# Compliance and subtyping in timed session types

Massimo Bartoletti[a] Tiziana Cimoli[a] Maurizio Murgia[a]
Alessandro Sebastian Podda[a] Livio Pompianu[a]

[a] *Dipartimento di Matematica e Informatica*
*University of Cagliari, Italy*

**Abstract**

We propose an extension of session types, to formalise timed communication protocols between two participants at the endpoints of a session. We introduce a decidable compliance relation, which generalises to the timed setting the usual progress-based notion of compliance between session types. We then show a sound and complete technique to decide when a timed session type admits a compliant one, and if so, to construct the least session type compliant with a given one, according to the subtyping preorder induced by compliance. Decidability of subtyping follows from these results.

## 1 Introduction

Session types are formal descriptions of interaction protocols involving two or more participants over a network [7]. They can be used to specify the behavioural interface of a service, and to statically check through a type system that this conforms to its implementation, so enabling compositional verification of distributed applications. Session types support formal definitions of *compliance* (when two or more session types, composed together, behave correctly), and of *subtyping* (when a service can be replaced by another one, while preserving the interaction capabilities with the context). Since these notions are often decidable or safely approximable, session typing is an attractive approach to the design of distributed applications.

In the simplest setting, session types are terms of a process algebra featuring a selection construct (an *internal choice* among a set of branches), a branching construct (an *external choice* offered to the environment), and recursion. In this basic form, session types cannot capture the timing constraints among the communication actions. While formal methods for time have been studied for at least a couple of decades, they have approached the realm of session types very recently [5]. However, some of the key notions (e.g., compliance, subtyping) have not been explored yet. We think that studying timed session types in a basic setting is worthy of attention. The goal is to preserve some decidability results, like those of com-

*This paper is electronically published in*
*Electronic Notes in Theoretical Computer Science*
*URL:* `www.elsevier.nl/locate/entcs`

pliance and subtyping, enabling the implementation of tools and infrastructures for the development of communication-oriented distributed applications, as done in [3].

This paper is a brief communication of an already published work ([2]). Proofs, omitted definitions and extended bibliography can be found in [4].

## 2 Timed session types

We introduce binary timed session types (TSTs), by giving their syntax and semantics, and a compliance relation between them. The main result of this section is the decidability of compliance (Theorem 2.6).

**Syntax and semantics.**

Let $\mathtt{A}$ be a set of *actions*, ranged over by $\mathtt{a}, \mathtt{b}, \ldots$. We denote with $\mathtt{A}^!$ the set $\{!\mathtt{a} \mid \mathtt{a} \in \mathtt{A}\}$ of *output actions*, with $\mathtt{A}^?$ the set $\{?\mathtt{a} \mid \mathtt{a} \in \mathtt{A}\}$ of *input actions*, and with $\mathtt{L} = \mathtt{A}^! \cup \mathtt{A}^?$ the set of *branch labels*, ranged over by $\ell, \ell', \ldots$. We use $\delta, \delta', \ldots$ to range over the set $\mathbb{R}_{\geq 0}$ of positive real numbers including zero, and $d, d', \ldots$ to range over $\mathbb{N}$. Let $\mathbb{C}$ be a set of *clocks*, namely variables in $\mathbb{R}_{\geq 0}$, ranged over by $t, t', \ldots$. We use $R, T, \ldots \subseteq \mathbb{C}$ to range over sets of clocks. The syntax of guards (ranged over by $g, g', \ldots$) is: $g \ ::= \ \mathtt{true} \mid \neg g \mid g \wedge g \mid t \circ d \mid t - t' \circ d, \circ \in \{\leq, \geq\}$.

A TST $p$ models the behaviour of a single participant involved in an interaction. Roughly, in an internal choice $\bigoplus_i !\mathtt{a}_i\{g_i, R_i\}\,.\,p_i$ a participant has to perform one of the outputs $!\mathtt{a}_i$ in a time window where $g_i$ is true. Conversely, in an external choice $\sum_i ?\mathtt{a}_i\{g_i, R_i\}\,.\,q_i$ the participant is available to receive each message $\mathtt{a}_i$ in *any instant* within the time window defined by $g_i$.

**Definition 2.1** *Timed session types* $p, q, \ldots$ are terms of the following grammar:

$$p \ ::= \ \mathbf{1} \mid \bigoplus_{i \in I} !\mathtt{a}_i\{g_i, R_i\}\,.\,p_i \mid \sum_{i \in I} ?\mathtt{a}_i\{g_i, R_i\}\,.\,p_i \mid \mathrm{rec}\, X\,.\,p \mid X$$

where (i) $I \neq \emptyset$ and finite, (ii) actions in internal/external choices are pairwise distinct, (iii) recursion is guarded. True guards, empty resets, and $\mathbf{1}$ can be omitted.

To define the behaviour of TSTs we use *clock valuations*, which associate each clock with its value. We denote with $\mathbb{V} = \mathbb{C} \to \mathbb{R}_{\geq 0}$ the set of clock valuations (ranged over by $\nu, \eta, \ldots$), and with $\nu_0$ the valuation mapping each clock to zero. We write $\nu + \delta$ for the valuation which increases $\nu$ by $\delta$, i.e., $(\nu + \delta)(t) = \nu(t) + \delta$. For a set $R \subseteq \mathbb{C}$, we write $\nu[R]$ for the *reset* of the clocks in $R$, i.e., $\nu[R](t) = 0$ if $t \in R$, and $\nu[R](t) = \nu(t)$ otherwise. We write $[\![g]\!]$ for the set of clock evaluations satisfying $g$. We use $\mathcal{K}, \mathcal{K}', \ldots$ to range over sets of clock evaluations.

**Definition 2.2 (Semantics of TSTs)** A *configuration* is a term of the form $(p, \nu) \mid (q, \eta)$, where $p, q$ are TSTs extended with *committed choices* $[!\mathtt{a}\{g, R\}]\,p$. The semantics of TSTs is defined as the smallest labelled relation closed under the rules in fig. 1, whose labels are either silent actions $\tau$, delays $\delta$, or branch labels.

Rule $[\oplus]$ allows to commit to the branch $!\mathtt{a}$ of an internal choice, when the corresponding guard is satisfied in the clock valuation $\nu$. This results in the term

$$(\texttt{!a}\{g,R\}.\,p \oplus p',\ \nu) \xrightarrow{\tau} ([\texttt{!a}\{g,R\}]\,p,\ \nu) \qquad \text{if } \nu \in [\![g]\!] \qquad [\oplus]$$

$$([\texttt{!a}\{g,R\}]\,p,\ \nu) \xrightarrow{\texttt{!a}} (p,\ \nu[R]) \qquad [!]$$

$$(\texttt{?a}\{g,R\}.\,p + p',\ \nu) \xrightarrow{\texttt{?a}} (p,\ \nu[R]) \qquad \text{if } \nu \in [\![g]\!] \qquad [?]$$

$$(p,\ \nu) \xrightarrow{\delta} (p,\ \nu+\delta) \qquad \text{if } \delta > 0 \ \wedge\ \nu+\delta \in \texttt{rdy}(p) \qquad [\textsc{Del}]$$

$$\frac{(p,\nu) \xrightarrow{\tau} (p',\nu')}{(p,\nu) \mid (q,\eta) \xrightarrow{\tau} (p',\nu') \mid (q,\eta)}\ [\text{S-}\oplus] \qquad \frac{(p,\nu) \xrightarrow{\delta} (p,\nu') \quad (q,\eta) \xrightarrow{\delta} (q,\eta')}{(p,\nu) \mid (q,\eta) \xrightarrow{\delta} (p,\nu') \mid (q,\eta')}\ [\text{S-}\textsc{Del}]$$

$$\frac{(p,\nu) \xrightarrow{\texttt{!a}} (p',\nu') \quad (q,\eta) \xrightarrow{\texttt{?a}} (q',\eta')}{(p,\nu) \mid (q,\eta) \xrightarrow{\tau} (p',\nu') \mid (q',\eta')}\ [\text{S-}\tau]$$

$$\downarrow \mathcal{K} = \{\nu \mid \exists \delta \geq 0 : \nu+\delta \in \mathcal{K}\} \qquad \mathcal{K}[T]^{-1} = \{\nu \mid \nu[T] \in \mathcal{K}\}$$

$$\texttt{rdy}(\textstyle\bigoplus \texttt{!a}_i\{g_i,R_i\}.\,p_i) = \downarrow \bigcup [\![g_i]\!] \quad \texttt{rdy}(\textstyle\sum \cdots) = \texttt{rdy}(\mathbf{1}) = \mathbb{V} \quad \texttt{rdy}([\texttt{!a}\{g,R\}]\,p) = \emptyset$$

Fig. 1. Semantics of timed session types (modulo unfolding of recursion, symmetric rules omitted).

$[\texttt{!a}\{g,R\}]\,p$ which can only fire $\texttt{!a}$ ([!]). Rule [?] allows an external choice to fire any of its enabled input actions. Rule [Del] allows time to pass; this is always possible for external choices and success term, while for an internal choice we require, through the function `rdy`, that some guard remains satisfiable. The last three rules are almost standard.

**Example 2.3** Let $p = \texttt{!a} \oplus \texttt{!b}\{t > 2\}$, $q = \texttt{?b}\{t > 5\}$, and consider the following:

$$\begin{aligned}(p,\nu_0) \mid (q,\eta_0) &\xrightarrow{7}\xrightarrow{\tau} ([\texttt{!b}\{t>2\}]\,,\nu_0+7) \mid (q,\eta_0+7)\\ &\xrightarrow{\tau} (\mathbf{1},\nu_0+7) \mid (\mathbf{1},\eta_0+7) \qquad (1)\\ (p,\nu_0) \mid (q,\eta_0) &\xrightarrow{\delta}\xrightarrow{\tau} ([\texttt{!a}]\,,\nu_0+\delta) \mid (q,\eta_0+\delta) \qquad (2)\\ (p,\nu_0) \mid (q,\eta_0) &\xrightarrow{3}\xrightarrow{\tau} ([\texttt{!b}\{t>2\}]\,,\nu_0+3) \mid (q,\eta_0+3) \qquad (3)\end{aligned}$$

The computation in (1) reaches success. In (2), $p$ commits to the choice $\texttt{!a}$ after some delay $\delta$; at this point, time cannot pass, and no synchronisation is possible. In (3), $p$ commits to $\texttt{!b}$ after 3 time units; here, the rightmost endpoint would offer $\texttt{?b}$, — but not in the time chosen by the leftmost endpoint.

**Compliance.**

We extend to the timed setting the standard progress-based compliance [1]. TSTs $p$ and $q$ are compliant when their composition never reaches a deadlock state.

**Definition 2.4** [Compliance] We say that $(p,\nu) \mid (q,\eta)$ is *deadlock* whenever $(i)$ both $p$ and $q$ are not $\mathbf{1}$, and $(ii)$ there is no $\delta$ such that $(p,\nu+\delta) \mid (q,\eta+\delta) \xrightarrow{\tau}$. We then write $(p,\nu) \bowtie (q,\eta)$ whenever:

$$(p,\nu) \mid (q,\eta) \to^* (p',\nu') \mid (q',\eta') \quad \text{implies} \quad (p',\nu') \mid (q',\eta') \text{ not deadlock}$$

$$\Gamma \vdash \mathbf{1} : \mathbb{V} \quad \text{[T-1]} \qquad \frac{\Gamma \vdash p_i : \mathcal{K}_i \qquad \text{for } i \in I}{\Gamma \vdash \sum_{i\in I} \mathtt{?a}_i\{g_i, T_i\}\,.\,p_i : \bigcup_{i\in I} {\downarrow}\left([\![g_i]\!] \cap \mathcal{K}_i[T_i]^{-1}\right)} \text{[T-+]}$$

$$\frac{\Gamma \vdash p_i : \mathcal{K}_i \qquad \text{for } i \in I}{\Gamma \vdash \bigoplus_{i\in I} \mathtt{!a}_i\{g_i, T_i\}\,.\,p_i : \left(\bigcup_{i\in I} {\downarrow}[\![g_i]\!]\right) \setminus \left(\bigcup_{i\in I} {\downarrow}\left([\![g_i]\!] \setminus \mathcal{K}_i[T_i]^{-1}\right)\right)} \text{[T-⊕]}$$

Fig. 2. Kind system for TSTs. (See [4] for the full set of judgements)

We say that $p$ and $q$ are *compliant* whenever $(p, \nu_0) \bowtie (q, \eta_0)$ (in short, $p \bowtie q$).

**Example 2.5** Let $p = \mathtt{?a}\{t < 5\}.\mathtt{!b}\{t < 3\}$. We have that $p$ is compliant with $q = \mathtt{!a}\{t < 2\}.\mathtt{?b}\{t < 3\}$, but it is not compliant with $q' = \mathtt{!a}\{t < 5\}.\mathtt{?b}\{t < 3\}$.

Compliance can be reduced to deadlock freedom in networks of timed automata.

**Theorem 2.6** *Compliance between TSTs is decidable.*

## 3 On duality and subtyping

In the untimed setting, every session type is compliant with its *dual* (obtained with a simple change of polarities [1]). This is not always true in the timed setting:

**Example 3.1** Consider $p_1 = \mathtt{!a}\{x \le 2\}.\mathtt{!b}\{x \le 1\}$ and $p_2 = \mathtt{!a}\{x \le 2\} \oplus \mathtt{!b}\{x \le 1\}.\mathtt{?a}\{x \le 0\}$. The TST $p_1$ is not compliant with $q_1 = \mathtt{?a}\{x \le 2\}.\mathtt{?b}\{x \le 1\}$: $p_1$ cannot perform $\mathtt{!b}$ if $\mathtt{!a}$ is performed after 1 time unit. Indeed, no TST is compliant with $p_1$. Note that $q_1 \bowtie \mathtt{!a}\{x \le 1\}.\mathtt{!b}\{x \le 1\}$. In $p_2$, a similar deadlock situation occurs if the $\mathtt{!b}$ branch is chosen, and so also $p_2$ does not admit a compliant.

We define a kind system which associates every $p$ with its *kind* $\mathcal{K}$ such that $p$ admits a compliant TST in all $\nu \in \mathcal{K}$.

**Definition 3.2 (Kind system)** Kind judgements $\Gamma \vdash p : \mathcal{K}$ are defined in Figure 2, where $\Gamma$ is a partial function which associates kinds with recursion variables.

Since $\mathbf{1}$ is compliant with itself, its kind is $\mathbb{V}$ ([T-1]). The kind of an exernal choice is the union of the kinds of its branches (rule [T-+]), which are the past of those clock valuations which satisfy both the guard and, after the reset, the kind of their continuation. Rule [T-⊕] computes the difference between the union of the past of the guards and those clock valuations which can satisfy a guard but not the kind of their continuations.

The kinds constructed by our inference rules can always be represented syntactically by guards [6], from which follows the following theorem.

**Theorem 3.3** *Kind inference is decidable.*

We now define the dual of kindable TSTs. Roughly, we turn internal choices into external ones (without changing guards nor resets), and external into internal, changing the guards so that the kind of continuations is preserved.

**Definition 3.4 (Dual)** For all kindable $p$ and kinding environments $\Gamma$, we define the TST $\mathrm{co}_\Gamma(p)$ (in short, $\mathrm{co}(p)$ when $\Gamma = \emptyset$) by the following equations (see [2,4] for the full set of equations):

$$\mathrm{co}_\Gamma\big(\textstyle\sum_{i\in I} \mathtt{?a}_i\{g_i, T_i\}\,.\,p_i\big) = \bigoplus_{i\in I} \mathtt{!a}_i\{g_i \wedge \mathcal{K}_i[T_i]^{-1}, T_i\}\,.\,\mathrm{co}_\Gamma(p_i) \quad \text{if } \Gamma \vdash p_i : \mathcal{K}_i$$

The following theorems state the soundness and completeness of the kind system.

**Theorem 3.5 (Soundness and completeness)** *Let* $\vdash p : \mathcal{K}$. *If* $\nu \in \mathcal{K}$, *then* $(p, \nu) \bowtie (\mathrm{co}(p), \nu)$. *If* $(p, \nu) \bowtie (q, \eta)$ *for some* $q, \eta$, *then* $\nu \in \mathcal{K}$.

We show that the dual is maximal w.r.t. the subtyping relation, like in the untimed setting. We start by defining the semantic subyting preorder, which is a model of the Gay and Hole subtyping relation (in reverse order) for session types [1].

**Definition 3.6 (Semantic subtyping)** For all TSTs $p$, we define the set $p^{\bowtie}$ as $\{q \mid p \bowtie q\}$. Then, we define the relation $p \sqsubseteq q$ whenever $p^{\bowtie} \supseteq q^{\bowtie}$.

**Theorem 3.7** $q \bowtie p \implies q \sqsubseteq \mathrm{co}(p)$

The following theorem reduces the problem of deciding $p \sqsubseteq q$ to that of checking compliance between $p$ and $\mathrm{co}(q)$, from which follows decidability of subtyping.

**Theorem 3.8** *If* $q$ *admits a compliant, then:* $p \sqsubseteq q \iff p \bowtie \mathrm{co}(q)$.

## 4 Conclusions

We have studied a theory of timed session types, featuring timed synchronous communication between two endpoints. We have defined a decidable notion of compliance between TSTs, a decidable procedure to detect when a TST admits a compliant, and a decidable subtyping relation.

**Acknowledgments.**

Work partially supported by Aut. Reg. of Sardinia L.R.7/2007 CRP-17285 (TRICS), P.I.A. 2010 ("Social Glue"), P.O.R. Sardegna F.S.E. Operational Programme of the Aut. Reg. of Sardinia, EU Social Fund 2007-13 – Axis IV Human Resources, Objective l.3, Line of Activity l.3.1), by MIUR PRIN 2010-11 project "Security Horizons", and by EU COST Action IC1201 "Behavioural Types for Reliable Large-Scale Software Systems" (BETTY).

## References

[1] F. Barbanera and U. de'Liguoro. Sub-behaviour relations for session-based client/server systems. *Math. Struct. in Comp. Science*, pages 1–43, 1 2015.

[2] M. Bartoletti, T. Cimoli, M. Murgia, A. S. Podda, and L. Pompianu. Compliance and subtyping in timed session types. In *FORTE 2015*, pages 161–177, 2015.

[3] M. Bartoletti, T. Cimoli, M. Murgia, A. S. Podda, and L. Pompianu. A contract oriented middleware. Technical report, 2015. co2.unica.it.

[4] M. Bartoletti, T. Cimoli, M. Murgia, A. S. Podda, and L. Pompianu. Compliance and subtyping in timed session types. Technical report, 2015. co2.unica.it.

[5] L. Bocchi, W. Yang, and N. Yoshida. Timed multiparty session types. In *CONCUR*, pages 419–434, 2014.

[6] T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Inf. Comput.*, 111(2):193–244, 1994.

[7] K. Takeuchi, K. Honda, and M. Kubo. An interaction-based language and its typing system. In *PARLE*, 1994.