

# Component Substitutability via Equivalencies of Component-Interaction Automata

I. Černá, P. Vařeková and B. Zimmerova

Faculty of Informatics, Masaryk University  
Brno, Czech Republic

FACS'06

September 21, 2006

# Reconfiguration correctness

## The issue:

- Component-based systems – evolve simply by component update
- Third party components – interaction correctness of updated system

## Our solution:

Formal **characterization of relationship** between components which guarantees that the update **does not break** existing functionality of the system.

## ① Component-interaction automata

Underlying formalism

## ② Reconfiguration correctness

Substitutability of equivalent components

Independent implementability

Substitutability of non-equivalent components

## ③ Conclusion

Related work and summary of the talk

# Component-interaction automata language

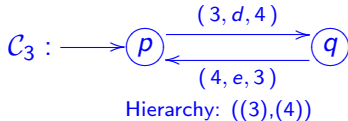
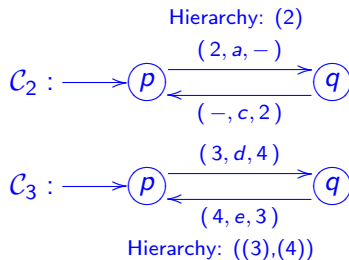
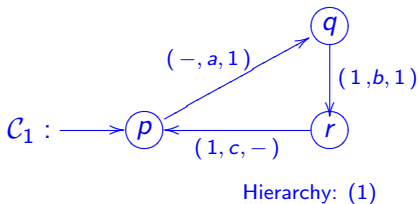
## Component-Interaction automata language (or CI automata for short)

- Automata-based language  
finite state model, infinite executions/traces
- Three types of actions (*input*, *output* and *internal*)  
general used concept
- CCS-like synchronization  
one input and one output action which becomes internal later on
- Flexible composition  
can be parametrized by architectural assembly of the system

# Definition of a CI automaton

## A component-Interaction automaton

- States (initial)
- Labeled transitions
- Labels (structured - component names, actions)
  - input, output and internal
- Hierarchy



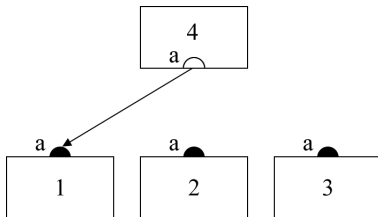
# Composition of CI automata

A parametrizable **composition operator**  $\otimes^{\mathcal{F}}$  determines a **composite automaton**  $\otimes^{\mathcal{F}} \mathcal{S}$  as

- a product of automata from  $\mathcal{S}$
- CCS-like synchronization
- the transitions with labels outside  $\mathcal{F}$  are **removed**

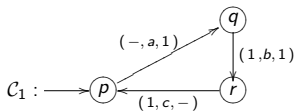
## Application:

The composition respects **bindings between components** given by feasible labels in  $\mathcal{F}$ .

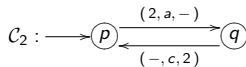


# Composition of CI automata - example

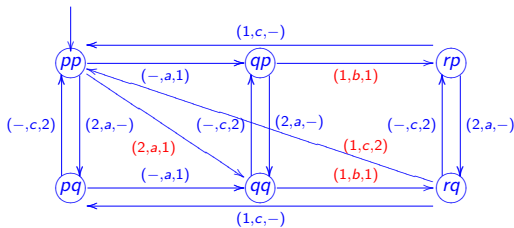
**Example:** The right blue figure depicts the composite automaton  $\mathcal{C} = \otimes^{\mathcal{F}} \{C_1, C_2\}$  where  $\mathcal{F} = \{(2, a, 1), (1, b, 1), (1, c, 2)\}$ .



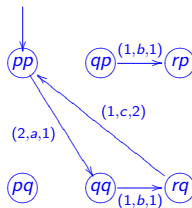
Hierarchy: (1)



Hierarchy: (2)

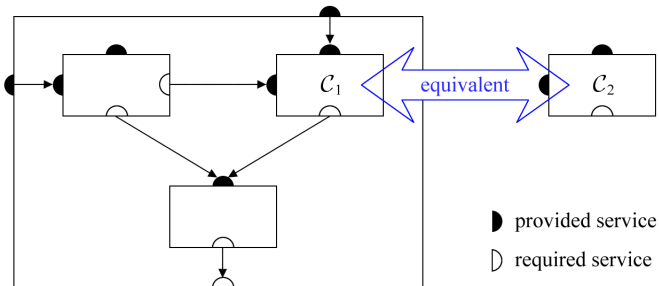


Hierarchy: ((1),(2))



Hierarchy: ((1),(2))

# Substitutability of equivalent components – motivation

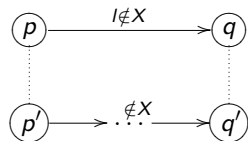
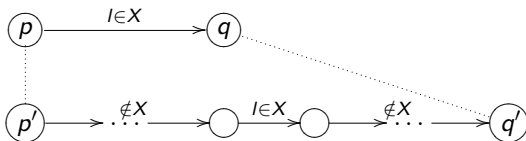


# Equivalence relation

**Equivalence**  $\mathcal{C} \equiv_X \mathcal{C}'$  of CI automata  $\mathcal{C}, \mathcal{C}'$  with respect to observable labels  $X$  is defined:

- similarly to **weak bisimulation**
- **observable** labels  $l \in X$ ; **silent** labels  $l \notin X$

**Illustration of the concept:**



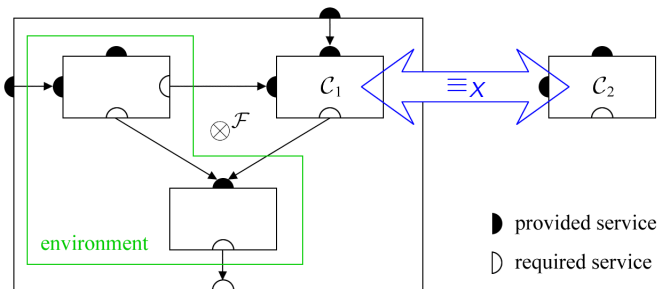
# Various form of the equivalence relation

For CI automata  $\mathcal{C}$  and  $\mathcal{C}'$  some special cases of the set  $X$  are:

- $X = \mathcal{L}_{\mathcal{C}} \cup \mathcal{L}_{\mathcal{C}'}$   
An analogy of **strong bisimulation**.
- $X = \mathcal{L}_{ext,\mathcal{C}} \cup \mathcal{L}_{ext,\mathcal{C}'}$   
An analogy of **weak bisimulation**.
- $X = \mathcal{L}_{\mathcal{C}} \cup \mathcal{L}_{ext,\mathcal{C}'}$   
A **refinement** of  $\mathcal{C}$  by  $\mathcal{C}'$ .

Equivalence of  $\mathcal{C}$  to  $\mathcal{C}'$  **up to 1:1 (1:N) renaming**

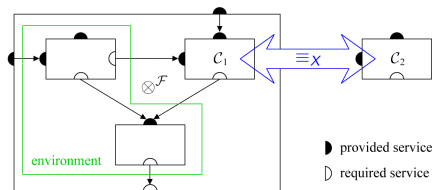
# Substitutability of equivalent components



## We need to consider:

- Form of the equivalence – given by  $X$  in  $\equiv_X$
- Role of the environment – given by  $\mathcal{F}$  in  $\otimes^{\mathcal{F}}$

# Substitutability of equivalent components



## Theorem:

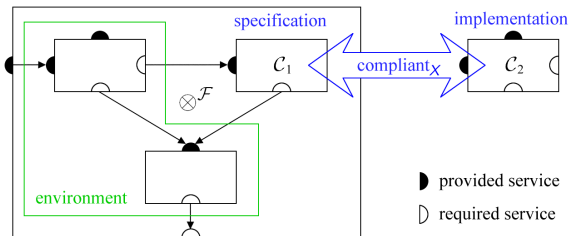
Let  $C_1$ ,  $C_2$  and  $C_3$  be CI automata and  $\mathcal{F}$  a set of labels such that the automata  $\otimes^{\mathcal{F}}\{C_1, C_3\}$  and  $\otimes^{\mathcal{F}}\{C_2, C_3\}$  are defined. Let  $X$  be a set of labels such that  $X \supseteq \bigcup_{i \in \{1,2\}} \mathcal{L}_{ext, C_i}$ . Then

$$\text{if } C_1 \equiv_X C_2 \text{ then } \otimes^{\mathcal{F}}\{C_1, C_3\} \equiv_X \otimes^{\mathcal{F}}\{C_2, C_3\}$$

# Independent implementability – motivation

## Motivation:

- Independent development of components
- Safe reuse of a component in any system where its implementation satisfies the specification stated by the environment



# Compliance relation

**The compliance relation** can be informally defined as follows:

*Interface requirements:*

- The implementation **provides/requires all** that the specification does
- The implementation **may provide/require more**

*Behavioural requirement:*

- When serving the services of the specification, the implementation **behaves according to the specification**

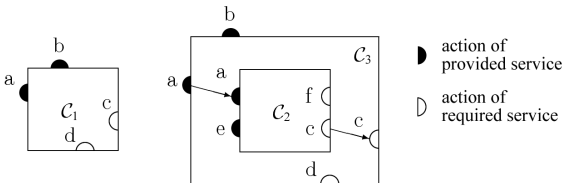
# Compliance relation

## Formal definition:

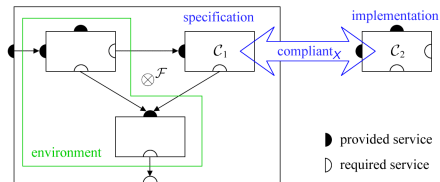
Let  $\mathcal{C}_1$  and  $\mathcal{C}_2$  be CI automata,  $X$  be a set of labels. Then

- $\mathcal{C}_2$  is **compliant to**  $\mathcal{C}_1$  with respect to observable labels  $X$  in a **blocking** environment, iff  $\mathcal{C}_1 \equiv_X \otimes^{\mathcal{R}} \{\mathcal{C}_2\}$  where  $\mathcal{R} = \mathcal{L}_{ext, \mathcal{C}_1} \cup \mathcal{L}_{int, \mathcal{C}_2}$ ,
- $\mathcal{C}_2$  is **compliant to**  $\mathcal{C}_1$  with respect to observable labels  $X$  in a **non-blocking** environment, iff  $\mathcal{C}_1 \equiv_X \otimes^{\mathcal{R}} \{\mathcal{C}_2\}$  where  $\mathcal{R} = \mathcal{L}_{ext, \mathcal{C}_1} \cup \mathcal{L}_{int, \mathcal{C}_2} \cup \mathcal{L}_{out, \mathcal{C}_2}$ .

Architectural view on restriction of  $\mathcal{C}_2$  to given  $\mathcal{C}_1$



# Independent implementability



## Theorem:

Let  $C_1$ ,  $C_2$  and  $C_3$  be CI automata,  $\mathcal{R}$  the set given by the def. Let  $\mathcal{F}$  be a set of labels such that  $\mathcal{F} \cap \text{Participate}(\mathcal{L}_{C_2} \setminus \mathcal{R}) = \emptyset$  and the automata  $\otimes^{\mathcal{F}}\{C_1, C_3\}$ ,  $\otimes^{\mathcal{F}}\{C_2, C_3\}$  are defined. Let  $X$  be a set of labels such that  $X \supseteq (\bigcup_{i \in \{1,2\}} \mathcal{L}_{\text{ext}, C_i}) \cap \mathcal{R}$ . Then

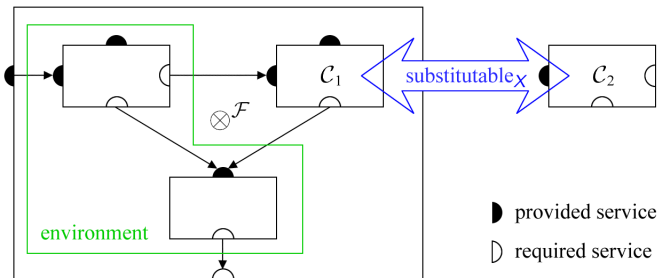
if  $C_2$  is compliant to  $C_1$  w.r.t.  $X$  then

$$\otimes^{\mathcal{F}}\{C_1, C_3\} \equiv_X \otimes^{\mathcal{F}}\{C_2, C_3\}$$

# Substitutability of non-equivalent components – motivation

## Motivation:

- The **substitutability of equivalent components fails** when the two components are not equivalent.
- Even then new system **may be equivalent** to the previous one.
- Because the behaviour that distinguishes the components may be **unused** in the system.



# Substitutability relation

## The substitutability relation

- Extends the notion of the specification–implementation relation
- A new component **does not have to simulate** behaviours of the former component that are **not used** by the environment

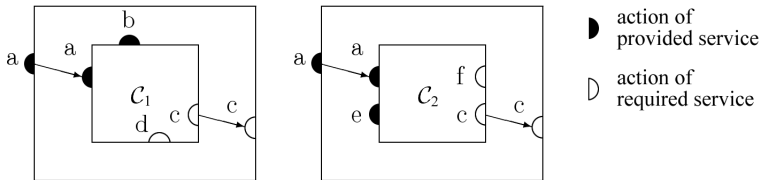
# Substitutability relation

## Formal definition:

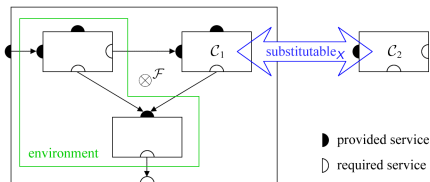
Let  $\mathcal{C}_1$  and  $\mathcal{C}_2$  be CI automata,  $X$ ,  $E$  be sets of labels. Then

- $\mathcal{C}_2$  is **substitutable for**  $\mathcal{C}_1$  with respect to observable labels  $X$  and labels (used by the environment)  $E$  in a **blocking** environment, iff  $\otimes^{\mathcal{R}}\{\mathcal{C}_1\} \equiv_X \otimes^{\mathcal{R}}\{\mathcal{C}_2\}$  where  $\mathcal{R} = E \cup \mathcal{L}_{int,\mathcal{C}_1} \cup \mathcal{L}_{int,\mathcal{C}_2}$ ,
- $\mathcal{C}_2$  is **substitutable for**  $\mathcal{C}_1$  with respect to observable labels  $X$  and labels (used by the environment)  $E$  in a **non-blocking** environment, iff  $\otimes^{\mathcal{R}}\{\mathcal{C}_1\} \equiv_X \otimes^{\mathcal{R}}\{\mathcal{C}_2\}$  where  $\mathcal{R} = E \cup \mathcal{L}_{int,\mathcal{C}_1} \cup \mathcal{L}_{int,\mathcal{C}_2} \cup \mathcal{L}_{out,\mathcal{C}_1} \cup \mathcal{L}_{out,\mathcal{C}_2}$ .

Restriction of  $\mathcal{C}_1$  and  $\mathcal{C}_2$  for the same environment



# Substitutability of non-equivalent components



## Theorem:

Let  $C_1$ ,  $C_2$  and  $C_3$  be CI automata,  $E$  a set of labels, and  $\mathcal{R}$  the set given by the def. Let  $\mathcal{F}$  be a set of labels such that  $\mathcal{F} \cap \text{Participate}((\mathcal{L}_{C_1} \cup \mathcal{L}_{C_2}) \setminus \mathcal{R}) = \emptyset$  and the automata  $\otimes^{\mathcal{F}}\{C_1, C_3\}$  and  $\otimes^{\mathcal{F}}\{C_2, C_3\}$  are defined. Let  $X$  be a set of labels such that  $X \supseteq (\bigcup_{i \in \{1,2\}} \mathcal{L}_{\text{ext}, C_i}) \cap \mathcal{R}$ . Then

if  $C_2$  is substitutable for  $C_1$  w.r.t.  $X, E$  then

$$\otimes^{\mathcal{F}}\{C_1, C_3\} \equiv_X \otimes^{\mathcal{F}}\{C_2, C_3\}$$

# Related work

## Related approaches

- Interface automata - refinement relation  
alternating simulation
- SOFA behavior protocols - compliance relation  
trace language inclusion
- Chaki et al. - substitutability relation  
a substitutability framework

# Summary of the talk

## Three specific issues:

- Substitutability of equivalent components  
via [equivalence](#) relation
- Independent implementability  
via [specification–implementation](#) relation, called [compliance](#)
- Substitutability of non-equivalent components  
via [substitutability](#) relation

# Thank you

**Thank you** for your attention

