Timed Interfaces

Stefan Stattelmann

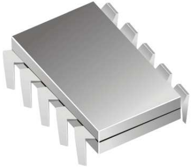
Seminar Games in Verification and Synthesis
Summer Term 2008
Saarland University

original paper by

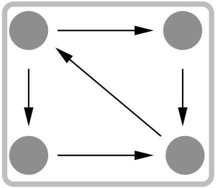
L. de Alfaro, T. Henzinger and M. Stoelinga

Motivation

Component



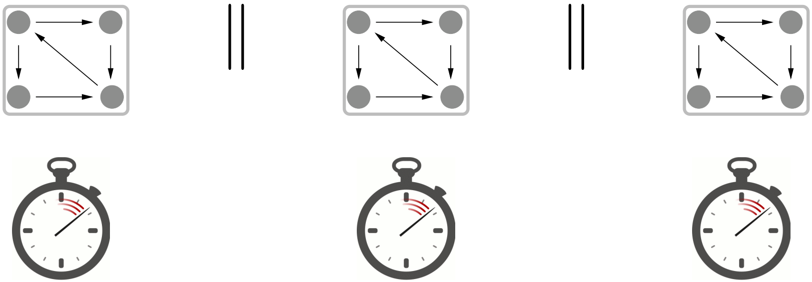
Interface



Timing
Requirements

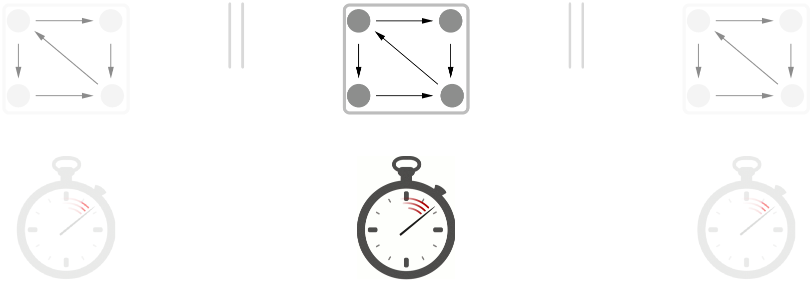
- complex real-time systems
- component based design
- interface describes component behavior

Motivation



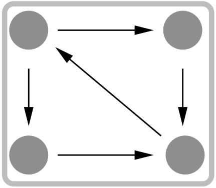
- model component interaction
- type system for interfaces

Motivation

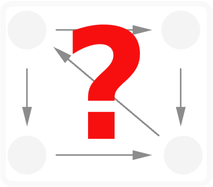


- well-formed?
- input assumptions: expected use
- output guarantees: correct input $\Rightarrow$ correct output

Motivation



$\|$



System vs. Environment

well-formed $\iff$ $\exists$ Environment that fulfils input assumptions

Timed Interface Theory

We are interested in

- Well-formedness
- Compatibility
- Composition

Talk Outline

Composition and Compatibilty

Timed Interfaces as Timed Games

Timed Interface Automata

Solving Timed Games

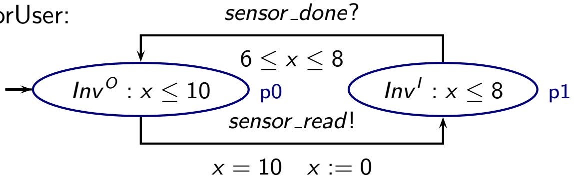
Definition: Timed Interface

Timed interface $\mathcal{P} = (S_{\mathcal{P}}, s^{init}_{\mathcal{P}}, Acts^{I}_{\mathcal{P}}, Acts^{O}_{\mathcal{P}}, \rho^{I}_{\mathcal{P}}, \rho^{O}_{\mathcal{P}})$ with:

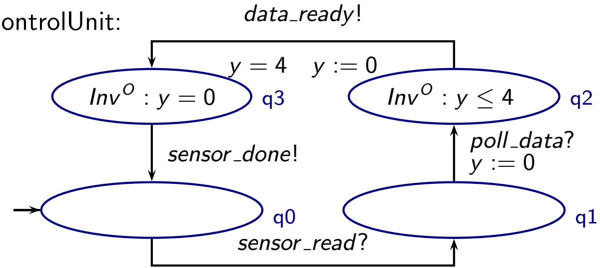
- $S_{\mathcal{P}}$: set of states
- $s^{init}_{\mathcal{P}} \in S_{\mathcal{P}}$: initial state
- $Acts^{I}_{\mathcal{P}}$ and $Acts^{O}_{\mathcal{P}}$: immediate input/output actions
- $\mathbb{T}$: set of timed actions ($\mathbb{T} = \mathbb{R}_{\geq 0}$ or $\mathbb{T} = \mathbb{N}$)
- $Acts_{\mathcal{P}} = Acts^{I}_{\mathcal{P}} \cup Acts^{O}_{\mathcal{P}}$
- $\Gamma^{I}_{\mathcal{P}} = Acts^{I}_{\mathcal{P}} \cup \mathbb{T}$: set of all input actions
- $\Gamma^{O}_{\mathcal{P}} = Acts^{O}_{\mathcal{P}} \cup \mathbb{T}$: set of all output actions
- $\rho^{I}_{\mathcal{P}} \subseteq S_{\mathcal{P}} \times \Gamma^{I}_{\mathcal{P}} \times S_{\mathcal{P}}$: input transition relation
- $\rho^{O}_{\mathcal{P}} \subseteq S_{\mathcal{P}} \times \Gamma^{O}_{\mathcal{P}} \times S_{\mathcal{P}}$: output transition relation

Example

SensorUser:



SensorControlUnit:



Composability

Timed interfaces $\mathcal{P}$ and $\mathcal{Q}$ are composable if

- $\mathcal{P}$ and $\mathcal{Q}$ are well-formed
- $Acts^O_{\mathcal{P}} \cap Acts^O_{\mathcal{Q}} = \emptyset$

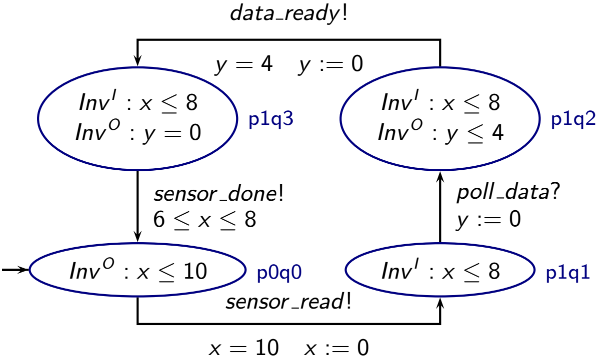
Shared actions: $shared(\mathcal{P}, \mathcal{Q}) := Acts_{\mathcal{P}} \cap Acts_{\mathcal{Q}}$

Interface Product

For $\mathcal{P}$ and $\mathcal{Q}$ composable timed interfaces

- $S_{\mathcal{P}\otimes\mathcal{Q}} = S_{\mathcal{P}} \times S_{\mathcal{Q}}$
- $s^{init}_{\mathcal{P}\otimes\mathcal{Q}} = (s^{init}_{\mathcal{P}}, s^{init}_{\mathcal{Q}})$.
- $Acts^{I}_{\mathcal{P}\otimes\mathcal{Q}} = Acts^{I}_{\mathcal{P}} \cup Acts^{I}_{\mathcal{Q}} \setminus shared(\mathcal{P},\mathcal{Q})$
- $Acts^{O}_{\mathcal{P}\otimes\mathcal{Q}} = Acts^{O}_{\mathcal{P}} \cup Acts^{O}_{\mathcal{Q}}$.
- $\rho^{I}_{\mathcal{P}\otimes\mathcal{Q}} = \{((s_1,s_2),\alpha,(s'_1,s'_2)|(s_1,\alpha,s'_1) \in \rho^{I}_{\mathcal{P}}$ *and* $(s_2,\beta,s'_2) \in \rho^{I}_{\mathcal{Q}}, \beta = \alpha$ *if* $\alpha \in Acts^{I}_{\mathcal{Q}}$ *or* $\beta = 0$ *otherwise*$\} \cup \{((s_1,s_2),\alpha,(s'_1,s'_2)|(s_2,\alpha,s'_2) \in \rho^{I}_{\mathcal{Q}}$ *and* $(s_1,\beta,s'_1) \in \rho^{I}_{\mathcal{P}}$, $\beta = \alpha$ *if* $\alpha \in Acts^{I}_{\mathcal{P}}$ *or* $\beta = 0$ *otherwise*$\}$
- $\rho^{O}_{\mathcal{P}\otimes\mathcal{Q}} = \{((s_1,s_2),\alpha,(s'_1,s'_2)|(s_1,\alpha,s'_1) \in \rho^{O}_{\mathcal{P}}$ *and* $(s_2,\beta,s'_2) \in \rho^{I}_{\mathcal{Q}}, \beta = \alpha$ *if* $\alpha \in Acts^{I}_{\mathcal{Q}}$ *or* $\beta = 0$ *otherwise*$\} \cup \{((s_1,s_2),\alpha,(s'_1,s'_2)|(s_2,\alpha,s'_2) \in \rho^{O}_{\mathcal{Q}}$ *and* $(s_1,\beta,s'_1) \in \rho^{I}_{\mathcal{P}}$, $\beta = \alpha$ *if* $\alpha \in Acts^{I}_{\mathcal{P}}$ *or* $\beta = 0$ *otherwise*$\}$

Example II



SensorUser $\otimes$ SensorControlUnit

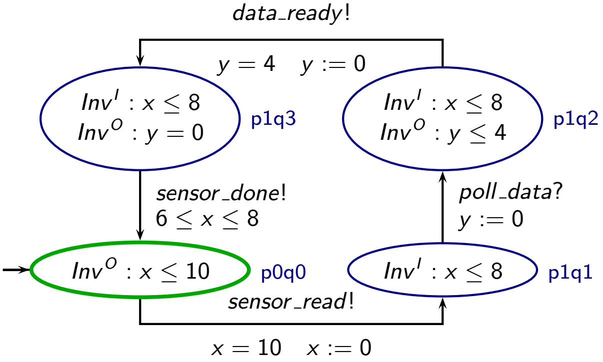
Error States

Problems with interface product:

- timing requirements of components not synchronized
- one component could create output that cannot be accepted

  $\Rightarrow$ error state

Example III

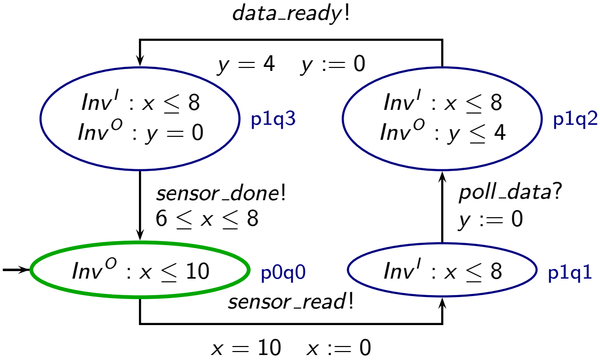


$x = 0$

$y = 0$

SensorUser $\otimes$ SensorControlUnit

Example III



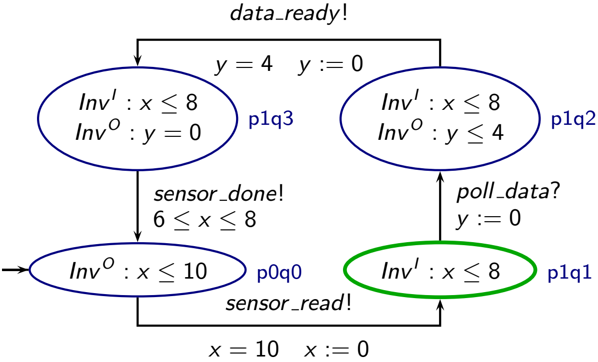
SensorUser $\otimes$ SensorControlUnit

$x = 10$

$y = 10$

$sensor\_read!$

Example III

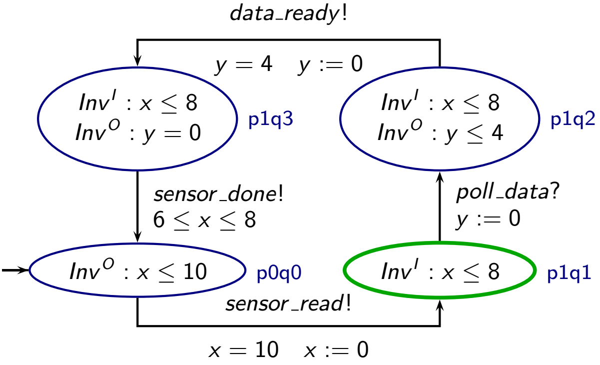


SensorUser $\otimes$ SensorControlUnit

$x = 0$

$y = 10$

Example III



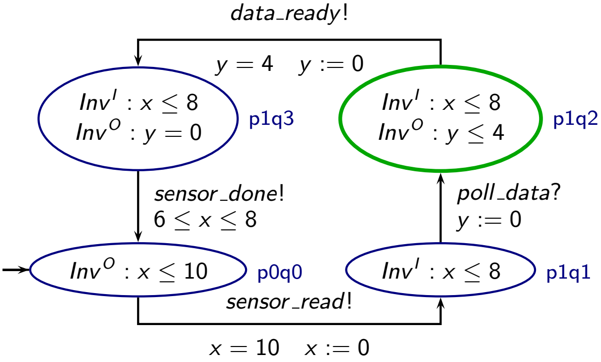
SensorUser $\otimes$ SensorControlUnit

$x = 8$

$y = 18$

$poll\_data?$

Example III



$x = 8$

$y = 0$

SensorUser $\otimes$ SensorControlUnit

Timed Interfaces as Timed Games [dAHS02]

- model interface as two-player game
- Player Input: Environment
- Player Output: Component
- Moves:

timed actions	wait for event
immediate actions	trigger event

- goal: time diverges or other player blocks

Definition: Moves and Outcome

Possible moves for player $\gamma \in \{I, O\}$ in state $s \in S_{\mathcal{P}}$

- $\Gamma^{\gamma}_{\mathcal{P}}(s) = \{\alpha \in \Gamma^{\gamma}_{\mathcal{P}} \mid \exists s' \in S_{\mathcal{P}}.(s, \alpha, s') \in \rho^{\gamma}_{\mathcal{P}}\}$
- $\Gamma^{\gamma}_{\mathcal{P}}(s) \neq \emptyset \implies (s, 0, s) \in \rho^{\gamma}_{\mathcal{P}}$
- $\Gamma^{\gamma}_{\mathcal{P}}(s) = \emptyset \implies$ player γ loses because he blocks

For $s \in S_{\mathcal{P}}$, $\alpha_I \in \Gamma^{I}_{\mathcal{P}}(s)$, $\alpha_O \in \Gamma^{O}_{\mathcal{P}}(s)$ and $bl \in \{I, O\}$,
define outcome $\delta_{\mathcal{P}}(s, \alpha_I, \alpha_O) := (\alpha, s', bl)$

- α_I, $\alpha_O \in \mathbb{T} \implies \alpha = min\{\alpha_I, \alpha_O\}$
  $bl = I$ if $\alpha_I < \alpha_O$, $bl = O$ otherwise (asymmetric!).
- If $\alpha_I \in Acts_{\mathcal{P}}$ and $\alpha_O \in \mathbb{T}$, then $\alpha = \alpha_I$ and $bl = I$.
- If $\alpha_I \in \mathbb{T}$ and $\alpha_O \in Acts_{\mathcal{P}}$, then $\alpha = \alpha_O$ and $bl = O$.
- Ifα_I, $\alpha_O \in Acts_{\mathcal{P}}$, choose $\alpha = \alpha_I$ and $bl = I$ or $\alpha = \alpha_O$ and $bl = O$ nondeterministically.

Definition: Strategies and Reachable States

Strategy $\pi^\gamma : S^*_{\mathcal{P}} \rightarrow \mathit{\Gamma}^\gamma_{\mathcal{P}}$ for player $\gamma \in \{I, O\}$ assigns move $\pi^\gamma(\bar{s}) \in \mathit{\Gamma}^\gamma_{\mathcal{P}}(s)$ to every $\bar{s} \in S^*_{\mathcal{P}}$ whose final state is s, if $\mathit{\Gamma}^\gamma_{\mathcal{P}} \neq \emptyset$. Otherwise, $\pi^\gamma(\bar{s})$ is undefined.

State $s \in S_{\mathcal{P}}$ is reachable if there are strategies π^I and π^O for player I and O s.t s is visited during game starting from $s^{init}_{\mathcal{P}}$ that is played according to π^I and π^O.

Well-formedness

Liveness

- sum of timed actions must not converge
  (no Zeno behavior [MPS95])
- player must not block game

Blocking

- player runs out of moves
- one player always plays, but time does not converge

A timed interface is well-formed if there is strategy for both players to let time diverge or blame the other player for blocking the game.

Combining Interfaces

- one component might produce output that cannot be accepted by others $\Rightarrow$ error state
- optimistic approach: restrict interface to make it work
- can't change components $\Rightarrow$ change use (environment)
- guarantee safety by avoiding error states

Error States

Immediate error state:
$(s,t) \in S_{\mathcal{P}\otimes\mathcal{Q}}$ with $\alpha \in \mathit{shared}(\mathcal{P},\mathcal{Q})$ such that
$\exists s' : (s,\alpha,s') \in \rho^O_{\mathcal{P}}$ and $\forall t' : (t,\alpha,t') \notin \rho^I_{\mathcal{Q}}$ or
$\exists t' : (t,\alpha,t') \in \rho^O_{\mathcal{Q}}$ and $\forall s' : (s,\alpha,s') \notin \rho^I_{\mathcal{P}}$.
set of all immediate error states: $\mathit{i\text{-}errors}(\mathcal{P},\mathcal{Q}) \subseteq S_{\mathcal{P}\otimes\mathcal{Q}}$.

Time error state:
$(s,t) \in S_{\mathcal{P}\otimes\mathcal{Q}}$ reachable in $\mathcal{P}\otimes\mathcal{Q}$, but there is no strategy to win the game for player I in $S_{\mathcal{P}\otimes\mathcal{Q}} \setminus \mathit{i\text{-}errors}(\mathcal{P},\mathcal{Q})$.
set of all time error states: $\mathit{t\text{-}errors}(\mathcal{P},\mathcal{Q})$

Interface Composition

well-formed, composable interfaces $\mathcal{P}$ and $\mathcal{Q}$ are compatible if $(s_{\mathcal{P}}^{init}, s_{\mathcal{Q}}^{init}) \notin \textit{t-errors}(\mathcal{P}, \mathcal{Q})$

composition $\mathcal{P} \parallel \mathcal{Q}$ defined like $\mathcal{P} \otimes \mathcal{Q}$

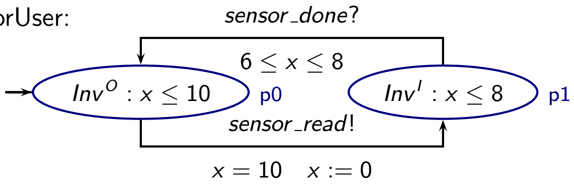
except for input transition relation:

$U = S_{\mathcal{P} \otimes \mathcal{Q}} \setminus \textit{t-errors}(\mathcal{P}, \mathcal{Q})$

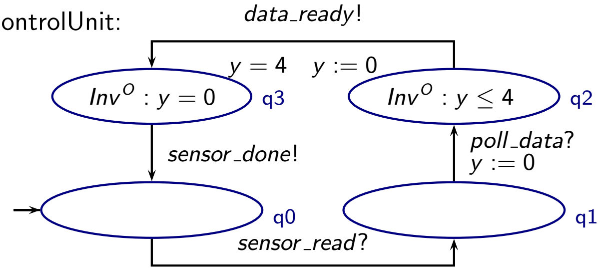
$\rho^I_{\mathcal{P} \parallel \mathcal{Q}} = \rho^I_{\mathcal{P} \otimes \mathcal{Q}} \cap (U \times \textit{Acts}^I_{\mathcal{P} \otimes \mathcal{Q}} \times U)$

Examples revisited

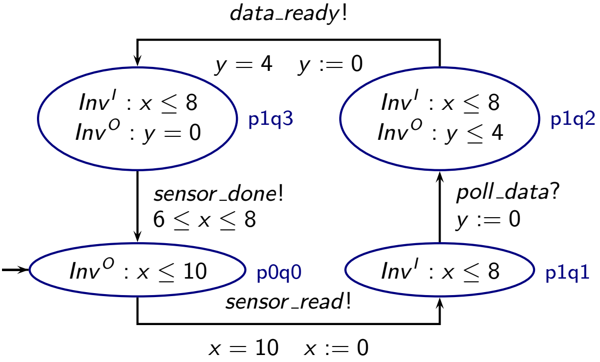
SensorUser:



SensorControlUnit:

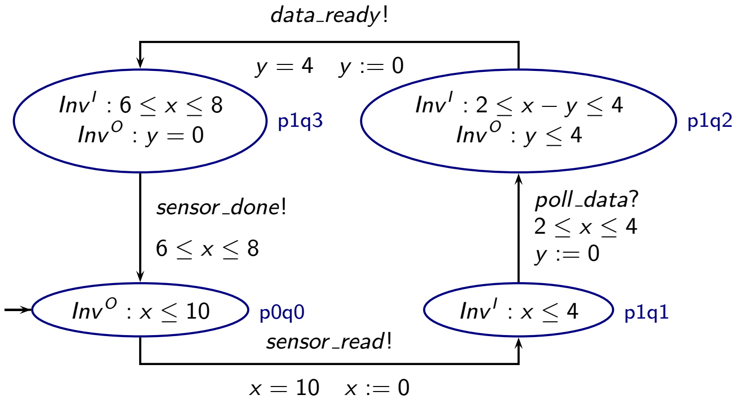


Examples revisited



SensorUser $\otimes$ SensorControlUnit

Examples revisited



SensorUser $\|$ SensorControlUnit

Timed Interface Automata

- finite representation for timed interfaces
- similiar to timed automata ([AD94])
- reuse existing algorithms for calculating live states, composition and checking well-formedness

Definition: Timed Interface Automata

Timed interface automaton
$\mathcal{A} = (Q_{\mathcal{A}}, q_{\mathcal{A}}^{init}, \mathcal{X}_{\mathcal{A}}, Acts_{\mathcal{A}}^{I}, Acts_{\mathcal{A}}^{O}, Inv_{\mathcal{A}}^{I}, Inv_{\mathcal{A}}^{O}, \rho_{\mathcal{A}})$ with:

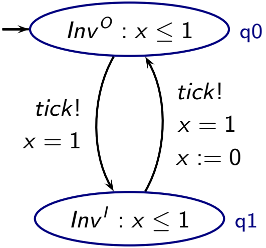
- $Q_{\mathcal{A}}$: set of *locations*
- $q_{\mathcal{A}}^{init} \in Q_{\mathcal{A}}$: initial location
- $\mathcal{X}_{\mathcal{A}}$: set of clocks
- $Acts_{\mathcal{A}}^{I}$ and $Acts_{\mathcal{A}}^{O}$: sets of input and output actions
- $Inv_{\mathcal{A}}^{I} : Q_{\mathcal{A}} \mapsto \Xi[\mathcal{X}_{\mathcal{A}}]$ and $Inv_{\mathcal{A}}^{O} : Q_{\mathcal{A}} \mapsto \Xi[\mathcal{X}_{\mathcal{A}}]$ map an input/output invariant to each location
- $\rho_{\mathcal{A}} \subseteq Q_{\mathcal{A}} \times \Xi[\mathcal{X}_{\mathcal{A}}] \times Acts_{\mathcal{A}} \times 2^{\mathcal{X}_{\mathcal{A}}} \times Q_{\mathcal{A}}$ transition relation

Solving Timed Games

Checking for winning strategy of I

- compose automaton with Tick automaton
- check if there is strategy for $\Box\Diamond q_1 \vee \Diamond\Box bl = O$
- use algorithm for untimed games
- similar for player Output

Tick automaton



Solving Timed Games

Reachable states

- definable by clock conditions
- use algorithms for timed automata [AD94]

Well-formedness

Check: *reachable* $\implies$ *both players have winning strategy*

Summary

- new approach: model interface as asymmetric game
- restrict moves of input to guarantee safety and liveness
- optimistic
- automata representation allows use of existing algorithms

$\Rightarrow$ better model for interaction of real-time components

Thank you for your attention!

Questions?

References

Rajeev Alur and David L. Dill.
A theory of timed automata.
Theoretical Computer Science, 126(2):183–235, 1994.

L. de Alfaro, T. Henzinger, and M. Stoelinga.
Timed interfaces, 2002.

Oded Maler, Amir Pnueli, and Joseph Sifakis.
On the synthesis of discrete controllers for timed systems (extended abstract), 1995.