Semi-Automatic Specification of Behavioural Service Adaptation Contracts

J. Cámara, J. Antonio Martín, G. Salaün, C. Canal, E. Pimentel

FESCA'10
27 March
Paphos, Cyprus
Service Adaptation
Service Adaptation
Definition [STS] A Symbolic Transition System or STS is a tuple $(A, S, I, F, T)$ where: $A$ is an alphabet which corresponds to the set of labels associated to transitions, $S$ is a set of states, $I \in S$ is the initial state, $F \subseteq S$ are the final states, and $T : S \times A \times S$ is the transition function.
Incompatible Services

\[ i, j \in \{1..n\} \quad i \neq j \]

\[ \langle s_i, E_i \rangle \xrightarrow{a!v} \langle s'_i, E_i \rangle \quad \langle s_j, E_j \rangle \xrightarrow{a?x} \langle s'_j, E_j \rangle \]

\[ \text{type}(x) = \text{type}(v) \quad E'_j = E_j \ominus \langle x, v \rangle \]

\[ \{as_1, .., \langle s_i, E_i \rangle, .., \langle s_j, E_j \rangle, .., as_n \} \xrightarrow{a!v} \{as_1, .., \langle s'_i, E_i \rangle, .., \langle s'_j, E'_j \rangle, .., as_n \} \]
Behavioural Incompatibilities

\[ \begin{align*}
  \{ a_{s_1}, \ldots, \langle s_i, E_i \rangle, \ldots, \langle s_j, E_j \rangle, \ldots, a_{s_n} \} & \xrightarrow{a!v} \{ a_{s_1}, \ldots, \langle s_i', E_i \rangle, \ldots, \langle s_j', E_j \rangle, \ldots, a_{s_n} \}
\end{align*} \]
Behavioural Incompatibilities

\[ i, j \in \{1..n\} \quad i \neq j \]

\[
\begin{align*}
\langle s_i, E_i \rangle & \xrightarrow{a!v} \langle s_i', E_i \rangle & \langle s_j, E_j \rangle & \xrightarrow{a?x} \langle s_j', E_j \rangle \\
\text{type}(x) = \text{type}(v) & & E_j' = E_j \ominus \langle x, v \rangle
\end{align*}
\]

\[
\{ a_{s_1}, \ldots, \langle s_i, E_i \rangle, \ldots, \langle s_j, E_j \rangle, \ldots, a_{s_n} \} \xrightarrow{a!v} \{ a_{s_1}, \ldots, \langle s_i', E_i \rangle, \ldots, \langle s_j', E_j' \rangle, \ldots, a_{s_n} \}
\]
Behavioural Adaptor

Service c

Service s

Adaptor
Behavioural Adaptor

Service c

Service s
Definition [Vector] A vector for a set of service $STS_i = (A_i, S_i, I_i, F_i, T_i), i \in \{1, \ldots, n\}$ is an element of $A_j \cup (A_j \times A_k)$ with $j, k \in \{1, \ldots, n\}, j \neq k$. Such a vector is noted $\langle s_j : l \rangle$, or $\langle s_j : l, s_k : l' \rangle$ where $s_j, s_k$ are service identifiers, and $l, l'$ are labels on the alphabets of services $A_j, A_k$, where message parameters are substituted by placeholders relating the arguments.
Adaptation Contract

<c:c?I,T;s:p!T,I>
<c:r!U,F;s:u?I,U>
<s:f?I,F>
...
 FREE

Service c

Service s

Adaptor
Adaptation Contract

Service c

<\text{c}:c?!I,T;\text{s}:p!T,I>
<\text{c}:r!U,F;\text{s}:u?!I,U>
<\text{s}:f?!I,F>
<\text{c}:d!L;\text{s}:g?!I,L>
<\text{c}:u?!D;\text{s}:r!l,D>
<\text{c}:a!;\text{s}:e?!I>
<\text{s}:e?!I>

Service s

Contract
"It's a simple procedure. I'm blindfolded, spun around and then I attempt to reattach your tail."
Evaluation

[ Time in logarithmic scale ]
Conclusions

Compared to manual contract specification:

- The time required to design the contract is reduced by 67%
- 77% less errors during the contract specification process

Our proposal worked especially well in cases where functionality is not scattered across multiple small interfaces

- The automatic contract generator works pairwise

As regards future work we have extended the underlying formalisation with adaptation goals expressed in temporal logic and we are currently working on including security concerns to the adaptation
Thank you
Adaptation Behaviour

\[
i \in \{1, \ldots, n\} \quad \langle s_i, E_i \rangle \stackrel{\tau}{\rightarrow}_b \langle s'_i, E_i \rangle \\
\{a_{s_1}, \ldots, \langle s_i, E_i \rangle, \ldots, a_{s_n} \} \stackrel{\tau}{\rightarrow}_c \{a_{s_1}, \ldots, \langle s'_i, E_i \rangle, \ldots, a_{s_n} \}
\]

\[
i, j \in \{1, \ldots, n\} \quad i \neq j \\
\langle s_i, E_i \rangle \stackrel{a_l}{\rightarrow}_b \langle s'_i, E_i \rangle \quad \langle s_j, E_j \rangle \stackrel{a?x}{\rightarrow}_b \langle s'_j, E_j \rangle \\
type(x) = type(v) \quad E'_j = E_j \otimes \langle x, v \rangle
\]

\[
\{a_{s_1}, \ldots, \langle s_i, E_i \rangle, \ldots, \langle s_j, E_j \rangle, \ldots, a_{s_n} \} \stackrel{a_l}{\rightarrow}_c \{a_{s_1}, \ldots, \langle s'_i, E_i \rangle, \ldots, \langle s'_j, E'_j \rangle, \ldots, a_{s_n} \}
\]
Comparing to manual contract specification:

- The time required to design the contract is reduced by 67%
- 77% less errors during the contract specification process

Our proposal worked especially well in cases where functionality is not scattered across multiple small interfaces:

- The automatic contract generator works pairwise

Y así mas...

- Y este de tercer nivel
  - Y este de cuarto
    - Y finalmente este de quinto (y último)

Segundo

Primero

- Y ya está

....
**Definition**  [STS] A Symbolic Transition System or STS is a tuple $(A, S, I, F, T)$ where: $A$ is an alphabet which corresponds to the set of labels associated to transitions, $S$ is a set of states, $I \in S$ is the initial state, $F \subseteq S$ are the final states, and $T : S \times A \times S$ is the transition function.
Definition  [STS] A Symbolic Transition System or STS is a tuple $(A, S, I, F, T)$ where: $A$ is an alphabet which corresponds to the set of labels associated to transitions, $S$ is a set of states, $I \in S$ is the initial state, $F \subseteq S$ are the final states, and $T : S \times A \times S$ is the transition function.
**Definition**  
A *vector* for a set of service $STS_i = (A_i, S_i, I_i, F_i, T_i), i \in \{1, \ldots, n\}$ is an element of $A_j \cup (A_j \times A_k)$ with $j, k \in \{1, \ldots, n\}, j \neq k$. Such a vector is noted $\langle s_j : l \rangle$, or $\langle s_j : l, s_k : l' \rangle$ where $s_j, s_k$ are service identifiers, and $l, l'$ are labels on the alphabets of services $A_j, A_k$, where message parameters are substituted by placeholders relating the arguments.
Adaptation Behaviour

Service c

Service s

Contract

Adaptor

FESCA'10 - 27 March - Paphos, Cyprus