

Travel Agency [1]

Participants:

- C: Customer
- A: Agency
- S: Service

Original version (with delegation)

$$\mu\mathbf{t}.C \rightarrow A \left\{ \begin{array}{l} \textcolor{red}{Request}(\textcolor{brown}{String}).A \rightarrow C \left\{ \begin{array}{l} \textcolor{brown}{Price}(\textcolor{brown}{double}).C \rightarrow A \left\{ \begin{array}{l} \textcolor{red}{Accept}.A \rightarrow S \{ \textcolor{blue}{Delegate}(\textcolor{brown}{?Address}.\textcolor{brown}{!Date}).C \rightarrow S \{ \textcolor{red}{Address}(\textcolor{brown}{String}).S \rightarrow C \{ \textcolor{blue}{Date}(\textcolor{brown}{Date}).\textcolor{red}{end} \} \} \} \\ \textcolor{red}{Reject}.\textcolor{red}{end} \\ \textcolor{red}{Repeat}.\mathbf{t} \end{array} \right\} \end{array} \right\} \end{array} \right\}$$

Without delegation (notice that we have to add messages to S in the $\textcolor{blue}{and}$ branches, otherwise the branches can not be merged).

$$\mu\mathbf{t}.C \rightarrow A \left\{ \begin{array}{l} \textcolor{red}{Request}(\textcolor{brown}{String}).A \rightarrow C \left\{ \begin{array}{l} \textcolor{brown}{Price}(\textcolor{brown}{double}).C \rightarrow A \left\{ \begin{array}{l} \textcolor{red}{Accept}.C \rightarrow S \{ \textcolor{red}{Address}(\textcolor{brown}{String}).S \rightarrow C \{ \textcolor{blue}{Date}(\textcolor{brown}{Date}).\textcolor{red}{end} \} \} \\ \textcolor{red}{Reject}.C \rightarrow S \{ \textcolor{red}{Quit}.\textcolor{red}{end} \} \\ \textcolor{red}{Repeat}.C \rightarrow S \{ \textcolor{blue}{Ping}.\mathbf{t} \} \end{array} \right\} \end{array} \right\} \end{array} \right\}$$

References

- [1] Raymond Hu, Nobuko Yoshida, and Kohei Honda. Session-Based Distributed Programming in Java. In Jan Vitek, editor, *ECOOP 2008 – Object-Oriented Programming*, volume 5142, pages 516–541. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.