# Robustness in Interaction Systems

Mila Majster-Cederbaum and Moritz Martens

University of Mannheim
Mannheim, Germany
mcb@informatik.uni-mannheim.de
mmartens@informatik.uni-mannheim.de

**Abstract.** We treat the effect of absence/failure of ports or components on properties of component-based systems. We do so in the framework of *interaction systems*, a formalism for component-based systems that strictly separates the issues of local behavior and interaction, for which ideas to establish properties of systems were developed. We propose how to adapt these ideas to analyze how the properties behave under absence or failure of certain components or merely some ports of components. We demonstrate our approach for the properties local and global deadlock-freedom as well as liveness and local progress.

## 1 Introduction

Component-based design techniques are an important paradigm for mastering design complexity and enhancing reusability. In the object-oriented approach subsystems interact by invoking in their code operations or methods of other subsystems and hence rely on the availability of these subsystems. In contrast to this, components are designed independently from their context of use. They are put together by some kind of gluing mechanism. This view has lead some authors, e.g. [1,2,3], to consider a component as a black box and to concentrate on the combination of components using a syntactic interface description of the components. However, if we want to make assertions about the behavior of a component system, be it functional, temporal or quantitative, knowledge about the components has to be provided.

There have been approaches using different techniques to model the behavior of a component, e.g. Petri-nets [4], process algebra [5,6] or channel-based methods [7]. Except for model-checking, where the complete global state space has to be analyzed, there are not many approaches that investigate generic properties of systems as deadlock-freedom, liveness, etc. In some previous work [5,8] the question of deadlock-freedom is addressed for special cases.

We build here on *interaction systems*, a model for component-based systems that was proposed and discussed by Sifakis et al. in [9,10,11,12] and has been implemented in the PROMETHEUS [13] as well as the BIP tool [14].

The model strictly separates the description of the components from the way they are glued together. Each component $i$ has a *static* description that gives the information about its interface, which is here modeled by a set $A_i$ *of ports.* The

J. Derrick and J. Vain (Eds.): FORTE 2007, LNCS 4574, pp. 325–340, 2007.

*dynamics* of a component is given by a transition system where the edges are labeled with elements from $A_i$. Components are glued together via *connectors*. A connector is a set of ports which contains at most one port for every component. The connectors give the information how components cooperate. When each component is ready to perform its port in a connector $c$ then all ports in $c$ can be performed conjointly. The same set of components can be glued together differently (i.e. with other connectors) for different applications. The behavior of the global system $Sys$, i.e. the component system, is fully determined by the static and dynamic description of each component and by the connectors. The model is suitable to investigate important properties of component-based systems, as e.g. local/global deadlock-freedom, local progress and liveness. In [15,16,17] it is shown that deciding deadlock-freedom is PSPACE-hard and deciding liveness is NP-hard for interaction systems. However, as the information about the individual components is maintained in the model it can be exploited to develop sufficient conditions for the desired properties that can be tested in polynomial time [18,19,17]. As violations of safety properties can be expressed as deadlocks broad classes of properties can be handled in this approach.

Here we deal with the question of robustness in interaction systems in the following sense. Consider e.g. an interaction system $Sys$ that is deadlock-free, i.e. the system may proceed in every state. Let us now assume that the system has been running for a certain amount of time when a subset $A'$ of the set of all ports becomes unavailable (out of service). This might be because the ports in $A'$ suffer some kind of failure or malfunction but it is also possible to model a situation where certain ports or components are switched off. Can the system $Sys$ still proceed in every state? How are other properties affected? Can a component that could previously make progress in the system still make progress? How do we know if a component is live in $Sys$ when some ports are out of service, etc?

In a first attempt one might try to solve these problems by simply removing the ports in $A'$ from the description of $Sys$ and by then investigating the resulting construct. However, this is not feasible as will be shown later. What we propose to do is to adapt the sufficient conditions and derived algorithms for the desired properties appropriately so that they can be used to answer the questions posed.

Not much work has been done that theoretically investigates the question what effect the failure/absence of parts of a component system has on interesting properties of the system. This is also due to the fact that there is not much work on the theoretical analysis of properties of component-based systems. In [20] component systems are modeled in a way such that they are fault tolerant to a certain extent. This is achieved by requesting that local faulty behavior in a component is detected and handled within the affected component itself. A particular question concerning the classification of safety and liveness in the context of failures has been investigated in [21].

The paper is structured as follows. In Sect. 2 we give a summary of the model of interaction systems. In Sect. 3 we present properties of interaction systems. In Sect. 4 we explain how the sufficient conditions for a desired property can be adapted to the situation where $A'$ is not available. We do so in detail at the hand of global deadlock-freedom of a system and liveness of a set of components. Finally we sketch how local progress and local deadlock-freedom can be treated in a similar way. The paper is summarized by a short conclusion in Sect. 5.

## 2 Components, Connectors and Interaction Systems

In this section we present the basic definitions for interaction systems that were first introduced in [9]. An interaction system models the behavior of a component-based system for a set $K$ of components. It is the superposition of a static model, called interaction model, that considers a component as a black box with interface description and specifies the "glue code", and the dynamic model, which gives the description of the local behavior of the components. For every component $i \in K$, a set $A_i$ of actions or ports is specified and constitutes the interface. Gluing of components is achieved via so-called connectors. A connector $c$ is a finite nonempty set of ports that contains at most one port for every component in $K$. It describes a cooperation of those components which have a port in $c$. When each component is ready to perform its port in $c$ then all ports in $c$ can be performed conjointly. A subset of a connector is called an interaction. We may declare certain interactions to be complete. If an interaction is declared complete it can be performed independently of the environment. It is a design decision which interactions are chosen to be complete. Connectors may be of different sizes and one port may be contained in two or more connectors of different sizes. Thus the model allows for a very flexible way of gluing and consequently of cooperation among components.

**Definition 1 (Interaction Model).** *Let $K$ be the set of* components *and $A_i$ be a* port set *for component $i \in K$ where any two port sets are disjoint. Ports are also referred to as* actions. *A finite nonempty subset $c$ of $A = \bigcup_{i \in K} A_i$ is called a* connector*, if it contains at most one port of each component $i \in K$, that is $|c \cap A_i| \leq 1$ for all $i \in K$. A* connector set *is a set $C$ of connectors that covers all ports and contains only maximal elements:*

$$1.\ \bigcup_{c \in C} c = A \qquad 2.\ c \subseteq c' \Rightarrow c = c' \text{ for all } c, c' \in C.$$

*$I(c)$ denotes the set of all nonempty subsets of connector $c$ and is called the* set of interactions of *$c$ and $I(C) = \bigcup_{c \in C} I(c)$ is the set of interactions of the connector set $C$. For component $i$ and interaction $\alpha \in I(C)$, we put $i(\alpha) = A_i \cap \alpha$. We say that component $i$* participates *in $\alpha$, if $i(\alpha) \neq \emptyset$. Let $Comp \subseteq I(C)$. We call*

$$IM := (C, Comp)$$

*an* interaction model. *The elements of $C$ are also called* maximal interactions *and those of $Comp$ are called* complete interactions.

If not otherwise stated we always assume that $K = \{1, \ldots, n\}$ for some $n \in \mathbb{N}$ or that $K$ is countably infinite. We take up an example from [22].

*Example 1.* We consider a set of tasks $i$ ($i \in K = \{1, ..., n\}$) that compete for some resource in mutual exclusion. Task $i$ is represented by the component $i$ with port set $A_i = \{activate_i, start_i, resume_i, preempt_i, finish_i, reset_i\}$. The connector set is chosen as $C_{tasks} = \{conn_1^i, conn_2^{ij}, conn_3^{ij}, conn_g | i, j \in K, i \neq j\}$, where

$$\begin{aligned} conn_1^i &:= \{activate_i\} \\ conn_2^{ij} &:= \{preempt_i, start_j\} \\ conn_3^{ij} &:= \{resume_i, finish_j\} \\ conn_g &:= \{reset_1, \ldots, reset_n\} \end{aligned}$$

and the complete interactions are given by

$$Comp_{tasks} = \{\{start_j\}, \{finish_j\} | i, j \in K \wedge i \neq j\},$$

and $IM_{tasks} := (C_{tasks}, Comp_{tasks})$.

So far we have only described components as black boxes with ports and have specified the possible structure of cooperation in between them. A further level of description of a component characterizes its local behavior. Basically this can be understood as a control of the way in which a component offers its ports. We assume here that this local behavior of every component $i \in K$ is given by a labeled transition system $T_i$. From the local transition systems and the interaction model we obtain the global behavior of the component-based system.

**Definition 2 (Interaction System).** *Let $K$ be a set of components with associated port sets $\{A_i\}_{i \in K}$ and $IM = (C, Comp)$ an interaction model for it. Let for each component $i \in K$ a transition system $T_i = \left(Q_i, A_i, \rightarrow_i, Q_i^0\right)$ be given where $\rightarrow_i \subseteq Q_i \times A_i \times Q_i$ and $Q_i^0 \subseteq Q_i$ is a non-empty set of* initial states. *We write $q_i \xrightarrow{a_i}_i q_i'$ instead of $(q_i, a_i, q_i') \in \rightarrow_i$.*

*The* induced interaction system *is given by $Sys := \left(IM, \{T_i\}_{i \in K}\right)$ where the* global behavior *$T = \left(Q, C \cup Comp, \rightarrow, Q^0\right)$ is obtained from the local transition systems of the individual components in a straightforward manner:*

1. *The global state space $Q := \prod_{i \in K} Q_i$ is the Cartesian product of the $Q_i$ which we consider to be order independent. We denote states by tuples $q := (q_1, \ldots, q_j, \ldots)$ and call them* (global) states. *Elements of $Q_i$ are called* local states *of component $i$.*
2. *$Q^0 := \prod_{i \in K} Q_i^0$, the Cartesian product of the local initial states. We call the elements of $Q^0$ (global)* initial states.
3. *$\rightarrow \subseteq Q \times (C \cup Comp) \times Q$, the labeled transition relation for Sys defined by*

$$\forall \alpha \in C \cup Comp \; \forall q, q' \in Q : q = (q_1, \ldots, q_j, \ldots) \xrightarrow{\alpha} q' = \left(q_1', \ldots, q_j', \ldots\right) \Leftrightarrow$$

$$\forall i \in K : q_i \xrightarrow{i(\alpha)}_i q_i' \text{ if } i \text{ participates in } \alpha \text{ and } q_i' = q_i \text{ otherwise.}$$

*A state $q_i \in Q_i$ is called* complete *if there is some interaction $\alpha \in C \cup Comp$ and some $q_i'$ such that $q_i \xrightarrow{\alpha}_i q_i'$. Otherwise it is called* incomplete.

Note that a system may proceed in a global state $q$ if $q_i$ is complete for some $i \in K$. The converse does not hold.

**Definition 3 (Enabled).** *Let Sys be an interaction system and let $i \in K$ be a component. For $a_i \in A_i$ we set $en(a_i) := \left\{q_i \in Q_i | \exists q_i' : q_i \xrightarrow{a_i}_i q_i'\right\}$. For $\alpha \in C \cup Comp$ we set $en(\alpha) := \left\{q \in Q | \exists q' : q \xrightarrow{\alpha} q'\right\}$.*

If $q_i \in en(a_i)$ we say that $a_i$ is enabled in $q_i$ or that $q_i$ offers $a_i$ and analogously for $q$ and $\alpha$. Given a set of components, an interaction model $IM = (C, Comp)$ and a transition system $T_i$ for each component $i$ the induced interaction system describes the behavior of the composed system. In particular, in a given global state $q = (q_1, \ldots, q_j, \ldots)$ an interaction $\alpha \in C \cup Comp$ may take place provided that each component $j$ participating in $\alpha$ offers $j(\alpha)$ in $q_j$.

*Example 1 continued.* The transition system $T_i$ for task $i$ is given in Fig. 1 where every local state is a starting state.

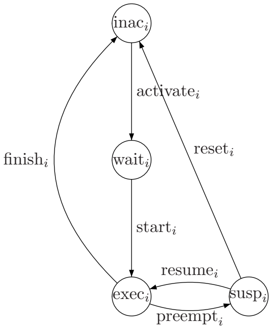

**Fig. 1.** Transition system of task $i$

We put $Sys_{tasks} := (IM_{tasks}, \{T_i\}_{i \in K})$.

*Remark 1.* In what follows, we often mention $Sys = \left(IM, \{T_i\}_{i \in K}\right)$. It is understood that $IM = (C, Comp)$ is an interaction model for the set $K$ of components with port sets $A_i$ and $T_i = \left(Q_i, A_i, \rightarrow_i, Q_i^0\right)$ for $i \in K$ and $T$ are given as above.

## 3 Properties of Interaction Systems

Properties of systems have been classified into safety- and liveness-properties in [23] and have been investigated in various settings, see for example [24,25]. In Sect. 3.1 we define the properties that we consider here w.r.t. absence/failure of ports. The properties are local/global deadlock-freedom, local progress of a set of components and liveness. These properties of interaction systems have been studied in detail in [22,18,19,17,15]. In Sect. 3.2 we define what we mean by robustness.

*Remark 2.* From now on we will assume that the local transition systems have the property that every local state offers at least one action. We also identify singleton sets with their element if it is convenient to do so.

### 3.1 Properties

**Definition 4 (Reachable).** *Let $Sys$ be an interaction system, $q \in Q$. $q$ is* reachable in *$Sys$ if there is a sequence $q^0 \overset{\alpha_0}{\rightarrow} q^1 \overset{\alpha_1}{\rightarrow} \dots \overset{\alpha_{n-1}}{\rightarrow} q$ such that $q^0 \in Q^0$.*

First we take up the notion of local and global deadlock-freedom for interaction systems from [18,22].

**Definition 5 (Local/Global Deadlock-Freedom).** *Let $Sys$ be an interaction system. $Sys$ is called* globally deadlock-free *if for every reachable state $q \in Q$ there exists $\alpha \in C \cup Comp$ such that $q \in en\,(\alpha)$.*

*A nonempty set $K' \subseteq K$* is in local deadlock *in the reachable global state $q$ if for all $i \in K', a_i \in A_i, \alpha \in C \cup Comp$: $(q_i \in en\,(a_i) \wedge a_i \in \alpha)$ implies that there is some $j \in K'$ with $j(\alpha) \neq \emptyset \wedge q_j \notin en(j(\alpha))$. We say that $Sys$* is locally deadlock-free *if there is no reachable state $q$ for which some subset $K' \subseteq K$ is in local deadlock in $q$.*

A subset $K'$ of components is in local deadlock in a reachable global state $q$ if every component $i \in K'$ needs for each of the actions enabled in $q_i$ the cooperation of some component in $j \in K'$ to proceed which in $q_j$ does not offer the action needed. If $K' = K$ we speak of a global deadlock in $q$. In such a state the system is not able to proceed. A system that is globally deadlock-free may still contain local deadlocks. As violations of safety properties can be expressed as deadlocks, the investigation of deadlock-freedom deserves particular attention.

**Definition 6 (Run).** *Let $Sys$ be a globally deadlock-free interaction system, $q \in Q$ a reachable state. A* run *of $Sys$ is an infinite sequence $\sigma = q \overset{\alpha_0}{\rightarrow} q^1 \overset{\alpha_1}{\rightarrow} q^2 \dots$ with $q^l \in Q$ for all $l \in \mathbb{N}$.*

*Let $i \in K$ be a component and let $\sigma$ be a run of $Sys$. If there exists $l$ such that $i$ participates in $\alpha_l$ we say that $i$* participates *in $\sigma$.*

The notions of local progress and liveness of a component have been defined for interaction systems in [22,19].

**Definition 7 (Local Progress and Liveness).** *Let $Sys$ be a globally deadlock-free interaction system and let $K' \subseteq K$ be a nonempty set of components.*

1. *$K'$ can* make local progress *in $Sys$ if for every reachable state $q \in Q$ there exists a run $\sigma = q \overset{\alpha_0}{\rightarrow} q^1 \overset{\alpha_1}{\rightarrow} \dots$ starting in $q$ such that some $i \in K'$ participates in $\sigma$.*
2. *$K'$ is* live *in $Sys$ if for every run $\sigma$ of $Sys$ there is some $i \in K'$ that participates in $\sigma$.*

*Example 1 continued.* In [22] this example was discussed in detail. In particular it was shown that $Sys_{tasks}$ is globally deadlock-free and that every component can make local progress. It was explained that mutual exclusion is achieved under a rule of maximal progress defined in [22].

### 3.2 Robustness of Properties

Let us now assume a situation where a set $A' \subsetneq A$ of ports may become unavailable in a running system. This might be because the ports in $A'$ suffer some kind of failure or malfunction at a certain point of time but it is also possible to model a situation where certain actions or components are switched off for performance reasons for example. We want to formulate what it means that a property is present when $A'$ becomes unavailable. For this we partition $C \cup Comp$ to separate those interactions that involve $A'$ from those that don't.

**Definition 8 (EXCL and WITH).** *Let Sys be an interaction system as above and let $A' \subsetneq A$. We define $EXCL\,(A') := \{\alpha \in C \cup Comp | \alpha \cap A' = \emptyset\}$ and $WITH\,(A') := \{\alpha \in C \cup Comp | \alpha \cap A' \neq \emptyset\}$*

$EXCL\,(A')$ denotes the set of all maximal and complete interactions that do not involve any action from $A'$. Analogously $WITH\,(A')$ is the set of all maximal and complete interactions that involve some action from $A'$.

We consider each of the above properties separately w.r.t. absence of $A'$. Note that it is not possible to just delete the ports of $A'$ from the interaction-system and then check if the definition of a certain property is satisfied by the resulting "system" for two reasons. Firstly, this construct may fail to be an interaction system according to the definition (see Sect. 4), and secondly, the failure of $A'$ may occur at a point of a run where actions from $A'$ may have been previously executed in this run. We discuss deadlock-freedom in terms of robustness which means that we consider a system that is deadlock-free and remains so under failure of $A'$.

**Definition 9 (Robustness of Deadlock-Freedom).** *Let Sys be a globally deadlock-free interaction system and let $A' \subsetneq A$ be a non-empty subset of ports. In Sys global deadlock-freedom is* robust w.r.t. absence of $A'$ *if for every reachable state $q \in Q$ there exists $\alpha \in EXCL\,(A')$ with $q \in en(\alpha)$.*

*Let Sys be locally deadlock-free. In Sys local deadlock-freedom is* not robust w.r.t. absence of $A'$, *if there is some reachable state $q$ and $K'$ such that for any $i \in K'$, for any $a_i$ which is enabled in $q_i$ and for any $\alpha \in EXCL\,(A')$ with $a_i \in \alpha$ there is some $j \in K'$ with $j(\alpha) \neq \emptyset$ and $q_j \notin en(j(\alpha))$. Otherwise local deadlock-freedom is said to be* robust w.r.t. absence of $A'$.

*Remark 3.* In a globally deadlock-free system $Sys$ where $K' \subseteq K$ is live it is not possible that global deadlock-freedom is robust w.r.t. absence of $A' := \bigcup_{i \in K'} A_i$. If this was the case it would be possible to construct a run not letting any component from $K'$ participate which is not possible. The converse does not hold.

We now consider local progress and liveness of a set of components in a system where global deadlock-freedom is robust w.r.t. absence of $A'$. First we need to adapt the notion of a run.

**Definition 10 (Run without $A'$).** *Let Sys be a globally deadlock-free interaction system and $A' \subsetneq A$. Let global deadlock-freedom in Sys be robust with respect to absence of $A'$. Let $q$ be a reachable state.*

*A* run without $A'$ *is an infinite sequence $\sigma = q \xrightarrow{\alpha_0} q^1 \xrightarrow{\alpha_1} \ldots$ with $q^l \in Q, l \geq 1$, and $\alpha_l \in EXCL\left(A'\right), l \geq 0$.*

In a system where global deadlock-freedom is robust w.r.t. absence of $A' \subsetneq A$ such runs always exist by a simple induction argument.

**Definition 11 (Local Progress and Liveness without $A'$).** *Let Sys be a globally deadlock-free interaction system and let $A' \subsetneq A$. Let global deadlock-freedom in Sys be robust w.r.t. absence of $A'$ and let $K' \subseteq K$ be a nonempty set of components.*

1. *$K'$* can make local progress without participation of $A'$ *if for every reachable state $q \in Q$ there exists a run without $A'$ $\sigma = q \xrightarrow{\alpha_0} q^1 \xrightarrow{\alpha_1} \ldots$ such that some $i \in K'$ participates in $\sigma$.*
2. *$K'$* is live without participation of $A'$ *if for every run without $A'$ $\sigma = q \xrightarrow{\alpha_0} q^1 \xrightarrow{\alpha_1} \ldots$ there is some $i \in K'$ that participates in $\sigma$.*

Note that, in analogy to deadlock-freedom, we could formulate a notion of robustness of the property of local progress. In a system where component $i$ can make local progress we could say that this property is robust w.r.t. absence of $A' \subsetneq A$ if $i$ can make local progress without participation of $A'$. By contrast it does not make sense to consider robustness of liveness. If a set $K'$ of components is live in a system, then for every run $\sigma$ there is a component $i \in K'$ that participates in $\sigma$. This is true in particular for all runs without $A'$. Therefore liveness of $K'$ without $A'$ follows from liveness of $K'$ and robustness of deadlock-freedom w.r.t. $A'$. Nonetheless it is interesting to investigate liveness of $K'$ without participation of $A' \subsetneq A$ because it is possible that certain runs in which $K'$ does not participate infinitely many often are no longer present when the ports from $A'$ are not available any more.

## 4 Testing Robustness

From our results about the PSPACE-hardness of deciding deadlock-freedom [16] and NP-hardness of deciding liveness of a set of components [15,17] it is clear that deciding robustness of deadlock-freedom w.r.t. $A' \subsetneq A$ respectively liveness without $A' \subsetneq A$ is at least as hard. One way to deal with the complexity issue for properties is to establish conditions that ensure a desired property and can be tested more easily, see for example [22,18,19,26]. In this paper we want to explain how one can systematically use such conditions to obtain results in the case of failure of $A'$. One could raise the question why we study robustness instead of applying the definitions and results of [22,18,19] to a suitably modified "interaction system". One could try to do so by simply removing the ports in $A'$ from the components of the interaction system under consideration. This approach does not work for two reasons. Firstly, a thus modified construct is in general no longer an interaction system according to our definition. One of the problems that arise can be seen as follows. Consider e.g. the removal of a port $a_j$ of component $j$. It could be the case that every $c \in C$ containing $a_k$ for some $k \in K$ also contains $a_j$. On removal of $a_j$ the connectors containing $a_j$ have to be removed as well. But then the condition in Definition 1 that every port of $k$ is contained in some connector $c \in C$ is violated. This condition is however crucial in various places and in particular for correctness of the criterion presented in [22]. Secondly, the failure of $A'$ may occur at a point of a run such that actions from $A'$ may have been previously executed in this run. It would not be possible to model this situation in a system with alphabet $A \backslash A'$.

### 4.1 Robustness of Deadlock-Freedom

**Definition 12 (Incomplete States).** *Let Sys be an interaction system and let $i \in K$ be a component. We denote by $inc\,(i) := \{q_i \in Q_i | q_i \text{ is incomplete}\}$ the set of incomplete states of component $i$.*

We obtain a criterion for robustness of global deadlock-freedom by adapting the condition of [22] for global deadlock-freedom of an interaction system. This condition involves a graph $G_{Sys}$. The nonexistence of certain cycles in $G_{Sys}$ guarantees deadlock-freedom. $G_{Sys}$ can be built in time polynomial in $|C \cup Comp|$ and the sum of the sizes of the local transition systems for finite interaction systems.

**Definition 13 (Dependency Graph).** *Let Sys be an interaction system. The* dependency graph *for Sys is a labeled directed graph $G_{Sys} := (K, E)$ where the set of nodes is given by the components of Sys, the set of labels is given by $L := L_1 \cup L_2$ with*

$$L_1 := \{c \in C | \nexists \alpha \in Comp : \alpha \subseteq c\}$$

$$L_2 := \{(c, \alpha) \,| c \in C, \alpha \in Comp \text{ such that } \alpha \subseteq c \wedge \nexists \beta \in Comp : \beta \subsetneq \alpha\},$$

*and the set of edges $E \subseteq V \times L \times V$ is defined as follows:*

1. *For $c \in L_1$ : $(i, c, j) \in E \Leftrightarrow j\,(c) \neq \emptyset \wedge \exists q_i \in en\,(i\,(c)) \cap inc\,(i)$.*
2. *For $(c, \alpha) \in L_2$ : $(i, (c, \alpha), j) \in E \Leftrightarrow j\,(\alpha) \neq \emptyset \wedge \exists q_i \in en\,(i\,(c)) \cap inc\,(i)$.*

*Further we define the* snapshot *of $G_{Sys}$* w.r.t. state *$q = (q_1, q_2, \ldots)$ as $G_{Sys}\,(q) := (K, E\,(q))$ where $E\,(q) \subseteq E$ such that*

1. *For $c \in L_1$ : $(i, c, j) \in E\,(q) \Leftrightarrow j\,(c) \neq \emptyset \wedge q_i \in en\,(i\,(c)) \cap inc\,(i)$.*
2. *For $(c, \alpha) \in L_2$ : $(i, (c, \alpha), j) \in E\,(q) \Leftrightarrow j\,(\alpha) \neq \emptyset \wedge q_i \in en\,(i\,(c)) \cap inc\,(i)$.*

*Let $G_f = (K_f, E_f)$ be a subgraph of $G_{Sys}$. $G_f$ is* successor-closed *if $K_f \neq \emptyset$ and for all $i \in K_f$ and all edges $e = (i, l, j) \in E$ where $l \in L$ and $j \in K$ we have $e \in E_f$ and $j \in K_f$.*

The intuitive meaning of the graph is as follows. An edge $(i, c, j)$ means that $i$ and $j$ participate in $c$ and that there is an incomplete local state $q_i \in Q_i$ such $i(c)$ is enabled in $q_i$. This means that there could be a global state where $i$ is waiting for $j$ due to the connector $c$.

*Example 1 continued.* The dependency graph $G_{Sys_{tasks}}$ is given in Fig. 2 for $n = 3$. For better readability we define $l_{ij} := \left(conn_3^{ij}, \{finish_j\}\right)$ where $conn_3^{ij} = \{resume_i, finish_j\}$. Moreover we omit the label $conn_g$. Therefore all edges without label in Fig. 2 carry the label $conn_g$.

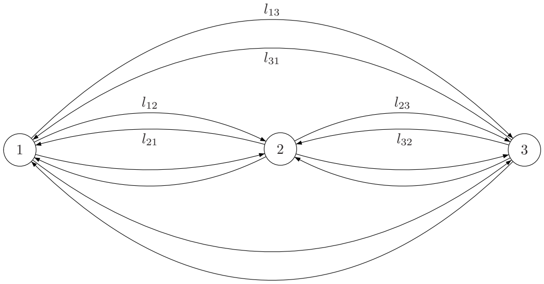

**Fig. 2.** $G_{Sys_{tasks}}$

Next we define predicates that are evaluated on $Q$.

**Definition 14.** *Let Sys be an interaction system.*

1. *For* $e = (i, c, j)$ *we set* $cond\,(e) := en\,(i\,(c)) \wedge \exists x \in c : \neg en\,(x)$.
2. *For* $e = (i, (c, \alpha), j)$ *we set* $cond\,(e) := en\,(i\,(c)) \wedge \exists x \in \alpha : \neg en\,(x)$.
3. *For a path* $p = e_1, \ldots, e_r$ *in* $G_{Sys}$ *we set* $cond\,(p) := \bigwedge_{l=1}^{r} cond\,(e_l)$.

For an edge $e = (i, c, j)$, $cond(e)$ is satisfied in state $q = (q_1, \ldots, q_i, \ldots) \in Q$ if $i(c)$ is enabled in $q_i$ but $c$ is not enabled in $q$ because at least one component does not provide the necessary action.

**Definition 15.** *Let Sys be an interaction system.*

1. *A path p in* $G_{Sys}$ *is called* critical *if* $\left(cond\,(p) \wedge \bigwedge_{i \in p} inc\,(i)\right) \not\equiv false$. *A path p in* $G_{Sys}\,(q)$ *is called* critical *if* $\left(cond\,(p) \wedge \bigwedge_{i \in p} inc\,(i)\right)(q) = true$. *A path that is not critical is called* non-critical.
2. *Let p be a critical cycle in a successor-closed subgraph* $G_f = (K_f, E_f)$ *of* $G_{Sys}$. *p is refutable, if, whenever p lies in* $G_f\,(q)$ *where* $q_i \in inc\,(i)$ *for all i, there is a non-critical path* $\hat{p}$ *in* $G_f\,(q)$.

A path is critical if there is some $q = (q_1, \ldots, q_i, \ldots) \in Q$ such that $q_i$ is incomplete for all components $i$ on the path and $cond(e)$ is satisfied in $q$ for every edge $e$ on the path. If a cycle in $G_{Sys}$ is critical it describes a potential circular waiting relation among components.

**Theorem 1.** *Let Sys be a globally deadlock-free interaction system as above and let $A' \subsetneq A$ be a set of ports. Global deadlock-freedom is robust in Sys w.r.t. absence of $A'$ if the following conditions hold.*

1. *There is no $a \in A'$ such that $\{a\} \in C \cup Comp$.*
2. *$G_{Sys}$ contains a finite successor-closed subgraph $G_f = (K_f, E_f)$ such that*
   (a) *For all $e = (i, c, j) \in E_f$ we have $c \in EXCL\,(A')$.*
   (b) *For all $e = (i, (c, \alpha), j) \in E_f$ we have $\alpha \in EXCL\,(A')$.*
   (c) *Every critical cycle in $G_f$ is refutable.*

The proof can be found in the technical report [27]. Basically, if $G_{Sys}$ contains a successor-closed subgraph $G_f$ as above, for every state $q \in Q$ this subgraph yields $\alpha \in C \cup Comp$ that can be executed in $q$.

*Example 1 continued.* It is not hard to see that the conditions of Theorem 1 are satisfied for any $A' \subseteq \{resume_1, \ldots, resume_n\}$ and robustness of global deadlock-freedom w.r.t. absence of $A'$ follows. A situation where $resume_i$ fails for some $i$ can be understood in such a way that the system may function as usual without this action as long as component $i$ does not allow any other component to enter the critical region before it has finished its task. In case it performs a $preempt_i$ action together with some other component, the component $i$ will be excluded from any further participation while the global system continues operating.

### 4.2 Liveness Without $A'$

Here we transform the criterion of [19] that ensures liveness of a set of components $K'$ to handle the case of failure of $A'$.

We define $excl\,(A', K')$ the set of maximal and complete interactions that neither involve any action from $A'$ nor any component from $K'$.

**Definition 16.** *Let $K' \subseteq K$ be a subset of components. Let $excl\,(A', K') := \{\alpha \in EXCL\,(A')\,|\forall i \in K' : i\,(\alpha) = \emptyset\}$.*

**Definition 17.** *Let Sys be an interaction system as above and let $j \in K$ be a component.*

1. *We define $need_j\,(A') := \{a_j \in A_j | a_j \in \alpha \Rightarrow \alpha \in WITH\,(A')\}$ the set of ports of $j$ that only occur in maximal or complete interactions also involving $A'$.*
2. *Let $B_j \subseteq A_j$ be a subset of actions of $j$. $B_j$ is* weakly inevitable w.r.t. *$A'$ in $T_j$ if the following two conditions hold:*
   (a) *There is an infinite path in the transition system obtained by canceling all transitions in $T_j$ that are labeled with an action from $need_j\,(A')$.*
   (b) *On every infinite path in the transition system obtained this way only finitely many transitions labeled with $a_j \in A_j \backslash B_j$ can be performed before some action from $B_j$ must be performed.*

3. *Let $\Lambda \subseteq I(C)$ be a nonempty set of interactions and let $j \in K$ be a component. We define $\Lambda[j] := A_j \cap \bigcup_{\alpha \in \Lambda} \alpha$ the set of ports of $j$ that participate in one of the interactions of $\Lambda$.*

The set $need_j(A')$ contains exactly those actions of $j$ that can only be performed in the global system if an action from $A'$ is also performed at the same time. Note that it is clear that $(A' \cap A_j) \subseteq need_j(A')$. Further a subset of actions of component $j$ is weakly inevitable w.r.t. $A'$ in $T_j$ if it is possible in $T_j$ to choose an infinite path that does not contain a transition labeled with an action from $need_j(A')$ and if for all such paths there are infinitely many transitions that are labeled with some action from the set in question. The last part of the definition introduces a sort of a projection-operator that yields those actions of component $j$ that participate in one of the interactions in $\Lambda$.

In the following we define a graph $G := (K, E)$ for an interaction system with a finite set $K$ of components and finite port sets which is a modification of the graph introduced [19] to establish liveness. Informally, an edge $e = (i, j) \in E$ has the meaning that component $j$ can only participate in finitely many global steps before $i$ has to participate as well.

**Definition 18.** *Let $G := (K, E)$ with $E := \bigcup_{m=0}^{\infty} E_m$, where:*

$$E_0 := \{(i,j) \,|A_j \backslash excl\,(A', i)\,[j] \text{ is weakly inevitable w.r.t. } A' \text{ in } T_j\}$$

$$E_{n+1} := \{(i,j) \,|A_j \backslash excl\,(A', R^n(i))\,[j] \text{ is weakly inevitable w. r. t. } A' \text{ in } T_j\}$$

$$R^n(i) := \{j | j \text{ is reachable from } i \text{ in } (K, \cup_{m=0}^{n} E_m)\}$$

**Theorem 2.** *Let Sys be a globally deadlock-free finite interaction system such that global deadlock-freedom is robust w.r.t. absence of $A' \subsetneq A$. Let $K' \subseteq K$ be a set of components. $K'$ is live without participation of $A'$ in Sys if all components $i$ in $K \backslash K'$ such that $T_i$ contains an infinite path that is only labeled with actions that are not in $need_i(A')$ are reachable from $K'$ in $G$. The construction of the graph and the reachability analysis can be performed in time polynomial in $|C \cup Comp|$ and the sum of the sizes of the local transition systems.*

The proof can be found in the technical report [27].

*Example 2.* We model a system consisting of a user $u$, two service components $s_1$ and $s_2$ and two maintenance components $m_1$ and $m_2$. The local transition systems of these components are given in Fig. 3. It is understood that the port sets are given implicitly by the transition systems. The initial states are marked by ingoing arrows.

The following connector set defines the allowed cooperations:

$$C := \left\{\{internal_i\}, \{req_i, service_i\}, \{maint_i, m_j^i\} \,|i, j = 1, 2\right\}$$

Further we define $Comp := \emptyset$. In the global system a state where a global deadlock occurs cannot be reached. It is clear that global deadlock-freedom is robust w.r.t. absence of $A_{m_2}$.

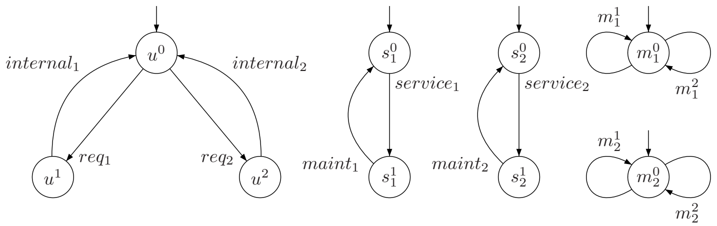

**Fig. 3.** A system of one user and two servers

Figure 4 depicts part of the graph $G$ for this system. It is clear that the condition of Theorem 2 is satisfied yielding liveness of $m_1$ without $A_{m_2}$. This property guarantees, that after each use a service component will undergo maintenance even if the second maintenance component fails.

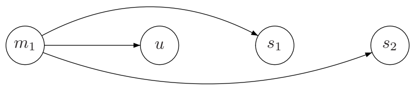

**Fig. 4.** $G$ for the user/server example

### 4.3 Treating Local Progress and Local Deadlock

Here we want to outline the ideas how the criteria for local progress of a component [22] and local deadlock-freedom [18] can be adapted such that they can be used to test whether a component $i \in K$ can make local progress without $A' \subsetneq A$ respectively whether local deadlock-freedom is robust w.r.t. absence of $A' \subsetneq A$.

In [22] a criterion for local progress of a component $i$ was presented. This criterion is based on the dependency graph from Definition 13. The criterion demands the existence of a successor-closed subgraph $G_{f,i}$ as in Theorem 1 such that $i \in G_{f,i}$. Moreover every subset of nodes of $G_{f,i}$ has to be controllable for the notion of controllability defined for subsets $K' \subseteq K$ of components in [22]. Controllability of $K'$ basically ensures that, whenever a global interaction needs participation of components from $K'$, a certain path ending in a state that provides the needed interaction can be chosen in the subsystem defined by $K'$. This idea can be adapted to test whether a component can make local progress without $A' \subsetneq A$. Again it must be possible to choose $G_{f,i}$ such that no label contains any action from $A'$. Furthermore the definition of controllability has to be changed such that the path eventually providing the needed interaction can be chosen such that it does not involve any port from $A'$.

Finally we discuss robustness of local deadlock-freedom. We informally explain how our algorithm from [18] can be adapted such that it can be used to ensure that local deadlock-freedom is robust with respect to absence of $A' \subsetneq A$.

First we will sketch the idea of the algorithm from [18]: in a first step for every three-element subset $\{i, j, k\} \subseteq K$ this algorithm calculates the states $q_{ijk}$ that are reachable in the system consisting of these three components under the assumption that for every connector the actions belonging to components from $K \backslash \{i, j, k\}$ are always available[1]. This amounts to an over-approximation of the projection of the set of the globally reachable states to $\{i, j, k\}$. Then for each of these triple-states the algorithm checks the following necessary condition for a local deadlock. If there is a global state $q$ and a set $D \subseteq K$ such that $D$ is in local deadlock in $q$ there must be $i, j, k \in D$ with $i \neq j \neq k$ such that $i$ is blocked by $j$ and $j$ is blocked by $k$ where a component $j$ blocks a component $i$ in $q$ if $i$ offers an action that occurs in a maximal or complete interaction $c$ that $j$ participates in, but $j(c)$ is not enabled in $q_j$. If this condition is violated for every such subsystem the algorithm affirms local deadlock-freedom. This idea only needs to be slightly adapted in order to ensure that local deadlock-freedom is robust w.r.t. absence of $A' \subsetneq A$ in a system. The first step of the algorithm is identical to the original algorithm. This reflects our assumption that $A'$ may fail at any point of time which means that to begin with all states that can be reached in the original system can also be reached in the system where $A'$ may fail. The necessary condition for a local deadlock has to be adapted. First it is possible that because of the absence of $A'$ there might be a local state $q_i$ of component $i$ for which all actions that are offered in this state only occur in $\alpha \in WITH(A')$. Such a state should be detected as a locally deadlocked state. The existence of such a state can be checked by investigating all local transition systems and the set $C \cup Comp$. If no such state exists a local deadlock can only occur if there is a set $D \subseteq K$ and a reachable state $q$ such that for every component $i \in D$ the fact that $a_i$ is enabled in $q_i$ and $a_i \in \alpha$ for $\alpha \in EXCL(A')$ implies that there is at least one $j \in D$ such that $j(\alpha)$ is not enabled in $q_j$. From the second step of the algorithm it follows that there is at least one such $\alpha$ for every $i \in D$. Moreover there must be at least one $i \in D$ such that $a_i$ is enabled in $q_i$ that occurs in $\alpha \in WITH(A')$. If this was not the case then the local deadlock would have been there before the failure of $A'$ which is a contradiction to the assumption. Therefore the necessary condition for a local deadlock amounts to checking whether there are $i, j, k \in K$ and a reachable sub-global state such that $k$ blocks $j$ and $j$ blocks $i$ (this time only interactions from $EXCL(A')$ are considered for possible blockings) and at least one of the three components is affected by the loss of $A'$ in the sense described above. If this condition is never fulfilled the system at hand does not contain any local deadlocks even if the actions from $A'$ are not available any more.

## 5 Conclusion and Future Work

This work investigates a notion of robustness in interaction systems. The contributions are as follows. 1) We presented notions of robustness of global and local deadlock-freedom w.r.t. failure of a set $A' \subsetneq A$ of ports. Further we introduced

[1] We can increase accuracy by considering subsystems of fixed size $d$.

notions of local progress and liveness without participation of a set $A' \subsetneq A$ of ports. 2) We explained how sufficient conditions for desired properties can be adapted to handle a situation where a set $A' \subsetneq A$ of ports becomes unavailable. We did so in detail for robustness of global deadlock-freedom and for liveness without $A' \subsetneq A$. 3) We informally explained how a similar adaptation is possible for local progress and local deadlock-freedom.

Work is in progress towards treating malfunction of components or ports by introducing probabilities into the framework of interaction systems. In every local state we assign each enabled action a probability that it might fail such that we can make statements such as "with probability $p$ no deadlock will arise" about properties of components. It is clear that this quantitative approach is different from the approach taken here were we want to make assertive statements about the properties in situation where services may fail.

## References

1. Arbab, F.: Abstract Behavior Types: A Foundation Model for Components and Their Composition. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2002. LNCS, vol. 2852, pp. 33–70. Springer, Heidelberg (2002)
2. Chouali, S., Heisel, M., Souquières, J.: Proving Component Interoperability with B Refinement. In: Proceedings of FACS'05. vol. 160. ENTCS, pp. 157–172 (2006)
3. Moschoyiannis, S., Shields, M.W.: Component-Based Design: Towards Guided Composition. In: Proceedings of ACSD'03, pp. 122–131. IEEE Computer Society, Los Alamitos (2003)
4. Bastide, R., Barboni, E.: Software Components: A Formal Semantics Based on Coloured Petri Nets. In: Proceedings of FACS'05. vol. 160, ENTCS, pp. 57–73 (2006)
5. Allen, R., Garlan, D.: A Formal Basis for Architectural Connection. ACM Trans. Softw. Eng. Methodol. 6(3), 213–249 (1997)
6. Nierstrasz, O., Achermann, F.: A Calculus for Modeling Software Components. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2002. LNCS, vol. 2852, pp. 339–360. Springer, Heidelberg (2003)
7. Broy, M.: Towards a Logical Basis of Software Engineering. In: Broy, M., Steinbrüggen, R. (eds.) Calculational System Design, IOS. NATO ASI Series, Series F: Computer and System Sciences, vol. 158, pp. 101–131. Springer, Heidelberg (1999)
8. Baumeister, H., Hacklinger, F., Hennicker, R., Knapp, A., Wirsing, M.: A Component Model for Architectural Programming. In: Proceedings of FACS'05. ENTCS, vol. 160, pp. 75–96. Elsevier, Amsterdam (2006)
9. Gössler, G., Sifakis, J.: Composition for Component-Based Modeling. Sci. Comput. Program. 55(1-3), 161–183 (2005)
10. Sifakis, J.: A Framework for Component-based Construction, SEFM 2005, pp. 293–300 (2005)
11. Gössler, G., Sifakis, J.: Component-Based Construction of Deadlock-Free Systems. In: Pandya, P.K., Radhakrishnan, J. (eds.) FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science. LNCS, vol. 2914, pp. 420–433. Springer, Heidelberg (2003)

12. Gössler, G., Sifakis, J.: Composition for Component-Based Modeling. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2002. LNCS, vol. 2852, pp. 443–466. Springer, Heidelberg (2002)
13. Gössler, G.: Prometheus — A Compositional Modeling Tool for Real-Time Systems. In: Proceedings of RT-TOOLS 2001, Technical report 2001-014, Uppsala University, Department of Information Technology (2001)
14. Basu, A., Bozga, M., Sifakis, J.: Modeling Heterogeneous Real-Time Components in BIP. In: Proceedings of SEFM'06, pp. 3–12. IEEE Computer Society Press, Los Alamitos (2006)
15. Martens, M., Minnameier, C., Majster-Cederbaum, M.: Deciding Liveness in Component-Based Systems is NP-hard. Technical report TR-2006-017, Universität Mannheim (2006)
16. Majster-Cederbaum, M., Minnameier, C.: Deriving Complexity Results for Interaction Systems from 1-Safe Petrinets (2007) (Submitted for publication)
17. Majster-Cederbaum, M., Martens, M., Minnameier, C.: Liveness in Interaction Systems (2007) (Submitted for publication)
18. Majster-Cederbaum, M., Martens, M., Minnameier, C.: A Polynomial-Time-Checkable Sufficient Condition for Deadlock-freeness of Component Based Systems. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 888–899. Springer, Heidelberg (2007)
19. Gössler, G., Graf, S., Majster-Cederbaum, M., Martens, M., Sifakis, J.: An Approach to Modelling and Verification of Component Based Systems. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 295–308. Springer, Heidelberg (2007)
20. Troubitsyna, E.: Developing Fault-Tolerant Control Systems Composed of Self-Checking Components in the Action Systems Formalism. In: Van, H.D., Liu, Z. (eds.) Proceeding of FACS'03, TR 284, UNU/IIST, pp. 167–186 (2003)
21. Charron-Bost, B., Toueg, S., Basu, A.: Revisiting Safety and Liveness in the Context of Failures. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 552–565. Springer, Heidelberg (2000)
22. Gössler, G., Graf, S., Majster-Cederbaum, M., Martens, M., Sifakis, J.: Ensuring Properties of Interaction Systems. In: Program Analysis and Compilation. LNCS, vol. 4444, pp. 201–224. Springer, Heidelberg (2007)
23. Lamport, L.: Proving the Correctness of Multiprocess Programs. IEEE Trans. Software Eng. 3(2), 125–143 (1977)
24. Berard, B., et al.: Systems and Software Verification. Springer, Heidelberg (1999)
25. Cheng, A., Esparza, J., Palsberg, J.: Complexity Results for 1-Safe Nets. Theoretical Computer Science 147(1-2), 117–136 (1995)
26. Attie, P.C., Chockler, H.: Efficiently Verifiable Conditions for Deadlock-Freedom of Large Concurrent Programs. In: Cousot, R. (ed.) VMCAI 2005. LNCS, vol. 3385, pp. 465–481. Springer, Heidelberg (2005)
27. Majster-Cederbaum, M., Martens, M.: Robustness in Interaction Systems. Technical report TR-2007-004, Universität Mannheim (2007)