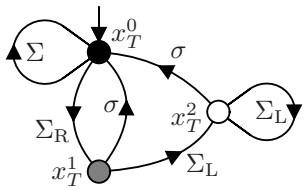

Fig. 3. The automaton $\mathbf{T}^\sigma$ for translating LDIC Property 5 into (α, ω)-nonblocking.

- $\mathbf{G}_L^\sigma$ is obtained from $\mathbf{G}_L$ by marking all states both α and ω. If $\mathbf{G}_L = \langle \Sigma, X_L, \rightarrow_L, \{x_L^\circ\}, X_L^m \rangle$, then $\mathbf{G}_L^\sigma = \langle \Sigma, \{\alpha, \omega\}, X_L, \rightarrow_L, \{x_L^\circ\}, \Xi_L \rangle$ with $\Xi_L(\alpha) = \Xi_L(\omega) = X_L$.
- $\mathbf{T}^\sigma = \langle \Sigma, \{\alpha, \omega\}, X_T, \{x_T^0\}, \rightarrow_T, \Xi_T \rangle$ with $X_T = \{x_T^0, x_T^1, x_T^2\}$, $\Xi_T(\alpha) = \{x_T^1\}$, and $\Xi_T(\omega) = \{x_T^0\}$ is the nondeterministic multi-colored automaton in Fig. 3.

Proposition 3. For each $\sigma \in \Sigma_{\mathrm{A}}$ construct multi-colored automata $\mathbf{G}_I^\sigma$, $\mathbf{G}_L^\sigma$, and $\mathbf{T}^\sigma$ as explained above. $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 5 if and only if $\mathbf{G}_I^\sigma \| \mathbf{G}_L^\sigma \| \mathbf{T}^\sigma$ is (α, ω)-nonblocking for each $\sigma \in \Sigma_{\mathrm{A}}$.

Proof. First, let $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 5. Let $\sigma \in \Sigma_{\mathrm{A}}$ such that $\mathbf{G}_I^\sigma \parallel \mathbf{G}_L^\sigma \parallel \mathbf{T}^\sigma \xrightarrow{s} (x_I, x_L, x_T) \in \Xi_I(\alpha) \times \Xi_L(\alpha) \times \Xi_T(\alpha)$. Since $\Xi_T(\alpha) = \{x_T^1\}$ by construction, it holds that $x_T = x_T^1$ and $s = s'\rho$ for some $s' \in \Sigma^*$ and $\rho \in \Sigma_{\mathrm{R}}$. Also $\mathbf{G}_I \parallel \mathbf{G}_L \xrightarrow{s} (x_I, x_L)$, and since $x_I \in \Xi_I(\alpha)$, it holds by construction that $\mathbf{G}_I \xrightarrow{P_{\Sigma_{\mathrm{I}}}(s)} x_I \xrightarrow{\sigma} y_I$ for some $y_I \in X_I = \Xi_I(\omega)$. Therefore, $s'\rho = s \in L(\mathbf{G}_I \parallel \mathbf{G}_L)$ and $P_{\Sigma_{\mathrm{I}}}(s')\rho\sigma \in L(\mathbf{G}_I)$. By LDIC Property 5, there exists $t \in \Sigma_{\mathrm{L}}^*$ such that $st\sigma = s'\rho t\sigma \in L(\mathbf{G}_I \parallel \mathbf{G}_L)$. Since $\mathbf{G}_L$ is deterministic, there exists $y_L \in X_L = \Xi_L(\omega)$ such that $x_L \xrightarrow{t\sigma} y_L$. Furthermore, note that $x_T = x_T^1 \xrightarrow{t\sigma} x_T^0 \in \Xi_T(\omega)$ for any string $t \in \Sigma_{\mathrm{L}}^*$. Therefore, $(x_I, x_L, x_T) \xrightarrow{t\sigma} \Xi_I(\omega) \times \Xi_L(\omega) \times \Xi_T(\omega)$, i.e., $\mathbf{G}_I^\sigma \parallel \mathbf{G}_L^\sigma \parallel \mathbf{T}^\sigma$ is (α, ω)-nonblocking.

Second, let $\mathbf{G}_I^\sigma \parallel \mathbf{G}_L^\sigma \parallel \mathbf{T}^\sigma$ be (α, ω)-nonblocking for all $\sigma \in \Sigma_{\mathrm{A}}$. Let $s \in \Sigma^*$, $\rho \in \Sigma_{\mathrm{R}}$, and $\sigma \in \Sigma_{\mathrm{A}}$ be such that $s\rho \in L(\mathbf{G}_I \parallel \mathbf{G}_L)$ and $P_{\Sigma_{\mathrm{I}}}(s)\rho\sigma \in L(\mathbf{G}_I)$. Then there exist states $x_I \in X_I$ and $x_L \in X_L$ such that $\mathbf{G}_I \xrightarrow{P_{\Sigma_{\mathrm{I}}}(s)\rho} x_I \xrightarrow{\sigma}$ and $\mathbf{G}_L \xrightarrow{s\rho} x_L$. Also, $\mathbf{T}^\sigma \xrightarrow{s\rho} x_T^1$ by construction of $\mathbf{T}^\sigma$, and therefore $\mathbf{G}_I^\sigma \parallel \mathbf{G}_L^\sigma \parallel \mathbf{T}^\sigma \xrightarrow{s\rho} (x_I, x_L, x_T^1) \in \Xi_I(\alpha) \times \Xi_L(\alpha) \times \Xi_T(\alpha)$. Since $\mathbf{G}_I^\sigma \parallel \mathbf{G}_L^\sigma \parallel \mathbf{T}^\sigma$ is (α, ω)-nonblocking, there exists $u \in \Sigma^*$ such that $(x_I, x_L, x_T^1) \overset{u}{\Rightarrow} \Xi_I(\omega) \times \Xi_L(\omega) \times \Xi_T(\omega)$. By construction of $\mathbf{T}^\sigma$, and since $\Xi_T(\omega) = \{x_T^0\}$, there exists a prefix $t\sigma$ of u such that $t \in \Sigma_{\mathrm{L}}^*$ and $(x_I, x_L, x_T^1) \overset{t\sigma}{\Rightarrow}$, i.e., $s\rho t\sigma \in L(\mathbf{G}_I \parallel \mathbf{G}_L)$. Since s, ρ, and σ have been chosen arbitrarily, $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 5. □

After this construction, the compositional approach for generalized nonblocking (Malik and Leduc, 2009) can be used to verify LDIC property 5. The construction can be performed modularly and therefore is feasible for composed systems: if $\mathbf{G}_I$ and $\mathbf{G}_L$ are composed of several automata, then the construction can be applied to each automaton individually. Furthermore, the construction typically produces systems where the majority of states are

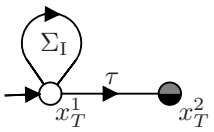

Fig. 4. The automaton $\mathbf{T}^{vi}$ for translating LDIC Property 6 into (α,ω)-nonblocking.

not marked α. This is important since several of the more powerful generalized nonblocking-preserving abstractions presented in (Malik and Leduc, 2009) require states that are not marked α.

4.2 LDIC Property 6

LDIC Property 6 is similar to the standard nonblocking property except that we only require states marked by the interface, but not necessarily the corresponding low level, to be coreachable. Also, we require that the path to a marked state only contains low-level events.

To verify that an n^{th} order HISC system satisfies LDIC Property 6, we have to check the property for each of the n subsystems. Again, we give definitions for a single low level $\mathbf{G}_L = \mathbf{G}_{L_j}$ and interface $\mathbf{G}_I = \mathbf{G}_{I_j}$, with associated event sets $\Sigma_{\mathrm{R}} = \Sigma_{\mathrm{R}_j}$, $\Sigma_{\mathrm{A}} = \Sigma_{\mathrm{A}_j}$, $\Sigma_{\mathrm{LD}} = \Sigma_{\mathrm{LD}_j}$, and $\Sigma_{\mathrm{L}} = \Sigma_{\mathrm{L}_j}$.

Definition 14. Let $\Sigma = \Sigma_{\mathrm{R}} \dot\cup \Sigma_{\mathrm{A}} \dot\cup \Sigma_{\mathrm{LD}} \dot\cup \Sigma_{\mathrm{L}}$, $\Sigma_{\mathrm{I}} = \Sigma_{\mathrm{R}} \cup \Sigma_{\mathrm{A}} \cup \Sigma_{\mathrm{LD}}$, and let $\mathbf{G}_I = \langle \Sigma_{\mathrm{I}}, X_I, \rightarrow_I, \{x_I^\circ\}, X_I^m\rangle$ and $\mathbf{G}_L = \langle \Sigma, X_L, \rightarrow_L, \{x_L^\circ\}, X_L^m\rangle$ be two deterministic automata. $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy *LDIC Property 6* if for all strings $s \in \Sigma^*$ such that $\mathbf{G}_I \| \mathbf{G}_L \overset{s}{\Rightarrow} (x^I, x^L)$ and $x^I \in X_I^m$, there exists $l \in \Sigma_L^*$ such that $(x^I, x^L) \overset{l}{\Rightarrow} X_I^m \times X_L^m$.

LDIC Property 6 can be expressed directly using generalized nonblocking. This is achieved by marking precisely those states α in $\mathbf{G}_I \parallel \mathbf{G}_L$ that are marked by $\mathbf{G}_I$. Multi-colored automata $\mathbf{G}_I^{vi}$, $\mathbf{G}_L^{vi}$, and $\mathbf{T}^{vi}$ are constructed such that $\mathbf{G}_I^{vi} \| \mathbf{G}_L^{vi} \| \mathbf{T}^{vi}$ is (α,ω)-nonblocking if and only if $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 6:

- From $\mathbf{G}_I$, we define $\mathbf{G}_I^{vi} = \langle \Sigma_{\mathrm{I}}, \{\alpha,\omega\}, X_I, \rightarrow_I, \{x_I^\circ\}, \Xi_I^{vi}\rangle$ where $\Xi_I^{vi}(\alpha) = \Xi_I^{vi}(\omega) = X_I^m$.
- From $\mathbf{G}_L$, we define $\mathbf{G}_L^{vi} = \langle \Sigma, \{\alpha,\omega\}, X_L, \rightarrow_L, \{x_L^\circ\}, \Xi_L^{vi}\rangle$ where $\Xi_L^{vi}(\alpha) = X_L$ and $\Xi_L^{vi}(\omega) = X_L^m$.
- We define $\mathbf{T}^{vi} = \langle \Sigma_{\mathrm{I}}, \{\alpha,\omega\}, \{x_T^1, x_T^2\}, \rightarrow_T, \{x_T^1\}, \Xi_T^{vi}\rangle$, with $\rightarrow_T = \{(x_T^1, \tau, x_T^2)\} \cup \bigcup_{\sigma\in\Sigma_{\mathrm{I}}} \{(x_T^1, \sigma, x_T^1)\}$ and $\Xi_T^{vi}(\alpha) = \Xi_T^{vi}(\omega) = \{x_T^2\}$, to be the nondeterministic multi-colored automaton in Fig. 4.

Proposition 4. $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 6 if and only if $\mathbf{G}_I^{vi} \parallel \mathbf{G}_L^{vi} \parallel \mathbf{T}^{vi}$ is (α,ω)-nonblocking.

Proof. First assume that $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 6, and let $\mathbf{G}_I^{vi} \parallel \mathbf{G}_L^{vi} \parallel \mathbf{T}^{vi} \overset{s}{\Rightarrow} (x^I, x^L, x^T) \in \Xi_I^{vi}(\alpha) \times \Xi_L^{vi}(\alpha) \times \Xi_T^{vi}(\alpha)$. Then by construction $\mathbf{G}_I \overset{P_{\Sigma_{\mathrm{I}}}(s)}{\Longrightarrow} x^I \in \Xi_I^{vi}(\alpha) = X_I^m$, i.e., $\mathbf{G}_I \| \mathbf{G}_L \overset{s}{\Rightarrow} (x^I, x^L)$ and $x^I \in X_I^m$. As $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 6, there exists $l \in \Sigma_L^*$ such that $(x^I, x^L) \overset{l}{\Rightarrow} (y^I, y^L) \in X_I^m \times X_L^m$. It follows that

- in $\mathbf{G}_I^{vi}$: $x^I \overset{P_{\Sigma_{\mathrm{I}}}(l)}{\Longrightarrow} y^I \in \Xi_I^{vi}(\omega)$ as $\Xi_I^{vi}(\omega) = X_I^m$.

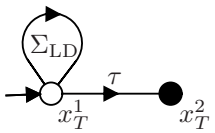

Fig. 5. The automaton $\mathbf{T}^{\mathrm{LD}}$ for translating LD nonblocking into standard nonblocking.

- in $\mathbf{G}_L^{vi}$: $x^L \overset{l}{\Rightarrow} y^L \in \Xi_L^{vi}(\omega)$ as $\Xi_L^{vi}(\omega) = X_L^m$.
- in $\mathbf{T}^{vi}$: $x^T \overset{P_{\Sigma_{\mathrm{I}}}(l)}{\Longrightarrow} x_T^2 \in \Xi_T^{vi}(\omega)$ by construction of $\mathbf{T}^{vi}$ as $P_{\Sigma_{\mathrm{I}}}(l) = \varepsilon$.

Thus, $\mathbf{G}_I^{vi} \parallel \mathbf{G}_L^{vi} \parallel \mathbf{T}^{vi} \overset{s}{\Rightarrow} (x^I, x^L, x^T) \overset{l}{\Rightarrow} (y^I, y^L, x_T^2) \in \Xi_I^{vi}(\omega) \times \Xi_L^{vi}(\omega) \times \Xi_T^{vi}(\omega)$, i.e., $\mathbf{G}_I^{vi} \parallel \mathbf{G}_L^{vi} \parallel \mathbf{T}^{vi}$ is (α,ω)-nonblocking.

Second, assume that $\mathbf{G}_I^{vi} \| \mathbf{G}_L^{vi} \| \mathbf{T}^{vi}$ is (α,ω)-nonblocking. Let $s \in \Sigma^*$, $\mathbf{G}_I \parallel \mathbf{G}_L \overset{s}{\Rightarrow} (x^I, x^L)$, and $x^I \in X_I^m$. Then by construction $\mathbf{G}_I^{vi} \overset{P_{\Sigma_{\mathrm{I}}}(s)}{\Longrightarrow} x^I \in X_I^m = \Xi_I^{vi}(\alpha)$ and $\mathbf{G}_L^{vi} \overset{s}{\Rightarrow} x^L \in X_L = \Xi_L^{vi}(\alpha)$ and $\mathbf{T}^{vi} \overset{P_{\Sigma_{\mathrm{I}}}(s)}{\longrightarrow} x_T^1 \overset{\tau}{\rightarrow} x_T^2$, i.e., $\mathbf{T}^{vi} \overset{P_{\Sigma_{\mathrm{I}}}(s)}{\Longrightarrow} x_T^2 \in \Xi_T^{vi}(\alpha)$. It follows that $\mathbf{G}_I^{vi} \parallel \mathbf{G}_L^{vi} \parallel \mathbf{T}^{vi} \overset{s}{\Rightarrow} (x^I, x^L, x_T^2) \in \Xi_I^{vi}(\alpha) \times \Xi_L^{vi}(\alpha) \times \Xi_T^{vi}(\alpha)$. Since $\mathbf{G}_I^{vi} \| \mathbf{G}_L^{vi} \| \mathbf{T}^{vi}$ is (α,ω)-nonblocking, there exists $t \in \Sigma^*$ such that $(x^I, x^L, x_T^2) \overset{t}{\Rightarrow} (y^I, y^L, y_T) \in \Xi_I^{vi}(\omega) \times \Xi_L^{vi}(\omega) \times \Xi_T^{vi}(\omega)$. Then $x_T^2 \overset{P_{\Sigma_{\mathrm{I}}}(t)}{\Longrightarrow}$, which by construction of $\mathbf{T}^{vi}$ implies $P_{\Sigma_{\mathrm{I}}}(t) = \varepsilon$, i.e., $t \in \Sigma_L^*$. Furthermore $y^I \in \Xi_I^{vi}(\omega) = X_I^m$ and $y^L \in \Xi_L^{vi}(\omega) = X_L^m$, which means that $(x^I, x^L) \overset{t}{\Rightarrow} X_I^m \times X_L^m$ with $t \in \Sigma_L^*$. Thus, $\mathbf{G}_I$ and $\mathbf{G}_L$ satisfy LDIC Property 6. □

Like in the case of LDIC property 5, this construction can be performed modularly and produces a large number of states not marked α, making it well suited for the compositional approach in (Malik and Leduc, 2009).

4.3 LDLWNB Property I

LDLWNB Property I is very similar to standard nonblocking. Every reachable state of the high-level must be coreachable, only the path to a marked state must not contain any low data events. To verify that an n^{th} order HISC system satisfies LDLWNB Property I, it is sufficient to verify that $\mathbf{G} = \mathbf{G}_H \parallel \mathbf{G}_{I_1} \parallel \cdots \parallel \mathbf{G}_{I_n}$ satisfies the LD-nonblocking definition below, with Σ_{LD} set to the HISC system's low data events.

Definition 15. Let $\mathbf{G} = \langle \Sigma, X, \rightarrow, X^\circ, X^m\rangle$ and $\Sigma_{\mathrm{LD}} \subseteq \Sigma$. Then $\mathbf{G}$ is called LD-nonblocking, if for all $s \in \Sigma^*$ and all $x \in X$ such that $\mathbf{G} \overset{s}{\Rightarrow} x$, there exists $t \in (\Sigma \setminus \Sigma_{\mathrm{LD}})^*$ such that $x \overset{t}{\Rightarrow} X^m$.

The following result shows how this LD-nonblocking condition can be rewritten equivalently as a standard nonblocking property. This makes it possible to verify LDLWNB Property I using the compositional approach for standard nonblocking (Flordal and Malik, 2009) or (after conversion) for generalized nonblocking (Malik and Leduc, 2009).

Proposition 5. Let $\mathbf{G} = \langle \Sigma, X, \rightarrow, X^\circ, X^m\rangle$ be an automaton, and $\Sigma_{\mathrm{LD}} \subseteq \Sigma$. Furthermore, let $\mathbf{T}^{\mathrm{LD}} = \langle \Sigma_{\mathrm{LD}},$

$\{x_T^1, x_T^2\}, \rightarrow_T, \{x_T^1\}, \{x_T^2\}\rangle$ where $\rightarrow_T = \{(x_T^1, \tau, x_T^2)\} \cup \bigcup_{\sigma \in \Sigma_{\mathrm{LD}}} \{(x_T^1, \sigma, x_T^1)\}$ be the nondeterministic automaton in Fig. 5. Then $\mathbf{G}$ is LD-nonblocking if and only if $\mathbf{G} \,\|\, \mathbf{T}^{\mathrm{LD}}$ is nonblocking.

Proof. Let $\mathbf{G}$ be LD-nonblocking, and let $\mathbf{G} \parallel \mathbf{T}^{\mathrm{LD}} \stackrel{s}{\Rightarrow} (x, x_T)$. Then $\mathbf{G} \stackrel{s}{\Rightarrow} x$, and since $\mathbf{G}$ is LD-nonblocking, there exists $t \in (\Sigma \setminus \Sigma_{\mathrm{LD}})^*$ such that $\mathbf{G} \stackrel{s}{\Rightarrow} x \stackrel{t}{\Rightarrow} X^m$. Given that $t \in (\Sigma \setminus \Sigma_{\mathrm{LD}})^*$, it follows that $P_{\Sigma_{\mathrm{LD}}}(t) = \varepsilon$, so

$$\mathbf{T}^{\mathrm{LD}} \stackrel{P_{\Sigma_{\mathrm{LD}}}(s)}{\Longrightarrow} x_T \stackrel{P_{\Sigma_{\mathrm{LD}}}(t)}{\Longrightarrow} x_T \stackrel{\varepsilon}{\Rightarrow} x_T^2 \in X_T^m \ .$$

Therefore, $\mathbf{G} \,\|\, \mathbf{T}^{\mathrm{LD}} \stackrel{s}{\Rightarrow} (x, x_T) \stackrel{t}{\Rightarrow} X^m \times X_T^m$, i.e., $\mathbf{G} \,\|\, \mathbf{T}^{\mathrm{LD}}$ is nonblocking.

Now assume that $\mathbf{G} \,\|\, \mathbf{T}^{\mathrm{LD}}$ is nonblocking, and let $\mathbf{G} \stackrel{s}{\Rightarrow} x$. Clearly by construction

$$\mathbf{T}^{\mathrm{LD}} \stackrel{P_{\Sigma_{\mathrm{LD}}}(s)}{\Longrightarrow} x_T^1 \stackrel{\tau}{\rightarrow} x_T^2 \ ,$$

and therefore $\mathbf{G} \parallel \mathbf{T}^{\mathrm{LD}} \stackrel{s}{\Rightarrow} (x, x_T^2)$. Since $\mathbf{G} \parallel \mathbf{T}^{\mathrm{LD}}$ is nonblocking, there exists $t \in \Sigma^*$ such that $(x, x_T^2) \stackrel{t}{\Rightarrow} X^m \times X_T^m$. So the string $P_{\Sigma_{\mathrm{LD}}}(t)$ is enabled in state x_T^2 of $\mathbf{T}^{\mathrm{LD}}$, but by construction no event in Σ_{LD} is enabled in this state. Therefore, $P_{\Sigma_{\mathrm{LD}}}(t) = \varepsilon$, or equivalently $t \in (\Sigma \setminus \Sigma_{\mathrm{LD}})^*$. Thus, $\mathbf{G} \stackrel{s}{\Rightarrow} x \stackrel{t}{\Rightarrow} X^m$ for $t \in (\Sigma \setminus \Sigma_{\mathrm{LD}})^*$, i.e., $\mathbf{G}$ is LD-nonblocking. □

5. CONCLUSIONS

This paper proposes a hybrid approach to the modeling and verification of large-scale discrete event systems. Large models are first structured into subsystems according to the principle of Hierarchical Interface-based Supervisory Control (HISC), and then the subsystems are analyzed individually using compositional verification.

This paper shows how the individual conditions to be checked for each HISC subsystem can be verified using compositional verification. The size of the systems that HISC can handle is primarily limited by the size of the largest subsystem. By evaluating the per-subsystem conditions using compositional verification, the explicit construction of the complete state space of each subsystem is avoided, making it possible to analyze larger HISC systems overall.

In addition to the improved model structure provided by HISC, the hybrid approach also has computational advantages over straightforward use of compositional methods. These advantages follow from the compartmentalization into subsystems that HISC provides, and the fact that all properties can be checked by considering only one subsystem at a time. Even though compositional verification may be able to identify subsystems automatically in some cases, this identification often remains a challenge. The user-defined subsystem structure reduces the choice of automata to be composed, making compositional verification much easier or even possible.

Furthermore, the hybrid method provides a means to easily parallelise the method. Each property of each subsystem can be evaluated individually, producing several independent tasks that can be run in parallel.

REFERENCES

Brandin, Bertil A., Robi Malik and Petra Malik (2004). Incremental verification and synthesis of discrete-event systems guided by counter-examples. *IEEE Trans. Contr. Syst. Technol.* **12**(3), 387–401.

Clarke, Jr., Edmund M., Orna Grumberg and Doron A. Peled (1999). *Model Checking*. MIT Press.

de Queiroz, Max H., José E. R. Cury and W. M. Wonham (2004). Multi-tasking supervisory control of discrete-event systems. In: *Proc. 7th Int. Workshop on Discrete Event Systems, WODES '04*. Reims, France. pp. 175–180.

Flordal, Hugo and Robi Malik (2009). Compositional verification in supervisory control. *SIAM J. Control and Optimization* **48**(3), 1914–1938.

Hoare, C. A. R. (1985). *Communicating Sequential Processes*. Prentice-Hall.

Leduc, R. J., Pengcheng Dai and Raoguang Song (2009). Synthesis method for hierarchical interface-based supervisory control. *IEEE Trans. Automat. Contr.* **54**(7), 1548–1560.

Leduc, Ryan J. (2009). Hierarchical interface-based supervisory control with data events. *Int. J. Control* **82**(5), 783–800.

Leduc, Ryan J., Bertil A. Brandin, Mark Lawford and W. M. Wonham (2005*a*). Hierarchical interface-based supervisory control—part I: Serial case. *IEEE Trans. Automat. Contr.* **50**(9), 1322–1335.

Leduc, Ryan J., Mark Lawford and Pengcheng Dai (2006). Hierarchical interface-based supervisory control of a flexible manufacturing system. *IEEE Trans. Contr. Syst. Technol.* **14**(4), 654–668.

Leduc, Ryan J., Mark Lawford and W. M. Wonham (2005*b*). Hierarchical interface-based supervisory control—part II: Parallel case. *IEEE Trans. Automat. Contr.* **50**(9), 1336–1348.

Leduc, Ryan James (2002). Hierarchical Interface-based Supervisory Control. PhD thesis. Dept. of Electrical Engineering, University of Toronto, Ontario, Canada. [Online] Available: `http://www.cas.mcmaster.ca/~leduc`.

Malik, Robi and Ryan Leduc (2008). Generalised nonblocking. In: *Proc. 9th Int. Workshop on Discrete Event Systems, WODES'08*. Göteborg, Sweden. pp. 340–345.

Malik, Robi and Ryan Leduc (2009). A compositional approach for verifying generalised nonblocking. In: *Proc. 7th Int. Conf. Control and Automation, ICCA '09*. Christchurch, New Zealand. pp. 448–453.

Ramadge, Peter J. G. and W. Murray Wonham (1989). The control of discrete event systems. *Proc. IEEE* **77**(1), 81–98.

Ware, Simon and Robi Malik (2008). The use of language projection for compositional verification of discrete event systems. In: *Proc. 9th Int. Workshop on Discrete Event Systems, WODES'08*. Göteborg, Sweden. pp. 322–327.