A Theory of Fault Recovery for Component-based Models*

Borzoo Bonakdarpour[1], Marius Bozga[2], and Gregor Gössler[3]

[1] School of Computer Science, University of Waterloo
Email: borzoo@cs.uwaterloo.ca
[2] VERIMAG/CNRS, Gieres, France
Email: marius.bozga@imag.fr
[3] INRIA-Grenoble, Montbonnot, France
Email: gregor.goessler@inria.fr

Abstract. This paper introduces a theory of *fault recovery* for component-based models. We specify a model in terms of a set of atomic components incrementally composed and synchronized by a set of glue operators. We define what it means for such models to provide a recovery mechanism, so that the model converges to its normal behavior in the presence of faults (e.g., in self-stabilizing systems). We present a sufficient condition for incrementally composing components to obtain models that provide fault recovery. We identify *corrector* components whose presence in a model is essential to guarantee recovery after the occurrence of faults. We also formalize component-based models that effectively *separate* recovery from functional concerns. We also show that any model that provides fault recovery can be transformed into an equivalent model, where functional and recovery tasks are modularized in different components.

Keywords: Fault-tolerance; Transformation; Separation of concerns; BIP

1 Introduction

Fault-tolerance has always been an active line of research in design and implementation of *dependable* systems. Intuitively, tolerating faults involves providing a system with the means to handle unexpected defects, so that the system meets its specification even in the presence of faults. In this context, the notion of specification may vary depending upon the guarantees that the system must deliver in the presence of faults. Such guarantees can be broadly characterized by *safety* and *liveness* properties. For instance, dependable mission-critical systems often employ monitoring or control techniques to ensure safety properties in the presence of faults, and, provide a *recovery* mechanism to meet liveness properties, if the system reaches an unexpected state. *Self-stabilization* is a special type of

[*] This is an extended version of the paper appeared in the *14th International Conference on Safety, Security, and Stabilization of Distributed Systems (SSS'12).*

fault-tolerance (largely concerned with liveness only), where a system always reaches a correct state no matter what state it is initialized with.

The concept of fault-tolerance as described above addresses the overall behavior of the system and is independent of the structure the system. In order to associate fault-tolerance properties with the structure of a system and study their interdependence, one has to focus on a specific methodology. The *component-based* approach is a popular divide-and-conquer technique for designing and implementing large systems as well as for reasoning about their correctness. Ideally, in this approach, a system is designed incrementally by composing smaller components, each responsible for delivering a certain set of tasks to separate different concerns. Thus, component-based design and analysis of fault-tolerant systems is highly desirable in order to achieve systematic modularization of such systems. For instance, fault tolerance is becoming one of the key issues for efficient multi-core programming [10]. The likelihood of fault occurrences is in fact proportional with the number of cores available in the underlying platform. Traditional fault detection and recovery mechanisms e.g., based on restore points and rollback, scale poorly and may even become unusable for many cores. That is, significant amount of core time and power are spent on fault recovery instead of performing useful computation. Such scenarios clarify the need for systematic and modularized approaches for fault recovery in large scale systems.

We believe that we currently lack a formal approach that rigorously relates a component-based methodology with fault-tolerance/self-stabilization concerns. With this motivation, in this paper, we propose a novel formal framework for component-based design and analysis of *non-masking* models [2], where recovery and, hence, liveness is guaranteed in the presence of faults. We use the semantics of the BIP (Behavior, Interaction, Priority) framework [14] to specify components and their composition. In BIP, the *behavior* of an atomic component is specified by a labelled transition system. A model (i.e., a composite component) is represented as the composition of a set of atomic components by using two types of operators: *interactions* describing synchronization constraints between components, and *priorities* to specify scheduling constraints. Given a BIP model, the tool chain can automatically generate a stand-alone, distributed, real-time, multi-threaded, or synchronous C++ implementation that is correct-by-construction (i.e., by preserving functional semantics of the original model) [1, 4, 5, 8]. Thus, our results in this paper can be applied in model-based design an analysis of component-based fault recovery for a wide range of settings such as in distributed systems.

Contributions Our contributions in this paper are the following:

- We formally define non-masking fault-tolerance for atomic and composite components based on their observational behavior. This is different from the approach in [2], where fault-tolerance is defined based on reachability of predicates.
- We present a sufficient condition for *incrementally constructing* non-masking composite components by starting from a set of non-masking atomic components.

- Inspired by the work in [3], we define *corrector* components that establish a desirable observational behavior and show that the necessary condition for a composite component to be non-masking is to contain atomic or composite correctors. We also introduce the notion of *pure correctors* that only exhibit recovery behavior and do not participate in functional tasks of a composite component. We show that models containing pure correctors can effectively separate functional from recovery concerns and, hence, can be compositionally verified.
- Leveraging the separation of concerns supported by pure components, we provide an automated transformation of a component-based model into an equivalent model consisting of pure components whose behaviors are orthogonal: when a normal execution phase is interrupted by the occurrence of faults, control is transferred from the impacted functional components to corrector components in charge of fault handling and recovery, and handed back to the functional components once normal behavior is reestablished.

We note that self-stabilization is equal to non-masking fault-tolerance when faults can perturb execution of a system to any arbitrary state. Thus, all results in this paper can be applied in the context of self-stabilizing systems as well.

Organization In Section 2, we present the preliminary concepts. Section 3 is dedicated to describe our fault model and the notion of fault recovery. Incremental composition of non-masking components is discussed in Section 4, while Section 5 introduces our theory of component-based recovery. Then, in Section 6, we describe separation of recovery and functional concerns. Related work is discussed in Section 7. Finally, we make concluding remarks and discuss future work in Section 8. All proofs appear in the appendix.

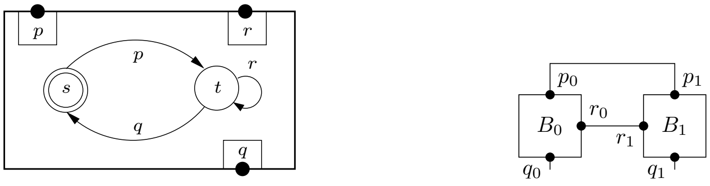
2 Basic Semantic Models of BIP

Atomic Components We define an *atomic component* as a transition system with a set of ports labeling individual transitions. These ports are used for communication between different components.

Definition 1. *An* atomic component B *is a labelled transition system represented by a tuple* $(Q, P, \rightarrow, q^0)$ *where*

- Q *is a set of* states*,*
- P *is a set of* communication ports*,*
- $\rightarrow \subseteq Q \times (P \cup \{\tau\}) \times Q$ *is a set of* transitions *including (1)* observable *transitions labelled by ports, and* unobservable τ *transitions, and*
- $q^0 \in Q$ *is the initial state.*

For any pair of states $q, q' \in Q$ and a port $p \in P \cup \{\tau\}$, we write $q \xrightarrow{p} q'$, iff $(q, p, q') \in \rightarrow$. When the label is irrelevant, we simply write $q \rightarrow q'$. Similarly, $q \xrightarrow{p}$ means that there exists $q' \in Q$, such that $q \xrightarrow{p} q'$. In this case, we say



(a) (b)

Fig. 1. A BIP atomic and composite component.

that p is *enabled* in state q. Figure 1(a) shows an atomic component B, where $Q = \{s, t\}$, $q^0 = s$, $P = \{p, q, r\}$, and $\rightarrow = \{(s, p, t), (t, q, s), (t, r, t)\}$.

A *trace* of a component $B = (Q, P, \rightarrow, q^0)$ is a finite or infinite sequence of ports $\pi = p_0 p_1 p_2 \cdots$, such that for all $i \geq 0$:

1. $p_i \in P \cup \{\tau\}$,
2. there exists state sequence $q_0 q_1 \cdots$, such that:
   - $q_0 = q^0$ (i.e., q_0 is the initial state), and
   - $q_0 \overset{p_0}{\rightarrow} q_1 \overset{p_1}{\rightarrow} q_2 \cdots$

For a trace $\pi = p_1 \cdots p_n$, by $q \xrightarrow{\pi} q'$, we denote $\exists q_1 \cdots q_{n-1} \ : \ q \xrightarrow{p_1} q_1 \xrightarrow{p_2} \cdots \xrightarrow{p_{n-1}} q_{n-1} \xrightarrow{p_n} q'$. The same concept applies for unobservable transitions (e.g., $q \xrightarrow{\tau^* \pi} q'$ is a trace that includes a prefix of τ-transitions and then suffix π).

Interaction For a given system built from a set of m atomic components $\{B_i = (Q_i, P_i, \rightarrow_i, q_i^0)\}_{i=1}^m$, we assume that their respective sets of ports are pairwise disjoint, i.e., for any two $i \neq j$ from $\{1..m\}$, we have $P_i \cap P_j = \emptyset$. We can therefore define the set $P = \bigcup_{i=1}^m P_i$ of all ports in the system. An *interaction* is a set $a \subseteq P$ of ports. When we write $a = \{p_i\}_{i \in I}$, we suppose that for $i \in I$, $p_i \in P_i$, where $I \subseteq \{1..m\}$.

Definition 2. *A* composite component *(or simply* model*) is defined by a composition operator parameterized by a set of interactions* $\gamma \subseteq 2^P$. $B \overset{def}{=} \gamma(B_1 \ldots B_m)$, *is a transition system* $(Q, \gamma, \rightarrow, q^0)$, *where* $Q = \bigotimes_{i=1}^m Q_i$, $q^0 = (q_1^0 \ldots q_m^0)$, *and* $\rightarrow$ *is the least set of transitions satisfying the rule*

$$\frac{a = \{p_i\}_{i \in I} \in \gamma \qquad \forall i \in I : \ q_i \overset{p_i}{\rightarrow}_i q_i' \qquad \forall i \notin I.\ q_i = q_i'}{(q_1, \ldots, q_m) \overset{a}{\rightarrow} (q_1', \ldots, q_m')}$$

In a composite component, τ-transitions do not synchronize but execute in an interleaving fashion.

The inference rule in Definition 2 says that a composite component $B = \gamma(B_1, \ldots, B_m)$ can execute an interaction $a \in \gamma$, iff for each port $p_i \in a$, the corresponding atomic component B_i can execute a transition labelled with p_i; the states of components that do not participate in the interaction stay unchanged.

In general, one can view a model $\gamma(\mathcal{B}_1, \mathcal{B}_2)$, where $\mathcal{B}_1$ and $\mathcal{B}_2$ are two sets of atomic components, as one component whose set of transitions is γ. Thus, $\gamma(\mathcal{B}_1, \mathcal{B}_2)$ denotes the composite component glued by γ, and, γ denotes the set of interactions of this composite component. In practice, atomic components are extended with variables. Transitions and interactions are associated with guards on variables. Also, interactions can transfer data.

Figure 1(b) illustrates a composite component $\gamma(B_0, B_1)$, where both B_0 and B_1 are identical to the component in Figure 1(a) and $\gamma = \{\{p_0, p_1\}, \{r_0, r_1\}, \{q_0\}, \{q_1\}\}$.

Similar to traces of an atomic component, a trace of a composite component $B = \gamma(B_1, \ldots, B_n)$ is a finite or infinite sequence of interactions $a_0 a_1 a_2 \cdots$, such that for all $i \geq 0$ (1) a_i is an interaction of γ, and (2) there exists states $q_0 q_1 \cdots$ of B, such that $q_0 = q^0$ and $q_0 \overset{a_0}{\rightarrow} q_1 \overset{a_1}{\rightarrow} q_2 \cdots$.

3 Fault Model and Fault Recovery

3.1 Fault Model

Let $B = (Q, P, \rightarrow, q^0)$ be an atomic component. We classify the observable transitions in $\rightarrow$ into the following three pairwise disjoint sets:

- A set $\rightarrow_n$ of observable *normal* transitions that embodies the normal execution of the component.
- A set $\rightarrow_f$ of observable *fault* transitions that expresses the faulty behavior of the component.
- A set $\rightarrow_r$ of observable *recovery* transitions that restore the normal behavior of the component or help other components to restore their normal behavior through participating in cross-component interactions.

Finally, $\rightarrow_\tau$ (i.e., τ-transitions of B) is the set of *unobservable fault* transitions and expresses the local faulty behavior of B. Intuitively, a component normally executes transitions in $\rightarrow_n$. However, faults in $\rightarrow_{f,\tau}$ may perturb the state of B to a state that may or may not be reachable by other transitions and in particular, $\rightarrow_n$.

Notation. Let $B = (Q, P, \rightarrow, q^0)$ be an atomic component. By $\rightarrow_x$, we denote the union of transitions of the types in x, where $x \in 2^{\{n,f,\tau,r\}}$. By B_x, we mean the component $(Q, P, \rightarrow_x, q^0)$ induced by transitions in x only.

Definition 3. *We say that $B = (Q, P, \rightarrow, q^0)$ is a* faulty component *if $\rightarrow_{f,\tau}$ is nonempty.*

	n	r	f
n	N	R	F
f	F	F	F
r	R	R	F

Table 1. Interaction types based on the participating transitions

Now, let $B = \gamma(B_1, \ldots, B_m)$ be a composite component. Observe that in an interaction $a = \{p_i\}_{i \in I}$ in γ, for any two $j \neq k$ in $\{1..m\}$, transitions $\xrightarrow{p_k}_k$ and $\xrightarrow{p_j}_j$ may belong to any of the above classes of transitions of their respective components. Thus, we define the type of interactions of a composite component as follows (see Table 1):

- Following Definition 2, an unobservable fault does not participate in an interaction; i.e., the corresponding component only takes a silent move from one state to another without synchronizing with other components.
- If an interaction consists of transitions of the same type, then the interaction type is equivalent to the type of participating transitions.
- Otherwise, the type of the interaction is determined by the greatest type of the participating transitions in the total order $n < r < f$.

Thus, we partition interactions of $B = \gamma(B_1, \ldots, B_n)$ into γ_N, γ_R, and γ_F.

3.2 Fault Recovery

Arora and Gouda [2] formally define the *levels of fault-tolerance* based on combinations of meeting safety and liveness in the presence of faults. In this paper, our focus is on *non-masking* fault-tolerance. Non-masking systems are only concerned with ensuring liveness in the presence of faults by guaranteeing deadlock- and livelock-freedom through providing a finite-step *recovery* mechanism; i.e., the system always eventually reaches a good state even in the presence of faults. However, in such a system, when faults occur, safety may be temporarily violated during recovery, but not after the system reaches a good state.

Non-masking Atomic Components We characterize fault recovery of an atomic component by ω-regular expressions based on the behavior of transition types identified in Subsection 3.1. For example, the ω-regular expression f^*rn^ω is the set of infinite traces of an atomic component where a finite number of observable fault transitions is followed by one recovery transition and an infinite sequence of normal transitions.

Definition 4. *We say that* $B = (Q, P, \to, q^0)$ *is a* non-masking atomic component *iff its set of traces satisfies the* ω*-regular expression* $[n^*((f+\tau)r^*)^*n]^\omega$.

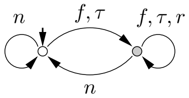


Fig. 2. Non-masking atomic component; the gray state models unstable period.

The intuitive description of Definition 4 is the following (see Figure 2). If no faults occur, the program executes only normal transitions (i.e., the left state in Figure 2). If fault(s) occur, the component reaches a state from where execution of normal transitions is not possible (the gray state in Figure 2). In this case, we say that the component enters a finite *unstable* period (i.e., sub-trace $(f + \tau)r^*$). After a finite number of steps, the component recovers and only executes normal transitions again. Also, note that according to Definition 4 the number of occurrences of faults in each unstable period is finite. Observe that a non-masking component does not exhibit deadlock or livelock in the absence or presence of faults. Also, a non-masking component can use any recovery transition, be it safe or unsafe, to converge to its normal behavior.

Non-masking Composite Components We characterize fault recovery of a composite component based on observational behavior of interaction types identified earlier; i.e., γ_N, γ_F, and γ_R. There is, however, an important difference between non-masking atomic and composite components. In a composite component, if a fault occurs in an atomic component, the fault may force a set of components to execute transitions other than their normal transitions, while a set of other atomic components can resume their normal operation. Thus, unlike non-masking atomic components, non-masking composite component may as well exhibit normal interactions in their unstable period.

Definition 5. *We say that* $B = \gamma(B_0 \cdots B_m)$ *is a* non-masking composite component *iff:*

1. *Its set of traces satisfies the following* ω*-regular expression:*
$$(N^*(F + R + N)^*N)^\omega.$$
2. *If a trace prefix of* B *ends with* NR*, then there exists an atomic component* B_i*,* $0 \leq i \leq m$*, such that projection of the prefix on* B_i *results in a local prefix that ends with* $n\tau^+$*.*

Intuitively, in Definition 5, traces of a non-masking composite component behave similarly to those of non-masking atomic components, except that normal interactions can also occur during the unstable period. Moreover, in a non-masking composite component if a recovery interaction occurs immediately after a normal interaction, then we require the existence of an atomic component in which an unobservable fault causes the execution of the recovery interaction.

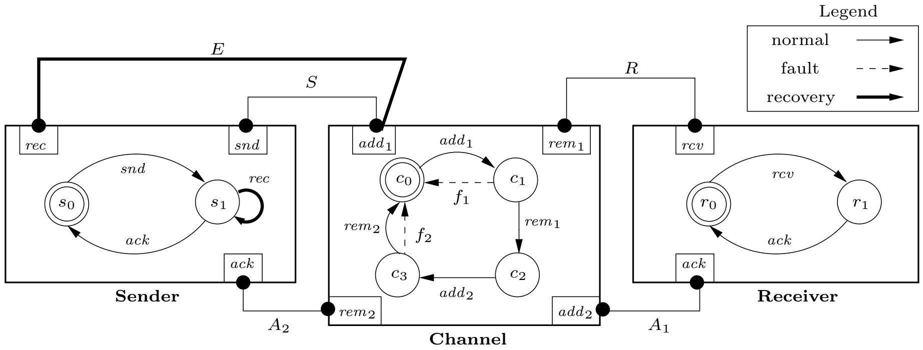


Fig. 3. A simple non-masking communication protocol.

Notice that in Definition 5, we do not require that atomic components of a non-masking composite component should be non-masking as well. This is because we would like our definition to cover cases where an atomic component is not subject to faults locally, but it participates in recovery interactions in the composite component that contains other faulty atomic components.

3.3 Example

Figure 3 illustrates a component-based non-masking communication protocol. The behavior of the model is as follows. The component Sender sends a packet via port *snd* and receives the corresponding acknowledgement through port *ack*. Likewise, Receiver receives the sent packet through port *rcv* and sends an acknowledgement through port *ack*. By each transmission, component Channel adds an item to its single-space buffer (through ports add_1 and add_2) and by each delivery, the item is removed (via ports rem_1 and rem_2). Our channel is lossy and faults cause loss of the sent packet (i.e., transition f_1) or the acknowledgement (i.e., transition f_2). Both faults are unobservable faults (i.e., f_1 and f_2 are τ-transitions). Recovery involves re-transmitting the packet through the *rec* port in Sender. Thus, the classification of transitions is as follows:

- Sender: $\rightarrow_n = \{s_0 \xrightarrow{snd} s_1, s_1 \xrightarrow{ack} s_0\}$, $\rightarrow_f = \rightarrow_\tau = \emptyset$, $\rightarrow_r = \{s_1 \xrightarrow{rec} s_1\}$.
- Receiver: $\rightarrow_n = \{r_0 \xrightarrow{rcv} r_1, r_1 \xrightarrow{ack} r_0\}$, $\rightarrow_f = \rightarrow_\tau = \rightarrow_r = \emptyset$.
- Channel: $\rightarrow_n = \{c_0 \xrightarrow{add_1} c_1, c_1 \xrightarrow{rem_1} c_2, c_2 \xrightarrow{add_2} c_3, c_3 \xrightarrow{rem_2} c_0\}$, $\rightarrow_f = \rightarrow_r = \emptyset$, $\rightarrow_\tau = \{c_1 \longrightarrow c_0, c_3 \longrightarrow c_0\}$.

In the composite component $\gamma(\text{Sender}, \text{Channel}, \text{Receiver})$, interactions $\gamma = \{S, R, E, A_1, A_2\}$ synchronize the atomic components as follows. A transmission by Sender or Receiver is synchronized with adding the item to the buffer of Channel (i.e., interactions S and A_1, respectively). Likewise, delivery of the item to Sender or Receiver is synchronized with its removal by Channel (i.e., interactions A_2 and R, respectively). The recovery interaction E ensures re-transmission of the message if a fault occurs. Thus, we have: $\gamma_N = \{S, R, A_1, A_2\}$, $\gamma_R = \{E\}$, and $\gamma_F = \emptyset$. In the absence of faults the set of traces of the composite component satisfies expression: $(SRA_1A_2)^\omega$. In the presence of faults, one possible characterization of the model is the set of traces: $(SE^*RA_1(E^+RA_1)^*A_2)^\omega$.

Notice that recovery interaction E occurs after normal interactions S or A_1 only if a fault occurs in Channel. Also, although the model is non-masking, atomic component Sender is not non-masking, as it has traces with prefix $(snd.ack)^*.snd.rec$; i.e., Sender exhibits a recovery transition although no local fault has occurred. Another interesting observation in this example is that although all faults occur in component Channel, this component does not contain any recovery transitions. In fact, the only way for Channel to recover after the occurrence of a fault is by getting assistance from component Sender.

4 Incremental Construction of Non-masking Models

First, note that composing a set of non-masking atomic components does not necessarily result in obtaining a non-masking composite component. In this section, we show that a composition that *preserves* non-masking properties of participating atomic components obtains a non-masking composite component, provided at least one component executes normal transitions for a *long enough period.* By preserving non-masking properties, we mean if one projects the set of traces of the composite component onto individual atomic components, the projected trace satisfies Definition 4. By 'long enough period', we mean that the unstable period of two components do not overlap with each other indefinitely. One characterization of the aforementioned constraints are presented in Theorem 1.

Theorem 1. *Let B_1 and B_2 be two non-masking atomic components and $B = \gamma(B_1, B_2)$ be a composite component. If*

1. *composition γ preserves the non-masking properties of B_1 and B_2, and*
2. *at least one component exhibits only one period of instability,*

then B is non-masking as well.

We note that although the assumption of having only one unstable period in Theorem 1 imposes a strong constraint, it is not unrealistic. This is due to the fact that in most commonly considered systems, if the frequency of occurrence of faults is low, all components have enough time to recover and, hence, the entire system recovers to a global good state. This is precisely our intention for adding the assumption. On the contrary, if the frequency of occurrence of faults is high, then most components spend most of their time in recovery and, hence, the entire system is not likely to reach a global good state nonetheless.

Incremental construction of non-masking models as prescribed in Theorem 1 imposes another strong restriction, i.e., preserving non-masking behavior of atomic components. A more relaxed approach is to somehow build a non-masking model, so that atomic components are not required to preserve their non-masking properties. Such an approach would be more involved and in particular requires reachability analysis of each component and their composition. This issue is outside the scope of this paper.

5 Correctors and Component-based Recovery

5.1 Correctors

The concept of correctors is inspired by the work in [3, 7]. The definition of correctors in [3,7] is based on correction of an invariant predicate, no matter how it is reached. Our definition of correctors in this paper is based on observation of recovery and normal transitions/interactions in atomic/composite components. In other words, our notion of correctors is tailored for component-based models.

Roughly speaking, a corrector is concerned with two types of transitions: recovery and normal. A corrector component ensures two properties: (1) once a fault occurs, the component somehow recovers and eventually exhibits normal behavior (i.e., recovery results in restoring the normal behavior), and (2) execution of normal transitions eventually stabilizes (i.e., once normal behavior is restored the component behaves normally unless another fault occurs).

Definition 6. *Let $B = (Q, P, \rightarrow, q^0)$ be an atomic component. We say that B is a* corrector *for the set $\rightarrow_n$ of normal transitions, if there exists the set $\rightarrow_r$ of recovery transitions, such that $\rightarrow_n \cap \rightarrow_r = \emptyset$ and any trace $\pi = p_0p_1\cdots$, where $p_i \in P$, satisfies the following two conditions:*

1. (Progress) *If there exists $i \geq 0$, such that transition $q_i \xrightarrow{p_i} q_{i+1}$ is not in $\rightarrow_{r,n}$, then there exists $j \geq i+1$, such that $q_j \xrightarrow{p_j} q_{j+1}$ is in $\rightarrow_n$.*
2. (Weak Stability) *For all $i \geq 0$, if $q_i \xrightarrow{p_i} q_{i+1}$ is in $\rightarrow_n$, then $q_{i+1} \xrightarrow{p_{i+1}} q_{i+2}$ is either (1) in $\rightarrow_n$, or (2) not in $\rightarrow_{r,n}$.*

A *composite corrector component* is defined in the same fashion for interactions of types R and N. A composite component may be a corrector for a set of transitions local to one of its atomic components. Such correctors are of interest where a faulty component achieves recovery to its normal behavior by the help of a set of other components. The model presented in Subsection 3.3 is an example of such correctors.

Formally, let $B = \gamma(B_0 \cdots B_m)$ be a composite component and $B_i = (Q_i, P_i, \rightarrow_i, q_i^0)$, $0 \leq i \leq m$, be an atomic component. We say that B is a corrector for the set $\rightarrow_{i_n}$ of normal transitions of B_i if and only if by projecting any trace $\pi = a_0a_1\cdots$, where $a_j \in \gamma$ for all j, on component B_i and obtaining trace π', there exists recovery transitions $\rightarrow_{i_r}$, such that $\rightarrow_{i_r}$ and $\rightarrow_{i_n}$ satisfy Progress and Weak Stability.

In our example in Figure 3, component Channel is faulty and if fault f_1 or f_2 occurs the whole model (without recovery interactions) deadlocks. Component

Sender provides the recovery mechanism, when a fault occurs. It is straightforward to observe that the composite component $\gamma(\text{Channel}, \text{Sender})$ acts as a corrector in the model for normal interactions of $\gamma_N(\mathit{Sender}, \mathit{Channel})$ (γ is the set of interactions identified in Subsection 3.3), where $\gamma_R(\mathit{Sender}, \mathit{Channel}) = \{E\}$. Observe that our model allows delivery of duplicate messages, which may be considered as violation of safety. However, this is not an issue, since by definition, a non-masking model allows temporary violation of safety while recovering in the presence of faults. Observe that when the model recovers to its normal behavior, each packet is delivered only once.

5.2 Containment of Correctors in Non-masking Models

In this subsection, we show that the necessary condition for a model to be non-masking is to contain a subset of components that act a corrector for each components that is subject to faults. Recall that in Definition 5, we allowed components that do not interact with a faulty component to continue their normal behavior, while interacting components with the faulty component recover. We note that in our model, fault propagation is possible in the sense that components that do not interact with a faulty component may get involved in achieving recovery as well. In order to ensure that recovery makes progress in non-masking models, we assume that composite components are *weakly fair.*

Assumption 1 *Let $B = \gamma(B_0 \cdots B_m)$ be a composite component. We assume that if an interaction $\alpha \in \gamma$ is continuously enabled in a trace $\pi = a_0 a_1 \cdots$, then there exists $i \geq 0$, such that $a_i = \alpha$.*

Assumption 1 is necessary to show containment of correctors in non-masking models. The containment theorem is the following.

Theorem 2. *Let $B = \gamma(B_0 \cdots B_m)$ be a non-masking composite component. For each faulty atomic component $B_l = (Q_l, P_l, \rightarrow_l, q_l^0)$, where $0 \leq l \leq m$, there exists a set $\mathcal{C}$ of atomic components, such that $\mathcal{C} \subseteq \{B_0 \cdots B_m\}$ and $\gamma(B_l, \mathcal{C})$ is a corrector for $\gamma_N(B_l, \mathcal{C})$.*

For example, in Figure 3, one obtains the composite corrector $\gamma(\text{Channel}, \text{Sender})$.

6 Separation of Functional and Recovery Concerns

In Subsection 6.1, we formally define the concept of pure correctors and discuss their role in a model that contains them. In Subsection 6.2, we show that any non-masking model can be transformed into another model that is observationally equivalent to the initial model and only contains pure components and, hence, separates functional from recovery concerns.

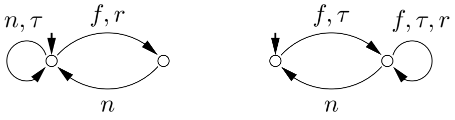


Fig. 4. Pure functional component (left) and corrector (right).

6.1 Pure Components and their Role in Models

Roughly speaking, a *purely functional component* is one that is responsible for performing normal computational tasks of the containing composite component. Such a component may be subject to faults, but is not concerned with achieving fault recovery. On the contrary, a *pure corrector* is a component that only helps a system restoring the normal behavior through achieving recovery and it does not perform any functional tasks.

Definition 7. *Let* $B = (Q, P, \rightarrow, q^0)$ *be an atomic component. We say that* B *is* purely functional *iff its set of traces satisfies the* ω*-regular expression:*

$$((n + \tau)^*(f + r)n)^\omega.$$

Intuitively, in a purely functional component a sequence of normal and unobservable fault transitions may occur (see also the left automaton in Figure 4). Then, the component executes one fault or recovery transition (normally in order to synchronize with a corrector) and reach normal behavior. Obviously, if no fault occurs, a purely functional component continues executing normal transitions.

Definition 8. *Let* $B = (Q, P, \rightarrow, q^0)$ *be an atomic component. We say that* B *is a* pure corrector *for the set* $\rightarrow_n$ *of normal transitions, iff*

1. *B is a corrector for* $\rightarrow_n$*.*
2. (Strong Stability) *For any trace* $\pi = p_0 p_1 \cdots$ *of component* B*, for all* $i \geq 0$*, if* $q_i \xrightarrow{p_i} q_{i+1}$ *is in* $\rightarrow_n$*, then* $q_{i+1} \xrightarrow{p_{i+1}} q_{i+2}$ *is not in* $\rightarrow_{n,r}$*.*

Notice that in a pure corrector when a normal transition is executed, it does not execute any more normal transitions (see also the right automaton in Figure 4). This intuitively means that this normal transition marks the completion of recovery and the pure corrector stops working unless another fault occurs. Thus, we require that this normal transition synchronizes with some normal or recovery transition (normally a purely functional component) in the composite component. The left state of the functional component models periods of normal behavior or where no fault has been detected yet; the right state models a failure state where the pure functional component is inactive. Symmetrically, the left state of the pure corrector models a period of normal behavior where the corrector is inactive, and the right-hand side stands for an unstable period.

We now show that in the absence of faults, a pure corrector plays no role in the behavior of a model that contains it. In other words, in the absence of faults, the existence of a pure corrector in a model can be overlooked.

Theorem 3. *Let $B = \gamma(B_0 \cdots B_m)$ be a composite component and B_i, $0 \leq i \leq m$, be the one and only pure corrector in B. The set of traces of $\gamma_N(B_0 \cdots B_m)$ and $\gamma(B_{0_n} \cdots B_{i-1_n}, B_{i+1_n} \cdots B_{m_n})$ are equal.*

A trivial but important consequence of Theorem 3 is that pure correctors do not *interfere* with pure functional components.

Corollary 1. *Let $B = \gamma(B_0 \cdots B_m)$ be a composite component and $B_i = (Q_i, P_i, \rightarrow_i, q_i^0)$, $0 \leq i \leq m$, be the one and only pure corrector in B. Let $\pi = a_0 a_1 \cdots$ be a trace of B. If for all $j \geq 0$, $a_j \in \gamma_N$, then no interaction in π involves a port in P_i.*

The other side of the coin is that when a fault occurs in a purely functional faulty component, it stops working until recovery from the fault is complete.

Theorem 4. *Let $B = \gamma(B_0 \cdots B_m)$ be a composite component and B_i, $0 \leq i \leq m$, be the one and only purely functional atomic component in B. The set of traces of $\gamma_R(B_0 \cdots B_m)$ and $\gamma(B_{0_r} \cdots B_{i-1_r}, B_{i+1_r} \cdots B_{m_r})$ are equal.*

An immediate application of Corollary 1 and Theorem 4 is in compositional analysis of fault-tolerant systems. For instance, in order to verify the correctness of functional (respectively, recovery) properties of a non-masking composite component, one can simply remove pure correctors (respectively, functional components) from the model and verify the remaining composite component with respect to functional (respectively, recovery) properties. Such decomposition clearly assists in reducing the size of state space in the context of model checking. In the context of monolithic programs represented in terms of guarded commands in the shared memory model, identifying correctors has shown to be effective in significantly reducing the cost of model checking [6]. However, as mentioned in Section 7, dealing with decomposition of monolithic models is not as straightforward as the same task in our component-based model in this paper.

6.2 Transforming a Non-masking Model to one that Only Contains Pure Components

The goal of this section is to show that any non-masking model can be transformed into another model that behaves *equivalently*, but ensures separation of concerns by only containing pure components. To this end, we provide an algorithm that automatically transforms a non-masking component $B = \gamma(B_1, ..., B_n)$ into a non-masking component $B' = \gamma'\big(f(B_1), ...f(B_n), c(B_1), ..., c(B_n)\big)$, such that all $f(B_i)$ (resp. $c(B_i)$) are purely functional (resp. pure corrector) components, and the behaviors of B and B' are related by a form of *bisimulation*. defined next.

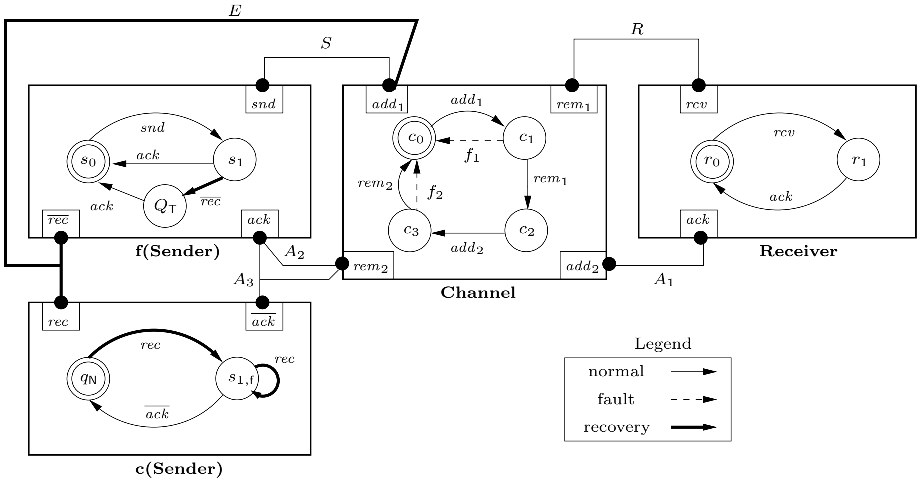


Fig. 5. Transformation applied to the communication protocol in Figure 3.

Definition 9 **($\simeq$).** *Let $B_i = (Q_i, \gamma_i, \rightarrow_i, q_i^0)$ with $\gamma_i \subseteq 2^{P_i}$, $i = 1, 2$. We define $\simeq \subseteq Q_1 \times Q_2$ as the largest relation such that*

1. *if $q_1 \simeq q_2$ and $q_1 \xrightarrow{\alpha_1}_1 q_1'$, then $\exists q_2' \in Q_2\ \exists \alpha_2 \in \gamma_2 : q_2 \xrightarrow{\alpha_2}_2 q_2' \wedge q_1' \simeq q_2' \wedge \alpha_1 \cap P_2 = \alpha_2 \cap P_1$; and*
2. *if $q_1 \simeq q_2$ and $q_2 \xrightarrow{\alpha_2}_2 q_2'$, then $\exists q_1' \in Q_1\ \exists \alpha_1 \in \gamma_1 : q_1 \xrightarrow{\alpha_1}_1 q_1' \wedge q_1' \simeq q_2' \wedge \alpha_1 \cap P_2 = \alpha_2 \cap P_1$.*

B_1 and B_2 are equivalent, *written $B_1 \simeq B_2$, if $q_1^0 \simeq q_2^0$.*

Intuitively, the transformation $\mathcal{T}r$ decomposes the behavior of each atomic component B_i into its normal sub-behavior and its unstable sub-behavior. A pure functional component $f(B_i)$ is then obtained by replacing the unstable behavior by a single state q^{T} that is reached by the first fault or recovery transition after a normal execution phase, and left again by the first normal transition after the unstable phase, as in Figure 4 (left). Similarly, a pure corrector $c(B_i)$ is obtained by replacing the normal behavior of B_i with a single state q^{N}, such that the obtained behavior refines Figure 4 (right). Both $f(B_i)$ and $c(B_i)$ interact on the transitions from and to q^{T} and q^{N} in such a way that the control is handed from $f(B_i)$ to $c(B_i)$ at the beginning of an unstable phase, and back to $f(B_i)$ again at the first normal transition.

Theorem 5. *If B is an atomic component, then $\gamma_B\big(f(B), c(B)\big) \simeq B$. If B is a composite component, then $\mathcal{T}r(B) \simeq B$.*

An immediate implication of Theorem 5 is that the output of our transformation results in a non-masking model.

Corollary 2. *If B is non-masking, then $\mathcal{T}r(B)$ is non-masking as well.*

Example 1. Applied to the communication protocol of Figure 3, we obtain the transformed protocol shown in Figure 5. In $f(Sender)$, q^{T} represents the unstable part of the behavior during which $c(Sender)$ has control. Conversely, during normal behavior $c(Sender)$ is in state q^{N} and inactive until the recovery interaction $\{rec, \overline{rec}, add_1\}$ is enabled. Maximal progress ensures that interaction $\{ack, rem_2\}$ is disabled whenever interaction $\{ack, \overline{ack}, rem_2\}$ is enabled.

In Figure 5, $f(Sender)$ is a purely functional component and $c(Sender)$ is a pure corrector. Since the original protocol is non-masking, the transformed protocol is non-masking by construction.

7 Related Work

Component-based analysis of fault-tolerant untimed models was first studied by Arora and Kulkarni [3]. They show that a fault-tolerant program that satisfies safety and liveness properties in the presence of faults can be decomposed into a fault-intolerant program and a set of components called detectors and correctors. Detectors ensure satisfaction of safety and correctors guarantee satisfaction of liveness properties in the presence of faults. In their work, a program is represented as a set of guarded commands in the shared memory model. Moreover, a detector (resp. corrector) component is defined based on state predicate detection (resp. correction) properties that a set of computations meets. In other words, unlike the results in this paper, the notion of a component in [3] does not resemble normal software modules, each having their own state space, behavior, and interface. The work in [3] is extended to the context of real-time systems by Bonakdarpour, Kulkarni, and Arora [7] and is enriched by introducing non-interference rules for compositional model checking in [6]. Based on this line of work, in [15], the authors propose a method for constructing fault-tolerant systems by starting from an intolerant version. However, their technique does not take into account the explicit structure of components in the way this paper does. Ebnenasir and Cheng [11] study design issues in systems that exhibit error recovery in the presence of faults from a software engineering point of view. In particular, they propose a corrector design pattern.

In [12], the authors propose a formal component model that incorporates the notion of a *safety* interface. This work is fundamentally different from our work in that we focus on recovery which implies guaranteeing liveness in the presence of faults. Lui and Joseph [18–20] introduce a uniform framework for specifying, refining, and transforming programs that provide fault-tolerance and schedulability using the Temporal Logic of Actions [16]. Our work is different from [20] in that we focus on the structure and analysis of *component-based* programs that provide fault recovery. In particular, our transformation is fundamentally different in that we propose a method to separate fault recovery from functional properties. A survey of similar methods on monolithic systems is presented in [13]. Leal and Arora [17] describe a compositional approach to ensure stabilization. The approach relies on an acyclic dependency relation between components, which is a more high-level (less fine-grained) approach compared to ours. Finally, the approach proposed by Brukman and Dolev [9] is also more high-level than ours, where they introduce a generic proof scheme for recovery-oriented programming.

8 Conclusion

In this paper, we proposed a generic formal framework for specifying and reasoning about fault recovery (also called *non-masking* fault-tolerance) for component-based models. We characterized component-based models based on the BIP (Behavior, Interaction, Priority) framework [14]. However, our method is not limited to BIP. Unlike the approaches in [3,7,16,18–20] where a monolithic model is analyzed or components are defined in terms of properties of sets of computations, our method is based on observational behavior of a model in the presence of faults. Also, we use explicit components, each having its own private state space and behavior. We presented a sufficient condition for incrementally constructing non-masking models. We defined what it means for a component to be a *corrector* and showed that non-masking models must contain corrector components. These components correct the observational behavior of a faulty model and we illustrated they can be constructed as stand-alone components interacting with components that provide functional tasks. We described the application of this result in compositional model checking. Moreover, we illustrated that any non-masking model can be transformed into an equivalent model, where functional and recovery tasks are modularized in different components.

We plan to incorporate the results in this paper in our work on automated derivation of distributed implementation from BIP models [5], where fault-tolerance plays an important role. An interesting future research direction is developing methods that transform an arbitrary non-masking model into a well-structured model, where all atomic components are non-masking. Another open problem is to develop an algorithm that transforms an arbitrary non-masking model into one where recovery is achieved locally in each atomic component (i.e., each atomic component is its own corrector).

9 Acknowledgement

This work is partially sponsored by Canada NSERC Discovery Grant 418396-2012.

References

1. T. Abdellatif, J. Combaz, and J. Sifakis. Model-based implementation of real-time applications. In *ACM International Conference on Embedded Software (EMSOFT)*, pages 229–238, 2010.

2. A. Arora and M. G. Gouda. Closure and convergence: A foundation of fault-tolerant computing. *IEEE Transactions on Software Engineering*, 19(11):1015–1027, 1993.
3. A. Arora and S. S. Kulkarni. Detectors and correctors: A theory of fault-tolerance components. In *International Conference on Distributed Computing Systems (ICDCS)*, pages 436–443, 1998.
4. A. Basu, B. Bonakdarpour, M. Bozga, and J. Sifakis. Systematic correct construction of self-stabilizing systems: A case study. In *International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 4–18, 2010.
5. B. Bonakdarpour, M. Bozga, M. Jaber, J. Quilbeuf, and J. Sifakis. A framework for automated distributed implementation of component-based models. *Springer Journal on Distributed Computing (DC)*, 2012. To appear.
6. B. Bonakdarpour and S. S. Kulkarni. Compositional verification of real-time fault-tolerant programs. In *ACM International Conference on Embedded Software (EMSOFT)*, pages 29–38, 2009.
7. B. Bonakdarpour, S. S. Kulkarni, and A. Arora. Disassembling real-time fault-tolerant programs. In *ACM International Conference on Embedded Software (EMSOFT)*, pages 169–178, 2008.
8. M. Bozga, V. Sfyrla, and J. Sifakis. Modeling synchronous systems in BIP. In *ACM International Conference on Embedded Software (EMSOFT)*, pages 77–86, 2009.
9. O. Brukman and S. Dolev. Recovery oriented programming: runtime monitoring of safety and liveness. *Springer Journal in Software Tools for Technology Transfer (STTT)*, 13(4):377–395, 2011.
10. F. Cappello, A. Geist, B. Gropp, L. Kale, B. Kramer, and M. Snir. Toward exascale resilience. *Journal of High Performance Computing Applications*, 23:374–388, November 2009.
11. A. Ebnenasir and B. H. C Cheng. *Architecting Dependable Systems IV*, chapter A Pattern-Based Approach for Modeling and Analyzing Error Recovery, pages 115–141. Springer Berlin / Heidelberg, 2007.
12. J. Elmqvist, S. Nadjm-tehrani, and M. Minea. Safety interfaces for component-based systems. In *Computer Safety, Reliability, and Security (SAFECOMP)*, pages 246–260, 2005.
13. F. C. Gärtner. Transformational approaches to the specification and verification of fault-tolerant systems: Formal background and classification. *Journal of Universal Computer Science*, 5(10):668–692, 1999.
14. G. Gössler and J. Sifakis. Composition for component-based modeling. *Sci. Comput. Program.*, 55(1-3):161–183, 2005.
15. R. D. Jeffords, C. L. Heitmeyer, M. Archer, and E. I. Leonard. Model-based construction and verification of critical systems using composition and partial refinement. *Formal Methods in System Design (FMSD)*, 37(2-3):265–294, 2010.
16. L. Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 16:872–923, May 1994.
17. W. Leal and A. Arora. Scalable self-stabilization via composition. In *Distributed Computing Systems (ICDCS)*, pages 12–21, 2004.
18. Z. Liu and M. Joseph. Transformation of programs for fault-tolerance. *Formal Aspects of Computing*, 4(5):442–469, 1992.
19. Z. Liu and M. Joseph. Specification and verification of recovery in asynchronous communicating systems. In *Formal techniques in real-time and fault-tolerant systems (FTRTFT)*, pages 137–163, 1993.

20. Z. Liu and M. Joseph. Specification and verification of fault-tolerance, timing, and scheduling. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 21(1):46–89, 1999.

A Proofs of Theorems 1-4

Theorem 1. Let B_1 and B_2 be two non-masking atomic components and $B = \gamma(B_1, B_2)$ be a composite component. If

1. composition γ preserves the non-masking properties of B_1 and B_2, and
2. at least one component exhibits only one period of instability,

then B is non-masking as well.

Proof. Let $\Pi = a_0a_1a_2\cdots$ be a trace of B. Let $\pi^1 = p^1_0p^1_1p^1_2\cdots$ and $\pi^2 = p^2_0p^2_1p^2_2\cdots$ be projections of trace Π on components B_1 and B_2. Since B_1 and B_2 are non-masking, π_1 and π_2 satisfy the ω-regular expression $[n^*((f+\tau)^+r^*)^*n]^\omega$. (cf. Definition 4). If π^1 and π^2 are both of the form n^ω, Π is of the form N^ω. For the case where both components enter an unstable period, since both π^1 and π^2 have prefixes of the form n^*, then Π has a prefix N^* as well. Moreover, since both π^1 and π^2 have suffixes of the form n^ω, then Π has a suffix N^ω. Now, if there exists $i \geq 0$ such that p^1_i or p^2_i correspond to a fault transition, then interaction a_i is a fault interaction. Furthermore, if there exists $i \geq 0$ such that p^1_i corresponds to a recovery transitions and p^2_i corresponds to a normal or recovery transition (or vice versa), then interaction a_i is a recovery interaction. Hence, B exhibits a sub-trace of the form $(F + N + R)^*$. Finally, occurrence of an unobservable fault in one component may result in observing an immediate recovery transition (and, hence, an interaction), which in turn validates the second constraint of Definition 5.

Theorem 2. Let $B = \gamma(B_0 \cdots B_m)$ be a non-masking composite component. For each faulty atomic component $B_l = (Q_l, P_l, \rightarrow_l, q^0_l)$, where $0 \leq l \leq m$, there exists a set $\mathcal{C}$ of atomic components, such that $\mathcal{C} \subseteq \{B_0 \cdots B_m\}$ and $\gamma(B_l, \mathcal{C})$ is a corrector for $\gamma_N(B_l, \mathcal{C})$.

Proof. First, if no fault occurs, traces of B consists of only normal interactions. Consequently, all components execute their normal transitions. Such a trace satisfies Progress and Weak Stability. Now, without loss of generality, we assume that there exists one and only one faulty component B_l. Our instantiations to show the existence of a corrector as defined in Definition 6 are as follows. Let $\mathcal{C} = \{B_0 \cdots B_m\}$. Observe that $\gamma(B_l, \mathcal{C}) = B$ and, hence, $\gamma_N(B_l, \mathcal{C}) = \gamma_N$. In order to show that B is a corrector for γ_N, we show that γ_R and γ_N satisfy Progress and Weak Stability. Let $\pi = a_0a_1\cdots$ be a trace of B:

- *(Progress)* Consider the case where there exists a_i in π, such that $q_i \xrightarrow{a_i} q_{i+1}$ is in γ_F. Since B is non-masking it enters its unstable period and has to start recovering at some point in π. Moreover, since B is non-masking it eventually a normal interaction will be enabled and by Assumption 1, there exists $j \geq i+1$ in π, such that only a normal interaction can be executed. This means that there are no recovery interactions enabled in B. Hence, Progress holds.

- *(Weak Stability)* Suppose that there exists a_i in π, such that $q_i \xrightarrow{a_i} q_{i+1}$ is in γ_N. To prove Weak Stability, we distinguish the following three cases based on the possible behaviors of a non-masking composite component:
  1. If $q_{i+1} \xrightarrow{a_{i+1}} q_{i+2}$ is in γ_N, then the first condition of Weak Stability on preserving execution of witness interaction holds.
  2. If $q_{i+1} \xrightarrow{a_{i+1}} q_{i+2}$ is in γ_F, then this transition is neither a normal nor a recovery interaction. Thus, the second condition of Weak Stability is met.

Theorem 3. Let $B = \gamma(B_0 \cdots B_m)$ be a composite component and B_i, $0 \leq i \leq m$, be the one and only pure corrector in B. The set of traces of $\gamma_N(B_0 \cdots B_m)$ and $\gamma(B_{0_n} \cdots B_{i-1_n}, B_{i+1_n} \cdots B_{m_n})$ are equal.

Proof. We distinguish two cases:

- ($\Rightarrow$) Let $\pi = a_0 a_1 \cdots$ be a trace of $\gamma_N(B_0 \cdots B_m)$. Observe that all interactions in π are normal. Thus, following Table 1, for all $i \geq 0$, the transitions that participate in a_i are all normal transitions as well. It follows that none of the participating transitions belong to B_i. Otherwise, the interaction would not be in γ_N, as transitions of B_i that are of type R and F result in interactions not in γ_N, and, transitions of type n in B_i can only synchronize with transitions of type r of other components (cf. Figure 4). Moreover, trace π is reproducible by $\gamma(B_{0_n} \cdots B_{i-1_n}, B_{i+1_n} \cdots B_{m_n})$, as component B_i plays no role in the sequence of interactions produced by $\gamma_N(B_0 \cdots B_m)$. Hence, π is a trace of $\gamma(B_{0_n} \cdots B_{i-1_n}, B_{i+1_n} \cdots B_{m_n})$ as well.
- ($\Leftarrow$) Now, let $\pi = a_0 a_1 \cdots$ be a trace of $\gamma(B_{0_n} \cdots B_{i-1_n}, B_{i+1_n} \cdots B_{m_n})$. Since all components only participate in interactions in γ with their normal transitions, the resulting interactions are in γ_N. This interaction cannot include the pure corrector B_i, as it would result in an interaction not in γ_N, which would be a contradiction. Similar to the previous case, the trace is re-producible by $\gamma_N(B_0 \cdots B_m)$, as the identical transitions, states, and interactions of $\gamma(B_{0_n} \cdots B_{i-1_n}, B_{i+1_n} \cdots B_{m_n})$ are present in $\gamma_N(B_0 \cdots B_m)$. Hence, π is a trace of $\gamma_N(B_0 \cdots B_m)$ as well.

Theorem 4. Let $B = \gamma(B_0 \cdots B_m)$ be a composite component and B_i, $0 \leq i \leq m$, be the one and only purely functional atomic component in B. The set of traces of $\gamma_R(B_0 \cdots B_m)$ and $\gamma(B_{0_r} \cdots B_{i-1_r}, B_{i+1_r} \cdots B_{m_r})$ are equal.

Proof. The proof is identical to the proof of Theorem 3, by replacing n-transitions with r-transitions and by replacing N-interactions with R-interactions.

B Proof of Existence of Transformation

Definition 10. *Given an atomic component* $(Q, P, \rightarrow, q^0)$ *and* $q \in Q$*, let* $\bullet q = \{a \mid \exists q' : q' \xrightarrow{a} q\}$ *be the set of actions entering* q*. For a set* $Q' \subseteq Q$*, let* $\bullet Q' = \{a \mid \exists q \in Q \setminus Q', \exists q' \in Q' : q \xrightarrow{a} q'\}$ *be the set of actions entering* Q'*.*

Given an atomic component $B = (Q, P, \rightarrow, q^0)$, a functional component $f(B)$ and a corrector component $c(B)$ are obtained as follows.

1. Compute the set of normal states $Q_{\mathsf{N}}(B)$ and the set of transient states $Q_{\mathsf{T}}(B)$ as the least sets such that

$$\begin{aligned} Q_{\mathsf{N}}(B) &= \{q^0_{\mathsf{n}}\} \cup \{q_{\mathsf{n}} \mid \bullet q \cap N \neq \emptyset\} \\ &\cup \{q'_{\mathsf{n}} \mid \exists q_{\mathsf{n}} \in Q_{\mathsf{N}}(B) : q_{\mathsf{n}} \xrightarrow{\tau} q'_{\mathsf{n}}\} \\ Q_{\mathsf{T}}(B) &= \{q_{\mathsf{f}} \mid \bullet q \cap (F \cup R) \neq \emptyset\} \\ &\cup \{q'_{\mathsf{f}} \mid \exists q_{\mathsf{f}} \in Q_{\mathsf{T}}(B) : q_{\mathsf{f}} \xrightarrow{\tau} q'_{\mathsf{f}}\} \end{aligned}$$

   Each state q is renamed into q_{n} (resp. q_{f}) if it is an element of $Q_{\mathsf{N}}(B)$ (resp. $Q_{\mathsf{T}}(B)$). This allows to split a state q that can be reached by both normal and unstable behavior into states q_{n} and q_{f} that are unique to each sub-behavior. The set $Q_{\mathsf{N}}(B)$ is the set of states reached by normal transitions and closed under unobservable faults τ. The set $Q_{\mathsf{T}}(B)$ is the set of states reached by fault or recovery transitions and closed under unobservable faults.

2. Let $f(B)$ and $c(B)$ be the normal and transient part of the behavior:

$$\begin{aligned} f(B) &= (Q_{\mathsf{N}}(B) \cup \{q^{\mathsf{T}}\},\ P_{\mathsf{N}},\ \rightarrow_{\mathsf{N}},\ q^0_{\mathsf{n}}) \\ c(B) &= (Q_{\mathsf{T}}(B) \cup \{q^{\mathsf{N}}\},\ P_{\mathsf{T}},\ \rightarrow_{\mathsf{T}},\ q^{\mathsf{N}}) \end{aligned}$$

where

$$\begin{aligned}
\rightarrow_{\mathsf{N}} = \{(q_{\mathsf{n}}, a, q_{\mathsf{n}}') \mid q_{\mathsf{n}}, q_{\mathsf{n}}' \in Q_{\mathsf{N}}(B) \wedge q \xrightarrow{a} q'\} \\
\cup \{(q_{\mathsf{n}}, \overline{a}, q^{\mathsf{T}}) \mid q_{\mathsf{n}} \in Q_{\mathsf{N}}(B) \wedge \exists q_{\mathsf{f}}' \in Q_{\mathsf{T}}(B) : \\
q \xrightarrow{a} q'\} \\
\cup \{(q^{\mathsf{T}}, a, q_{\mathsf{n}}') \mid q_{\mathsf{n}}' \in Q_{\mathsf{N}}(B) \wedge \exists q_{\mathsf{f}} \in Q_{\mathsf{T}}(B) : \\
q \xrightarrow{a} q'\} \\
\rightarrow_{\mathsf{T}} = \{(q_{\mathsf{f}}, a, q_{\mathsf{f}}') \mid q_{\mathsf{f}}, q_{\mathsf{f}}' \in Q_{\mathsf{T}}(B) \wedge q \xrightarrow{a} q'\} \\
\cup \{(q_{\mathsf{f}}, \overline{a}, q^{\mathsf{N}}) \mid q_{\mathsf{f}} \in Q_{\mathsf{T}}(B) \wedge \exists q_{\mathsf{n}}' \in Q_{\mathsf{N}}(B) : \\
q \xrightarrow{a} q'\} \\
\cup \{(q^{\mathsf{N}}, a, q_{\mathsf{f}}') \mid q_{\mathsf{f}}' \in Q_{\mathsf{T}}(B) \wedge \exists q_{\mathsf{n}} \in Q_{\mathsf{N}}(B) : \\
q \xrightarrow{a} q'\} \\
P_{\mathsf{N}} = \{a \mid \exists q, q' \in Q_{\mathsf{N}}(B) \cup \{q^{\mathsf{T}}\} : q \xrightarrow{a}_{\mathsf{N}} q'\} \\
P_{\mathsf{T}} = \{a \mid \exists q, q' \in Q_{\mathsf{T}}(B) \cup \{q^{\mathsf{N}}\} : q \xrightarrow{a}_{\mathsf{T}} q'\}
\end{aligned}$$

To obtain a pure functional component $f(B)$, the unstable behavior is collapsed into a single state q^{T}; to obtain a pure corrector $c(B)$, the normal behavior is collapsed into a single state q^{N}.

For the sake of simplicity we assume that **(A1)** if $q_1 \xrightarrow{a_1} q_1'$ and $q_2 \xrightarrow{a_2} q_2'$ with $q_1, q_2 \in Q_{\mathsf{N}}(B)$ and $q_1', q_2' \in Q_{\mathsf{T}}(B)$ (or vice versa), then $a_1 = a_2 \implies q_1' = q_2'$. That is, entering transitions have a unique interaction for each target state. This property can be checked and ensured on B by renaming actions.

We define all new actions $\overline{a}$ to be of the same type as a.

3. We define the set of interactions of the composite component formed by $f(B)$ and $c(B)$ as

$$\begin{aligned}
\gamma_B = \big\{\{n\} \mid n \in N\big\} \;\cup\; \big\{\{a\} \mid a \in F \cup R\big\} \\
\cup \big\{\{a, \overline{a}\} \mid a \in \bullet Q_{\mathsf{N}}(B) \cup \bullet Q_{\mathsf{T}}(B)\big\}
\end{aligned}$$

That is, actions entering the sets of states $Q_{\mathsf{N}}(B)$ and $Q_{\mathsf{T}}(B)$ synchronize with their overlined counterpart entering q^{N} and q^{T}, respectively; all other actions interleave. The transformed component is the composite $\gamma_B\big(f(B), c(B)\big)$.

Given a component $B = \gamma(B_1, ..., B_n)$ composed of atomic components $B_i = (Q_i, P_i, \rightarrow_i, q_i^0)$, $i = 1, ..., n$, the composite

$$\mathcal{T}r(B) = \gamma'\big(f(B_1), c(B_1), ..., f(B_n), c(B_n)\big)$$

is obtained from the functional and corrector components by adapting the interaction model such that transitions between normal and transient states of both components synchronize: $\gamma' = \gamma \cup \big\{\alpha \cup \mathit{cmpl}(\alpha) \mid \alpha \in \gamma\big\}$ where

$$\mathit{cmpl}(\alpha) = \big\{\overline{a} \mid \exists i : a \in \alpha \cap \big(\bullet\, Q_{\mathsf{T}}(B_i) \cup \bullet Q_{\mathsf{N}}(B_i)\big)\big\}$$

Lemma 1. *For each atomic component B, $f(B)$ is a purely functional component. If B satisfies (Progress), then $c(B)$ is a pure corrector.*

Proof. Sketch: by construction, the behavior of $f(B)$ is included in the behavior shown in Figure 4 (left), with the left state of Figure 4 (left) simulating $Q_{\mathsf{N}}(B)$, and the right state simulating q^{T}.

Likewise, if B satisfies (Progress), then so does $c(B)$, and the behavior of $c(B)$ is included in the behavior shown in Figure 4 (right), with the left state of Figure 4 (right) simulating q^{N}, and the right state simulating both $Q_{\mathsf{T}}(B)$ and τ-transitions issued from q^{N}.

Theorem 5. If B is an atomic component, then $\gamma_B\big(f(B), c(B)\big) \simeq B$. If B is a composite component, then $\mathcal{T}r(B) \simeq B$.

Proof. Let $\rightarrow_1$ and $\rightarrow_2$ denote the transition relations of B and $\gamma_B\big(f(B), c(B)\big)$, respectively. For $f(B)$ and $c(B)$ we use the same notations as in Subsection 6.2. We use the inductive invariant *inv* characterizing the set of states $Q_{\mathsf{N}} \times (Q_{\mathsf{N}}(B) \times \{q^{\mathsf{N}}\}) \cup Q_{\mathsf{T}}(B) \times (\{q^{\mathsf{T}}\} \times Q_{\mathsf{T}}(B))$ on $\gamma_B\big(f(B), c(B)\big)$. Suppose that $q_1 \simeq q_2$. We have to check two directions.

1. Suppose that $q_1 \xrightarrow{a}_1 q_1'$. We distinguish four cases.
   (a) If $q_1, q_1' \in Q_{\mathsf{N}}(B)$ then $(q_1, q^{\mathsf{N}}) \xrightarrow{a}_2 (q_1', q^{\mathsf{N}})$ and $a \cap P_1 = a \cap P_2 = a$.
   (b) If $q_1, q_1' \in Q_{\mathsf{T}}(B)$ then $(q^{\mathsf{T}}, q_1) \xrightarrow{a}_2 (q^{\mathsf{T}}, q_1')$ and $a \cap P_1 = a \cap P_2 = a$.
   (c) If $q_1 \in Q_{\mathsf{N}}(B)$ and $q_1' \in Q_{\mathsf{T}}(B)$ then $(q_1, q^{\mathsf{N}}) \xrightarrow{a|\overline{a}}_2 (q^{\mathsf{T}}, q')$ and $a \cap P_2 = (a|\overline{a}) \cap P_1 = a$. By $q_1 \xrightarrow{a} q_1'$ and **(A1)** we know that $q' = q_1'$.
   (d) If $q_1 = q^{\mathsf{T}}$ and $q_1' \in Q_{\mathsf{N}}(B)$ then $(q^{\mathsf{T}}, q_1) \xrightarrow{a|\overline{a}}_2 (q', q^{\mathsf{N}})$ and $a \cap P_2 = (a|\overline{a}) \cap P_1 = a$. By $q_1 \xrightarrow{a} q_1'$ and **(A1)** we know that $q' = q_1'$.
2. Suppose that $q_2 \xrightarrow{a}_2 q_2'$. We distinguish four cases again.
   (a) If $q_2 = (q_1, q^{\mathsf{N}})$ and $q_2' = (q_1', q^{\mathsf{N}})$ with $q_1' \in Q_{\mathsf{N}}(B)$ and $a \in N \cup \{\tau\}$ then by definition, $q_1 \xrightarrow{a}_1 q_1'$ and $a \cap P_1 = a \cap P_2 = a$.
   (b) If $q_2 = (q^{\mathsf{T}}, q_1)$ and $q_2' = (q^{\mathsf{T}}, q_1')$ with $q_1' \in Q_{\mathsf{T}}(B)$ and $a \in F \cup R \cup \{\tau\}$ then $q_1 \xrightarrow{a}_1 q_1'$ and $a \cap P_1 = a \cap P_2 = a$.
   (c) If $q_2 = (q_1, q^{\mathsf{N}})$ and $q_2' = (q^{\mathsf{T}}, q_1')$ with $a = b|\bar{b}$ and $b \in \bullet Q_{\mathsf{T}}(B)$ then $q_1 \xrightarrow{b}_1 q_1'$ and $b \cap P_2 = (b|\bar{b}) \cap P_1 = b$.
   (d) If $q_2 = (q^{\mathsf{T}}, q_1)$ and $q_2' = (q_1', q^{\mathsf{N}})$ with $a = b|\bar{b}$ and $b \in \bullet Q_{\mathsf{N}}(B)$ then $q_1 \xrightarrow{b}_1 q_1'$ and $b \cap P_2 = (b|\bar{b}) \cap P_1 = b$.

By definition, $q_1^0 \in Q_{\mathsf{N}}(B)$ and $q_2^0 \in Q_{\mathsf{N}}(B) \times \{q^{\mathsf{N}}\}$. The claim $\gamma_B\big(f(B), c(B)\big) \simeq B$ follows from the fact that $\simeq$ is a greatest fixpoint.

The preservation of non-masking follows from the fact that for any pair of transitions related by bisimulation, both actions have by construction the same type N, F, or R.