Effective Verification of Systems with a Dynamic Number of Components

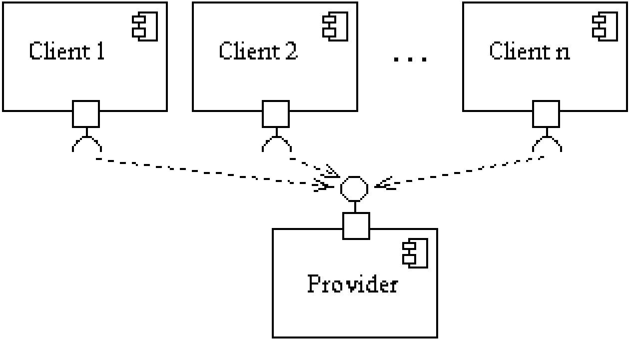
P.Vařeková, P. Moravec, I. Černá, and B. Zimmerova

Faculty of Informatics
Masaryk University, Brno

SAVCBS'07

Contents

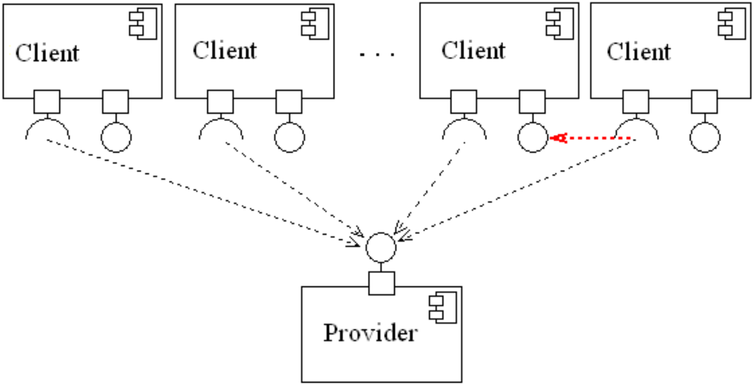
Dynamic systems – Introduction



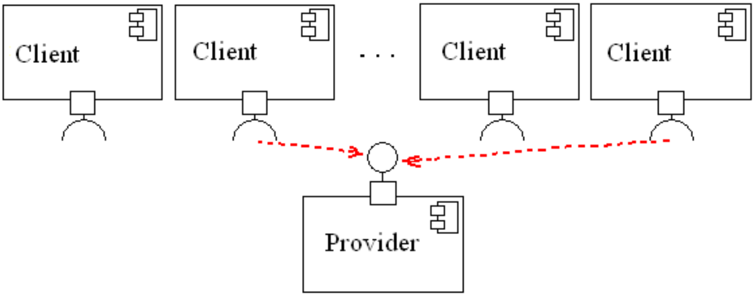
$\mathcal{S}$ - Dynamic system

$\mathcal{S}_n$ - Dynamic system with n clients deployed

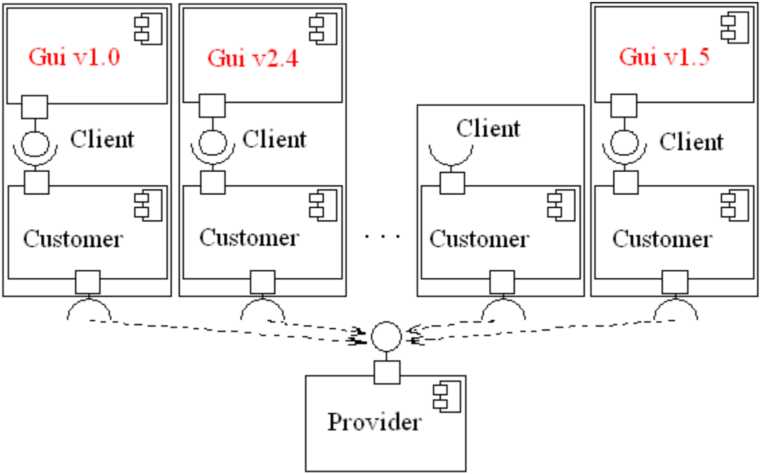
Dynamic system – Definition



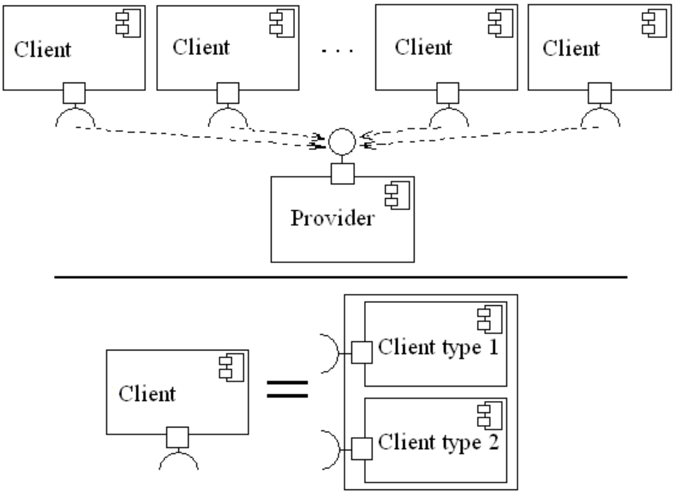
Dynamic system – Definition



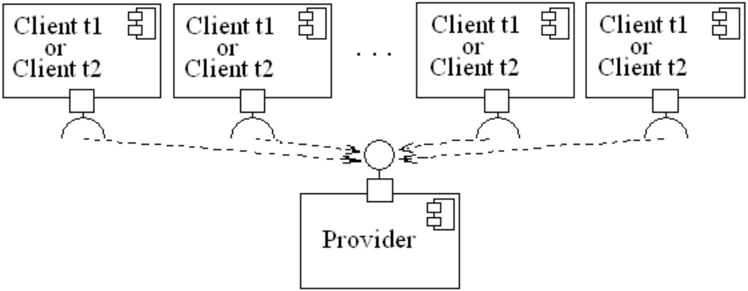
Dynamic system – Definition



Dynamic system – Definition

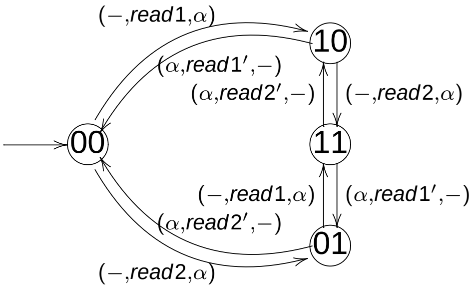


Dynamic system – Definition



Specification language

We use Component Interaction automata



A hierarchy of component names: (α)

Can be modelled by

- Finite transitions systems or
- Regular-like expressions

Contents

Properties – Example

”If a client of the system sends a request, then he will receive a response.”

Properties – Example

"If a client of the system sends a request, then he will receive a response."

$$\Updownarrow$$

$\forall \mathcal{S}_n \ (n \in \mathbb{N}_0)$:
"If client $i_{\in\{1,\dots,n\}}$ *sends a request, then he will receive a response."*

Properties – Example

"If a client of the system sends a request, then he will receive a response."

$$\Updownarrow$$

$\forall \mathcal{S}_n \ (n \in \mathbb{N}_0)$:

"If client $i_{\in\{1,\ldots,n\}}$ *sends a request, then he will receive a response."*

$$\Updownarrow$$

$\forall \mathcal{S}_n \ (n \in \mathbb{N}_0)$:

$\varphi_n = \bigwedge_{i\in\{1,\ldots,n\}} G(\mathcal{P}_{(i,request,\alpha)} \Rightarrow F\ \mathcal{P}_{(\alpha,response,i)}\ )$

Properties – Example

"If a client of the system sends a request, then he will receive a response."

$$\Updownarrow$$

$\forall \mathcal{S}_n \ (n \in \mathbb{N}_0)$:
"If client $i_{\in\{1,\ldots,n\}}$ sends a request, then he will receive a response."

$$\Updownarrow$$

$\forall \mathcal{S}_n \ (n \in \mathbb{N}_0)$:
$\varphi_n = \bigwedge_{i\in\{1,\ldots,n\}} G(\mathcal{P}(i, request, \alpha) \Rightarrow F\ \mathcal{P}(\alpha, response, i)\ )$

$$\Updownarrow$$

$$\forall n \in \mathbb{N}_0: \quad \mathcal{S}_n \models \varphi_n$$

Properties – Introduction

- Property: $\{\varphi_i\}_{i \in \mathbb{N}_0}$
- Property is satisfied $\Leftrightarrow \forall n \in \mathbb{N}_0 : \mathcal{S}_n \models \varphi_n$
- We use
  - φ_i - temporal logic $CI-LTL$
  - $CI-LTL$ - an extension of action based LTL
    - $\mathcal{P}(I)$ - I is performed as the first action of the path
    - $\mathcal{E}(I)$ - I is enabled in the first state of the path

Properties – Main restriction

Restrictions

- no distinctions among clients,
- properties whose violation involves only a finite number of observed components

- $\mathit{Property}(\mathcal{S}, m)$
  - no distinction among clients
  - violation involves m observed components

Properties – *Property*$(\mathcal{D}, m)$ – Examples

Example 1/3:

- *"If a client of the system sends a request, then he will receive a response."*
- path π violates it $\Rightarrow$ a client "*send a request and does not receive a response*"
- we can observe only this client, to show that this property is violated in π
- we need to observe 1 client.
- $\in$ *Property*$(\mathcal{S}, 1)$

Properties – *Property*$(\mathcal{D}, m)$ – Examples

Example 2/3:

- *"Two clients can not be able to receive a response at the same time."*
- path π violates it $\Rightarrow$ clients j_1 and j_2 "can receive a response at the same time"
- we can observe only clients j_1 and j_2, to show that this property is violated in π
- we need to observe 2 clients.
- $\in$ *Property*$(\mathcal{S}, 2)$

Properties – $Property(\mathcal{D}, m)$ – Examples

Example 3/3:

- *"The system does not contain a deadlock."*
- path π violates it $\Rightarrow$ all clients and prvider reach the state from which they can not continue
- we must observe all clients, to show that this property is violated in π
- we need to observe n clients in $\mathcal{S}_n$.
- $\notin Property(\mathcal{S}, m)$ for any $m \in \mathbb{N}_0$

Properties – *Property*$(\mathcal{D}, m)$ – Overview

√ *If a component* **tries to emit** *an event on its required interface, the counterpart is* **able to absorb** *it.*

Interface automata, SOFA

× *System does not contain a* **deadlock**.

FOCUS, JavaA, rCOS, SOFA

√ *Situation when communication of components in the group never finished is unreachable.*

SOFA

× *A state in which more than half of clients are in a critical section is unreachable.*

Contents

Verification – Introduction

Verification problem

Input: $\mathcal{S}$,

$\{\varphi_i\}_{i \in \mathbb{N}_0} \in \mathit{Property}(\mathcal{S}, m)$ for some $m \in \mathbb{N}_0$

Question: $\forall i \in \mathbb{N}_0 : \mathcal{S}_i \models \varphi_i$?

Verification of infinitely many finite state transition systems.

Our solution

$$\text{find } k \in \mathbb{N}_0 \text{ such that if } \begin{array}{l} \mathcal{S}_0 \models \varphi_0, \\ \mathcal{S}_1 \models \varphi_1, \\ \quad\vdots \\ \mathcal{S}_k \models \varphi_k, \end{array}$$

$$\text{then} \quad \forall n \in \mathbb{N}_0 : \quad \mathcal{S}_n \models \varphi_n.$$

Verification of finitely many finite state transition systems.

Verification – Our solution

Input $\mathcal{S}$,

$\{\varphi_i\}_{i\in\mathbb{N}_0}$,

$m : \{\varphi_i\}_{i\in\mathbb{N}_0} \in Property(\mathcal{S}, m)$.

Intermediate data

- X - set containing all labels necessary for verification of $\{\varphi_i\}_{i\in\mathbb{N}_0}$
- $|\mathcal{D}|_X \in \mathbb{N}_0 \cup \{\infty\}$

Output

$$k = |\mathcal{D}|_X + m \in \mathbb{N}_0 \cup \{\infty\}$$

Verification – Problem

Input $\mathcal{S}$,
$\{\varphi_i\}_{i\in\mathbb{N}_0}$,
$m : \{\varphi_i\}_{i\in\mathbb{N}_0} \in Property(\mathcal{S}, m)$.

Intermediate data

- X - set containing all labels necessary for verification of $\{\varphi_i\}_{i\in\mathbb{N}_0}$
- $|\mathcal{D}|_X \in \mathbb{N}_0 \cup \{\infty\}$

Output

$$k = |\mathcal{D}|_X + m \in \mathbb{N}_0 \cup \{\infty\}$$

Conclusions

- Dynamic systems
- Properties
- Properties whose violation involves finite number of clients
- Verification

Conclusions

Thank you for you attention.