Weighted Timed Automata and Timed Migration in Distributed Systems

Bogdan Aman Gabriel Ciobanu

Romanian Academy, Institute of Computer Science, Iași, Romania
and "A.I.Cuza" University, Iași, Romania

WATA 2018
May 24, 2018

Outline

Introduction

- TiMo is a calculus describing distributed systems in which processes are able to migrate between a number of explicit locations.
- Processes can move from location to location to communicate with other processes (rather than using the client/server method).
- Two processes may communicate if are present at the same location.
- Timing constraints over migration and communication are used to co-ordinate processes in time and space.
- Timing constraints for migration allow one to specify a temporal interval after which a mobile process must move to another location.
- Timing constraints for communication allow interaction any time in the temporal interval.

- We work here with cTiMo (cost Timed Mobility), an extension of TiMo with costs over locations and actions.

Syntax and Semantics of cTiMo

cTiMo Syntax

$$
\begin{array}{llll}
\textit{Processes} & P, Q & ::= & go_c^t\ l \text{ then } P \mid \hfill \text{(move)} \\
 & & & a_c^{\Delta t}!\langle v\rangle \text{ then } P \text{ else } Q \mid \hfill \text{(output)} \\
 & & & a_c^{\Delta t}?(u) \text{ then } P \text{ else } Q \mid \hfill \text{(input)} \\
 & & & 0 \mid \hfill \text{(termination)} \\
 & & & id(v) \hfill \text{(recursion)} \\
\textit{Located processes} & L & ::= & l[[_c\ P]] \\
\textit{Networks} & N & ::= & L \mid L \mid N
\end{array}
$$

- A timer in cTiMo is denoted either by:
  - t (for migration actions) or
  - Δt (for output and input actions).
- When it is associated with a migration process $go_c^t\ l$ then P it indicates that process P moves from the current location to location l after t time units, while the cost of movement is c.

Syntax and Semantics of cTiMo

cTiMo Syntax

Processes	P, Q	$::=$	$go^{t}_{c}\ l$ then P $\mid$	(move)
			$a^{\Delta t}_{c}!\langle v\rangle$ then P else Q $\mid$	(output)
			$a^{\Delta t}_{c}?(u)$ then P else Q $\mid$	(input)
			0 $\mid$	(termination)
			$id(v)$	(recursion)
Located processes	L	$::=$	$l[[_c\ P]]$	
Networks	N	$::=$	$L \mid L \mid N$	

- Since l can be a variable with its value assigned dynamically through communication with other processes, this form of migration supports a flexible scheme for the movement of processes between locations.
- Thus, the behaviour can adapt to various changes of the environment.

cTiMo Syntax

Processes	P, Q	$::=$	$go^{t}_{c}\ l$ then P $\mid$	(move)
			$a^{\Delta t}_{c}!\langle v\rangle$ then P else Q $\mid$	(output)
			$a^{\Delta t}_{c}?(u)$ then P else Q $\mid$	(input)
			0 $\mid$	(termination)
			$id(v)$	(recursion)
Located processes	L	$::=$	$l[[_c\ P]]$	
Networks	N	$::=$	$L \mid L \mid N$	

- A timer Δt associated with an output/input communication process makes the channel a available for communication for at most t time units with the associated cost c.
- In both cases, if the interaction does not happen in the interval $[0, t]$, the process gives up, and continues as the alternative process Q.

cTiMo Operational Semantics

- We use multiset labelled transitions of form $N \xrightarrow{\Lambda, C} N'$ where:
  - the multiset Λ indicates the actions executed in parallel in one step;
  - the multiset C of costs indicates the costs of each of the actions from Λ.
- The order of the costs in C depends on the order of actions in Λ, as each action has an unique execution cost.
- When the multiset Λ contains only one action λ with cost c, we simply write $N \xrightarrow{\lambda, c} N'$.
- The transitions of form $N \overset{t, C}{\rightsquigarrow} N'$ represent a time step of length t and costs C for a network N.

cTiMo Operational Semantics

(Move0) $$l[[_c\ \mathtt{go}^{0}_{c''}\ l'\ \mathtt{then}\ P]] \xrightarrow{l \triangleright l', c''} l'[[_{c'}\ P]]$$

(Com) $$l[[_c\ a^{\Delta t}_{c_1}!\langle v\rangle\ \mathtt{then}\ P\ \mathtt{else}\ Q]] \mid l[[_c\ a^{\Delta t'}_{c_2}?(u)\ \mathtt{then}\ P'\ \mathtt{else}\ Q']] \xrightarrow{!v@l|?u@l, c_1|c_2} l[[_c\ P]] \mid l[[_c\ \{v/u\}P']]$$

(DPut) $$\frac{t \geq t' \geq 0}{l[[_c\ a^{\Delta t}_{c'}!\langle v\rangle\ \mathtt{then}\ P\ \mathtt{else}\ Q]] \overset{t', c*t'}{\rightsquigarrow} l[[_c\ a^{\Delta t-t'}_{c'}!\langle v\rangle\ \mathtt{then}\ P\ \mathtt{else}\ Q]]}$$

(Put0) $$l[[_c\ a^{\Delta 0}_{c'}!\langle v\rangle\ \mathtt{then}\ P\ \mathtt{else}\ Q]] \xrightarrow{a!^{\Delta 0}@l, c'} l[[_c\ Q]]$$

(DPar) $$\frac{N_1 \overset{t, C_1}{\rightsquigarrow} N_1' \qquad N_2 \overset{t, C_2}{\rightsquigarrow} N_2' \qquad N_1 \mid N_2 \not\rightarrow}{N_1 \mid N_2 \overset{t, C_1|C_2}{\rightsquigarrow} N_1' \mid N_2'}$$

Syntax and Semantics of cTiMo

- A complete computational step is captured by a derivation:

$$N \xrightarrow{\Lambda, C} N_1 \overset{t, C'}{\rightsquigarrow} N'.$$

Theorem

For any networks N, N' and N'' we have the following properties:

(idle) $N \overset{0,0}{\rightsquigarrow} N$;

(time-determinism *If* $N \overset{t,C}{\rightsquigarrow} N'$ *and* $N \overset{t,C}{\rightsquigarrow} N''$, *then* $N' \equiv N''$;

(time-continuity) $N \overset{(t+t'),C''}{\rightsquigarrow} N'$ *if and only if there is a* N'' *such that* $N \overset{t,C}{\rightsquigarrow} N''$ *and* $N'' \overset{t',C'}{\rightsquigarrow} N'$, *where* $C'' = C' + C''$ *(component-wise sum).*

- As the cost cannot be tested upon in any guard or invariants, it does not influence the behaviour of the system.
- This means that a network N is guaranteed to proceed even when costs are omitted or null (i.e., $erase(N)$).

Theorem

1. *If* $N \xrightarrow{\Lambda, C} N'$, *then* $erase(N) \xrightarrow{\Lambda} erase(N')$.
2. *If* $N \overset{t, C'}{\rightsquigarrow} N'$, *then* $erase(N) \overset{t}{\rightsquigarrow} erase(N')$.

Example

- Consider the network $N = L_1 \mid L_2 \mid L_3$, where L_1 and L_2 are defined below, and L_3 can be of any form.
  $L_1 = l[[_2\, \mathrm{go}_1^3\, l' \text{ then } P_1]]$ and
  $L_2 = l'[[_3\, a_1^{\Delta 5}?(x) \text{ then } 0 \text{ else } 0]]$, where
  $P_1 = a_2^{\Delta 2}!\langle v\rangle \text{ then } P_2 \text{ else } P_3$
  $P_2 = \mathrm{go}_1^2\, l'' \text{ then } 0$
  $P_3 = \mathrm{go}_1^1\, l'' \text{ then } 0.$
  In the above network we assume that the cost of location l'' is 3.

Example

- If L_3 does not interfere with L_1 and L_2 executions, a possible execution of N can lead to the following finite executions of L_1 andL_2:

$$L_1 = l[[_2\, go^3_1\, l' \text{ then } P_1]]$$
$$\xrightarrow{3,6} l[[_2\, go^0_1\, l' \text{ then } P_1]]$$
$$\xrightarrow{l \triangleright l',1} l'[[_3\, P_1]] = l'[[_3\, a_2^{\Delta 2}!\langle v\rangle \text{ then } P_2 \text{ else } P_3]]$$
$$\xrightarrow{!\langle v\rangle @ l',2} l'[[_3\, P_2]] = l'[[_3\, go^2_1\, l'' \text{ then } 0]]$$
$$\xrightarrow{2,6} l'[[_3\, go^0_1\, l'' \text{ then } 0]]$$
$$\xrightarrow{l' \triangleright l'',1} l''[[_3\, 0]]$$
$$L_2 = l'[[_3\, a_1^{\Delta 5}?(x) \text{ then } 0 \text{ else } 0]]$$
$$\xrightarrow{3,9} l'[[_3\, a_1^{\Delta 2}?(x) \text{ then } 0 \text{ else } 0]]$$
$$\xrightarrow{?(x) @ l',1} l'[[_3\, 0]]$$

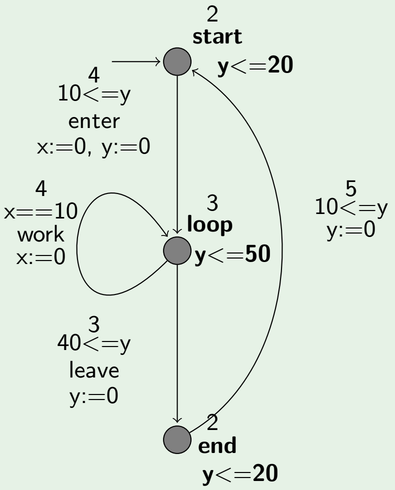
The costs for the executions of L_1 and L_2 are 16 and 10, respectively.

Definition

A weighted timed automaton $\mathcal{A}$ over X (finite set of clocks) and AP (finite set of atomic propositions) is a tuple $(L, l_0, T, \lambda, cost)$, where:

- L is a finite set of locations,
- $l_0 \in L$ is the initial location,
- $T \subseteq L \times \mathcal{C}(X) \times \Sigma \times 2^X \times L$ is a finite set of transitions,
- $\lambda : L \to 2^{\mathrm{AP}}$ a labelling function,
- $cost : L \cup T \to \mathbb{N}$ assigns costs to locations and transitions.

A Weighted Timed Automata



A network state is a pair $\langle l, v\rangle$, where l denotes a vector of current locations of the network (one for each automaton), and v is a clock assignment storing the current values of all network clocks.

Definition

The operational semantics of a weighted timed automaton is given by:

- *delay transitions*: $(l, v) \xrightarrow{\delta(d)} (l, v + d)$ if $d \in \mathbb{R}_+$;
- *discrete transitions*: $(l, v) \xrightarrow{tr} (l', v')$ if exists a transition $tr = (l, g, \sigma, Y, l') \in T$ such that $v \models g$. $v' = [Y \leftarrow 0]v$ and $\sigma \in \Sigma$.

For each step there is associated a cost defined by:

- $cost((l, v) \xrightarrow{\delta(d)} (l, v + d)) = cost(l) \cdot d$;
- $cost((l, v) \xrightarrow{tr} (l', v')) = cost(tr)$.

Network of weighted timed automata

- Is the parallel composition $\mathcal{A}_1 \mid \ldots \mid \mathcal{A}_n$ of a set of timed automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ combined into a single system.
- Synchronous communication inside the network is by handshake synchronisation of input and output actions.
- In this case, the action alphabet Σ consists of
  - $a?$ symbols (for input actions),
  - $a!$ symbols (for output actions),
  - τ symbols (for internal actions).
- A network can perform:
  - delay transitions (delay for some time),
  - action transitions (following an enabled edge).

Translating cTiMo into Weighted Timed Automaton

cTiMo Network decomposition

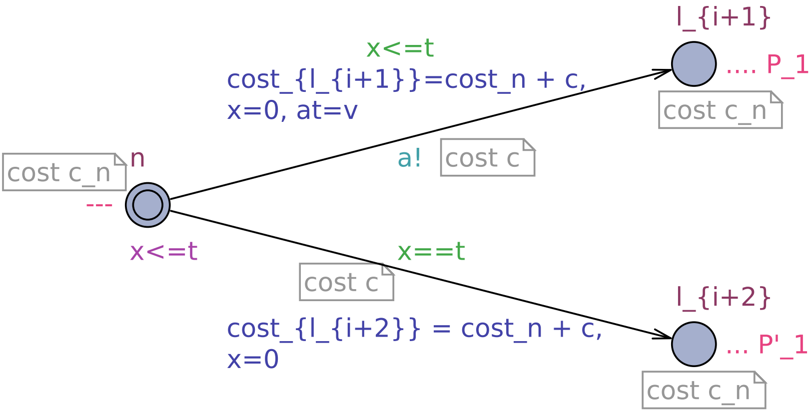
A given network N can always be transformed into a finite parallel composition of located processes of the form $l_1[[_{c_1} P_1]] \mid \dots \mid l_n[[_{c_n} P_n]]$ such that no process P_i has the parallel composition operator at its topmost level.

Construction

- Given a component $l[[_c P]]$ of an cTiMo network, we translate it into a weighted timed automaton $\mathcal{A} = (L, l_0, T, \lambda, cost)$ with a local clock x, where $L = \{l_0\}$, $T = \emptyset$, $\lambda(l_0) = \emptyset$ and $cost(l_0) = c$.
- The nodes of this weighted timed automata are labelled starting from the current location of the located process P.
- The components L, T, λ and $cost$ are updated depending on the structure of process P.

Translating cTIMO into Weighted Timed Automaton

$P = a_c^{\Delta t}!\langle v\rangle$ then P_1 else P_1'



Definition

A symmetric relation $\sim$ over cTiMo networks and their corresponding weighted timed automata is a bisimulation if, whenever it holds that $(N, (\mathcal{A}, \langle l_N, v_N\rangle)) \in \sim$, we have

- if $N \xrightarrow{\lambda,c} N'$, then $\langle l_N, v_N\rangle \xrightarrow{tr_\lambda} \langle l_{N'}, v_{N'}\rangle$ and $(N', (\mathcal{A}, \langle l_{N'}, v_{N'}\rangle)) \in \sim$ for some N', where $cost(tr_\lambda) = c$;
- if $N \overset{d,c}{\rightsquigarrow} N'$, then $\langle l_N, v_N\rangle \xrightarrow{\delta(d)} \langle l_{N'}, v_{N'}\rangle$ and $(N', (\mathcal{A}, \langle l_{N'}, v_{N'}\rangle)) \in \sim$ for some N', where $v_{N'} = v_N + d$ and $cost(\delta(d)) = c$.

Theorem

Given a cTiMo network N, there exists a network $\mathcal{A}_N$ of parallel timed automata having a bisimilar behaviour. Formally, $N \sim (\mathcal{A}, \langle l_N, v_N\rangle)$.

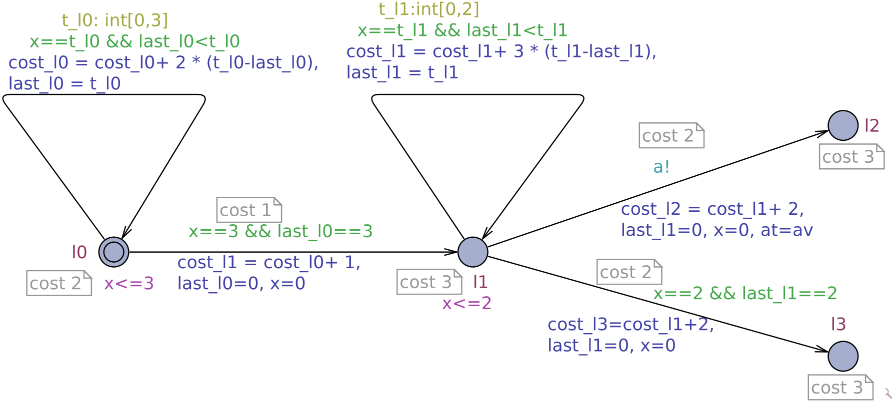
Example

$L_1 = l[[{}_2\,\mathrm{go}_1^3\ l' \text{ then } P_1]]$ where

$P_1 = a_2^{\Delta 2}!\langle v\rangle \text{ then } P_2 \text{ else } P_3$

$P_2 = \mathrm{go}_1^2\ l'' \text{ then } 0$

$P_3 = \mathrm{go}_1^1\ l'' \text{ then } 0.$



Verification of cTiMo Networks in Uppaal

Uppaal can be used to check temporal properties of networks of weighted timed automata, properties expressed in CTL (Computation Tree Logic).

Example

We performed some verifications in Uppaal for the running example.

- E$\langle\ \rangle$ $\mathrm{L1.cost\_l2} + \mathrm{L2.cost\_l2} == 19$

  This formulae is used to check whether there is an evolution such that the cost to reach location $L1.l2$, added to the the cost to reach location $L2.l2$ is equal to 19.

- A[ ] $\mathrm{L1.cost\_l2} <= 9$

  This is used to verify that, whatever are the interactions between the involved located processes, the cost to reach location $l2$ of $L1$ is always less than 9.

- E$[<= 6; 100000](\max : \mathrm{L1.cost_l2})$

  This is used to estimate the maximum value of $cost\_l2$ of $L1$ by performing 100000 simulations no longer than 6 time units. As expected, the value is 9.

Conclusion

- We presented cTIMO, an extension with costs of TIMO, a calculus suitable to describe complex distributed systems with mobility.
- Given a cTIMO network N, there exists a network $\mathcal{A}_N$ of parallel timed automata having a bisimilar behaviour, i.e., $N \sim (\mathcal{A}, \langle l_N, v_N \rangle)$.
- We used a running example of applying cTIMO, illustrating that cTIMO provides an appropriate framework for modelling and reasoning about costs in distributed systems with migration and communication.
- We shown how we can model and verify qualitative and quantitative properties of cTIMO networks by using the software tool UPPAAL.