Dynamic and Heterogeneous Timed Systems

José Luiz Fiadeiro

Department of Computer Science, Royal Holloway University of London, UK

Joint work with Antónia Lopes (U Lisbon),
Benoît Delahaye (U Nantes) and Axel Legay (INRIA Rennes)

IFIP WG1.3, January 11, 2017

Motivation

Dynamic and heterogeneous timed systems

Many software applications operating in cyberspace need to connect, dynamically, to other software systems.

For example, systems for congestion avoidance or coordination of self-driven convoys of cars need to be able to accommodate interconnections that are established at run time between components in ways that cannot be pre-determined at design time.

Components interconnected at run time will be likely to operate over platforms with different time granularities (clock periods), resulting in a timed heterogeneous system.

This talk

Summary

We investigate a suitable component algebra for such dynamic and timed heterogeneous systems, i.e., formal methods through which we can understand and reason in a compositional way about the behaviour of such systems.

Along the way, we challenge some of the acquired practices in algebraic development techniques, including the meaning of composition in the context of component-based systems.

TIOA

Timed I/O automata

A TIOA is a tuple $\mathcal{A} = \langle Loc, l^\star, \mathbb{C}, v^\star, E, Act, Inv \rangle$ where:

- Loc is a finite set of locations;
- $l^\star \in Loc$ is the initial location;
- $\mathbb{C}$ is a finite set of clocks;
- $v^\star$ is the initial clock valuation — a mapping $\mathbb{C} \to \mathbb{R}_{\geq 0}$;
- $Act = Act^I \cup Act^O \cup Act^\tau$ is a finite set of actions partitioned into input, output and internal actions, respectively;
- $E \subseteq Loc \times 2^{Act} \times \mathcal{B}(\mathbb{C}) \times 2^{\mathbb{C}} \times Loc$ is a finite set of edges — $\mathcal{B}(\mathbb{C})$ is a set of conditions over $\mathbb{C}$ (guards);
- $Inv\colon Loc \to \mathcal{B}(\mathbb{C})$ is a mapping that associates an invariant with every location.

and, for all $l \in Loc$, there is $(l, \emptyset, \mathrm{true}, \emptyset, l')$ where $Inv(l)$ implies $Inv(l')$.

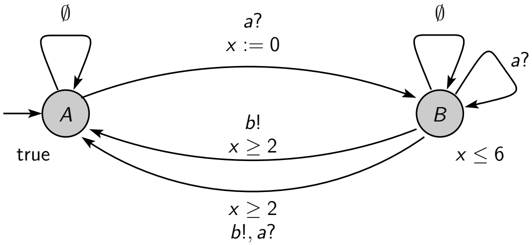
Execution

An *execution* starting in location l_0 and valuation v_0 is an infinite sequence $(l_0, v_0, d_0) \xrightarrow{S_0, R_0} (l_1, v_1, d_1) \xrightarrow{S_1, R_1} \ldots$ where, for all i:

- $l_i \in Loc$, v_i is a clock valuation and $d_i \in \mathbb{R}_{>0}$;
- $S_i \subseteq Act$ and $R_i \subseteq \mathbb{C}$;
- for all $0 \leq t \leq d_i$, $v_i + t \models Inv(l_i)$;
- $v_{i+1} = (v_i + d_i)^{\mathbf{R_i}}$;
- there is $(l_i, S_i, C_i, R_i, l_{i+1}) \in E$ such that $v_i + d_i \models C_i$.

Example

$\mathcal{A}^x$ waits for receiving *a*, after which it sends *b* (possibly receiving *a* at the same time) within six time units but not before two times units have passed (all *a*'s received in the meanwhile being ignored); then, $\mathcal{A}^x$ waits for receiving *a* again.

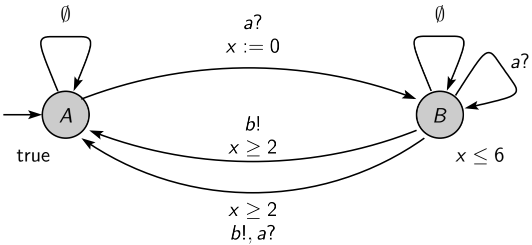


Example

An example of a partial execution of $\mathcal{A}^x$ is

$$(A, 0, 2) \xrightarrow{\{a\},\{x\}} (B, 0, 3) \xrightarrow{\{b\},\emptyset} (A, 3, 5) \xrightarrow{\{a\},\{x\}} (B, 0, 2)$$

which shows that $(B, 0)$ is reachable at time 2 (after the first transition) and at time 10 $(= 2 + 3 + 5)$ (after three transitions).



Composition

Given two compatible TIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$ (disjoint sets of clocks, inputs and outputs), $\mathcal{A}_1 \parallel \mathcal{A}_2 = \langle Loc, l^\star, \mathbb{C}, v^\star, E, Act, Inv \rangle$ where:

- $Loc = Loc_1 \times Loc_2$,
- $l^\star = (l_1^\star, l_2^\star)$,
- $\mathbb{C} = \mathbb{C}_1 \cup \mathbb{C}_2$,
- $v^\star = v_1^\star \cup v_2^\star$, i.e., the clock valuation for $\mathbb{C}$ induced by $v_1^\star$ and $v_2^\star$,
- $Act^I = (Act_1^I \setminus Act_2^O) \cup (Act_2^I \setminus Act_1^O)$,
  $Act^O = (Act_1^O \setminus Act_2^I) \cup (Act_2^O \setminus Act_1^I)$,
- $Inv((l_1, l_2)) = Inv_1(l_1) \wedge Inv_2(l_2)$;
- $((l_1, l_2), S, C, R, (l_1', l_2')) \in E$ iff $C = C_1 \wedge C_2$, $S_i = S \cap Act_i$,
  $R = R_1 \cup R_2$ for $(l_1, S_1, C_1, R_1, l_1') \in E_1$ and $(l_2, S_2, C_2, R_2, l_2') \in E_2$.

Timed machines

A timed machine is a TIOA that executes in the context of a clock granularity δ, i.e., its actions are always executed at multiples of δ.

Discrete timed I/O machine

A DTIOM is a pair $\mathcal{M} = \langle \delta_{\mathcal{M}}, \mathcal{A}_{\mathcal{M}} \rangle$ where $\delta_{\mathcal{M}} \in \mathbb{R}_{>0}$ and $\mathcal{A}_{\mathcal{M}} = \langle Loc, l^\star, \mathbb{C}, v^\star, E, Act, Inv \rangle$ is a TIOA such that $v^\star$ assigns a multiple of $\delta_{\mathcal{M}}$ to every clock in $\mathbb{C}$.

Execution

The *executions* $\mathcal{M}$ are those of $\mathcal{A}_{\mathcal{M}}$ restricted to transitions at every $\delta_{\mathcal{M}}$, i.e., $(l_0, v_0, d_0) \xrightarrow{S_0, R_0} (l_1, v_1, d_1) \xrightarrow{S_1, R_1} \ldots$ where the durations d_i are $\delta_{\mathcal{M}}$.

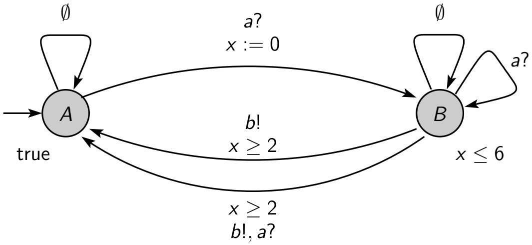
Therefore, we represent executions of DTIOMs as sequences $(l_0, v_0) \xrightarrow{S_0, R_0} (l_1, v_1) \xrightarrow{S_1, R_1} \ldots$

Example

A partial execution of $\mathcal{M}^x = \langle \delta_x, \mathcal{A}^x \rangle$ with $\delta_x = 2$ and $\mathcal{A}^x$ as below

$$(A, 0) \xrightarrow{\{a\},\{x\}} (B, 0) \xrightarrow{\emptyset,\emptyset} (B, 2) \xrightarrow{\{b\},\emptyset} (A, 4)$$

a is executed at time 2 — $\mathcal{M}^x$ remaining in the initial state for 2 time units, nothing is executed at time 4, and b is executed at time 6.



Composition of DTIOA can be extended to DTIOMs with the same clock granularity.

Composition

Given two TIOAs $\mathcal{A}_1$ and $\mathcal{A}_2$ that are compatible, we define the composition $\langle \delta, \mathcal{A}_1 \rangle \| \langle \delta, \mathcal{A}_2 \rangle = \langle \delta, \mathcal{A}_1 \| \mathcal{A}_2 \rangle$.

What if the clock granularities are different, i.e., what should $\langle \delta_1, \mathcal{A}_1 \rangle \| \langle \delta_2, \mathcal{A}_2 \rangle$ be?

What should parallel composition model?

Typically, we should have something like

$$[\![S_1 \parallel S_2]\!] = [\![S_1]\!] \cap [\![S_2]\!]$$

where $[\![S]\!]$ models the "behaviour" of S.

This is true of TIOA where $[\![\mathcal{A}]\!]$ is the set of executions of $\mathcal{A}$.

This is also true of DTIOM where $[\![\langle \delta, \mathcal{A} \rangle]\!]$ is the set of executions of $\langle \delta, \mathcal{A} \rangle$.

However, if δ_1 and δ_2 are different, the executions of $\langle \delta_1, \mathcal{A}_1 \rangle$ and of $\langle \delta_2, \mathcal{A}_2 \rangle$ will have different time lines and, therefore, it doesn't make much sense to calculate the intersections of the corresponding sets of executions.

Traces

Timed trace

Let A be a finite set (of actions).

- A *time sequence* τ is an infinite sequence of non-negative real numbers such that: $\tau(0) = 0$; $\tau(i) < \tau(i+1)$ for every $i \in \mathbb{N}$; the set $\{\tau(i) : i \in \mathbb{N}\}$ is unbounded, i.e., time progresses.
- An *action sequence* σ is an infinite sequence of elements of 2^{A} — i.e., of sets of actions — such that $\sigma(0) = \emptyset$.
- A *timed trace* is a pair $\langle \sigma, \tau \rangle$.

δ-timed trace

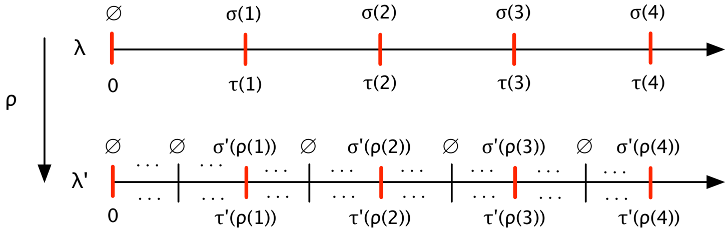
Let A be a finite set and $\delta \in \mathbb{R}_{>0}$.

- A *δ-timed trace* is a pair $\langle \sigma, \tau_\delta \rangle$ where $\tau_\delta(i) = i \cdot \delta$ for every $i \in \mathbb{N}$.
- A *δ-timed property* is a set of δ-timed traces.

Time refinement

Let $\rho : \mathbb{N} \to \mathbb{N}$ be monotonically increasing and satisfy $\rho(0) = 0$.

- Let τ, τ' be two time sequences. We say that $\tau' \preceq_\rho \tau$ iff $\tau(i) = \tau'(\rho(i))$ for every $i \in \mathbb{N}$.
- We say that $\langle \sigma', \tau' \rangle \preceq_\rho \langle \sigma, \tau \rangle$ iff $\tau' \preceq_\rho \tau$ and, for every $i \in \mathbb{N}$ and $\rho(i) < j < \rho(i+1)$, $\sigma(i) = \sigma'(\rho(i))$ and $\sigma'(j) = \emptyset$.
- We say that Λ is r-closed iff $\lambda' \in \Lambda$ whenever there exists $\lambda \in \Lambda$ such that $\lambda' \preceq_\rho \lambda$ for some ρ.



Behaviour of timed machines

Execution

The *executions* $\mathcal{M}$ are those of $\mathcal{A}_{\mathcal{M}}$ restricted to transitions at every $\delta_{\mathcal{M}}$, i.e., $(l_0, v_0, d_0) \xrightarrow{S_0,R_0} (l_1, v_1, d_1) \xrightarrow{S_1,R_1} \ldots$ where the durations d_i are $\delta_{\mathcal{M}}$.

Therefore, we represent executions of DTIOMs as sequences $(l_0, v_0) \xrightarrow{S_0,R_0} (l_1, v_1) \xrightarrow{S_1,R_1} \ldots$

Denotation

Every execution defines the $\delta_{\mathcal{M}}$-timed trace $\lambda = \langle \sigma, \tau_{\delta_{\mathcal{M}}} \rangle$ over *Act* where $\sigma(0) = \emptyset$ and, for all $i \geq 0$, $\sigma(i+1) = S_i$.

We denote by $[\![\mathcal{M}]\!]$ the r-closure of the set of such timed traces.

It now makes sense to calculate the intersection

$$[\![\langle \delta_1, \mathcal{A}_1 \rangle]\!] \cap [\![\langle \delta_2, \mathcal{A}_2 \rangle]\!]$$

which is the set of timed-traces in which both components can agree.

We can actually conclude that, if δ_1 and δ_2 are not commensurate (i.e., do not have a common divisor), that intersection is empty unless the two machines do not share any actions: they will never be able to synchronise on input/output pairs otherwise.

What if δ_1 and δ_2 are commensurate? Is there a machine $\mathcal{M}$ such that $[\![\mathcal{M}]\!] = [\![\langle \delta_1, \mathcal{A}_1 \rangle]\!] \cap [\![\langle \delta_2, \mathcal{A}_2 \rangle]\!]$?

Given $\mathcal{M} = \langle \delta, \mathcal{A} \rangle$, we define its *k-refinement* $\mathcal{M}_k = \langle \delta/k, \mathcal{A}_k \rangle$ by dividing every state of $\mathcal{A}$ in k copies such that the original transitions are performed in the last 'tick', all previous 'ticks' performing no actions.

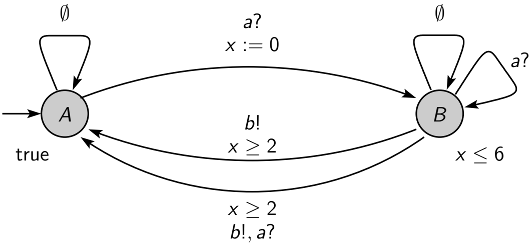
Refinement of timed machines

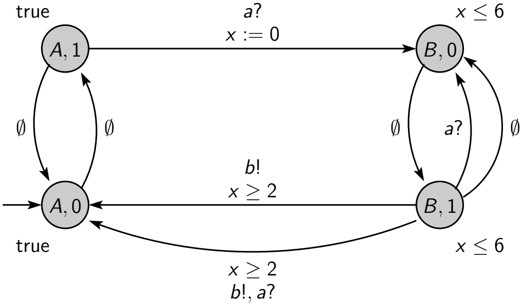
Given a TIOA $\mathcal{A} = \langle Loc, q_0, \mathbb{C}, E, Act, Inv \rangle$ and $k \in \mathbb{N}_{>0}$, its *k-refinement* is the TIOA $\mathcal{A}_k = \langle Loc_k, q_{k0}, \mathbb{C}, E_k, Act, Inv_k \rangle$ where:

- $Loc_k = Loc \times [0..k-1]$;
- $q_{k0} = (q_0, 0)$;
- $Inv_k(l, i) = Inv(l)$;
- for every (l, S, C, R, l') of E, E_k consists of the edge $((l, k-1), S, C, R, (l', 0))$ and all edges of the form $((l, i), \emptyset, true, \emptyset, (l, i+1))$, $i \in [0..k-2]$.

Given a timed machine $\mathcal{M} = \langle \delta, \mathcal{A} \rangle$, its *k-refinement* is $\mathcal{M}_k = \langle \delta/k, \mathcal{A}_k \rangle$.

Example


∅
a?
x := 0
∅
a?
A
B
b!
x ≥ 2
true
x ≤ 6
x ≥ 2
b!, a?


true
a?
x := 0
x ≤ 6
A, 1
B, 0
∅
∅
∅
a?
∅
b!
x ≥ 2
A, 0
B, 1
true
x ≤ 6
x ≥ 2
b!, a?

Compatibility of timed machines

Two DTIOMs $\mathcal{M}_i = \langle \delta_i, \mathcal{A}_i \rangle$, $i = 1, 2$, are said to be δ-*compatible* (where $\delta \in \mathbb{R}_{>0}$) if δ is a common divisor of δ_1 and δ_2.

Heterogeneous composition of timed machines

The δ-*composition* of two δ-*compatible* DTIOMs is

$$\mathcal{M}_1 \parallel_\delta \mathcal{M}_2 = \mathcal{M}_{1(\delta_1/\delta)} \parallel \mathcal{M}_{2(\delta_2/\delta)} = \langle \delta, \mathcal{A}_{1(\delta_1/\delta)} \parallel \mathcal{A}_{2(\delta_2/\delta)} \rangle$$

If δ is the greatest common divisor of δ_1 and δ_2, we use the notation $\mathcal{M}_1 \| \mathcal{M}_2$ and simply refer to *the composition* of $\mathcal{M}_1$ and $\mathcal{M}_2$.

Do we now have $[\![\mathcal{M}_1 \| \mathcal{M}_2]\!] = [\![\mathcal{M}_1]\!] \cap [\![\mathcal{M}_2]\!]$?

Approximation

Timed properties

- A timed property Λ' refines a timed property Λ — $\Lambda' \preceq \Lambda$ — if, for every $\lambda' \in \Lambda'$, there exists $\lambda \in \Lambda$ such that $\lambda' \preceq \lambda$.
- A timed property Λ' *approximates* a timed property Λ — $\Lambda' \precapprox \Lambda$ — if $\Lambda' \preceq \Lambda$ and, for every $\lambda \in \Lambda$, there exists $\lambda' \in \Lambda'$ such that $\lambda' \preceq \lambda$.

Timed machines

Given a timed machine $\mathcal{M} = \langle \delta, \mathcal{A} \rangle$ and $k \in \mathbb{N}_{>0}$:

- Every execution of $\mathcal{M}$ defines a unique execution of $\mathcal{M}_k$
- Because $[\![\mathcal{M}]\!]$ is closed, $[\![\mathcal{M}_k]\!] \subseteq [\![\mathcal{M}]\!]$ and, hence, $[\![\mathcal{M}_k]\!] \preceq [\![\mathcal{M}]\!]$.
- $[\![\mathcal{M}_k]\!] \precapprox [\![\mathcal{M}]\!]$, which we also write $\mathcal{M}_k \precapprox \mathcal{M}$.

Theorem

Let $\mathcal{M}_i = \langle \delta_i, \mathcal{A}_i \rangle$ be compatible and δ be the greatest common divisor of δ_1 and δ_2.

- $[\![\mathcal{M}_1 \parallel_\delta \mathcal{M}_2]\!] \precapprox [\![\mathcal{M}_1]\!] \cap [\![\mathcal{M}_2]\!]$
- If $[\![\mathcal{M}]\!] \precapprox [\![\mathcal{M}_1]\!] \cap [\![\mathcal{M}_2]\!]$ then $\mathcal{M} \precapprox \mathcal{M}_1 \parallel_\delta \mathcal{M}_2$

The machine $\mathcal{M}_1 \parallel_\delta \mathcal{M}_2$ approximates and is the best approximation of the joint behaviour of $\mathcal{M}_1$ and $\mathcal{M}_2$, i.e., of $[\![\mathcal{M}_1]\!] \cap [\![\mathcal{M}_2]\!]$.

This is important so that properties of the joint behaviour of the two timed machines can be inferred from that of the composite machine or that their joint behaviour can be simulated through a machine.

Dynamic homogoneous composition

Given two DTIOMs $\mathcal{M}_i = \langle \delta, \mathcal{A}_i \rangle$ and triples (l_i, v_i, t_i) such that (l_i, v_i) is a state of $\mathcal{M}_i$ reachable at time t_i, we define

$$\mathop{\Big|\Big|}_{i=1,2}^{(l_i,v_i,t_i)} \mathcal{M}_i$$

as $\langle \delta, \mathcal{A} \rangle$ where $\mathcal{A}$ is obtained by replacing the initial location and clock valuation of $\mathcal{A}_1 \parallel \mathcal{A}_2$ with (l_1, l_2) and $v_1 \cup v_2$, respectively.

Dynamic heterogeneous composition

The δ-*composition* of two δ-*compatible* DTIOMs $\mathcal{M}_1$ and $\mathcal{M}_2$ at (l_i, v_i, t_i) such that each (l_i, v_i) is a state of $\mathcal{M}_i$ reachable at time t_i is:

$${}_{\delta}\Big|\Big|_{i=1,2}^{(l_i,v_i,t_i)} \mathcal{M}_i \triangleq \mathop{\Big|\Big|}_{i=1,2}^{((l_i,0),v_i,t_i)} \mathcal{M}_{i(\delta_i/\delta)}$$

.

Consistency

Consistency

A DTIOM $\mathcal{M}$ is said to be *consistent* if $[\![\mathcal{M}]\!] \neq \emptyset$.

How can we check that a timed machine is consistent without calculating its semantic?

Refinement

Let $k \in \mathbb{N}_{>0}$. A DTIOM $\mathcal{M}$ is consistent iff its k-refinement $\mathcal{M}_k$ is consistent. More generally, for arbitrary DTIOM $\mathcal{M}$ and $\mathcal{M}'$,

- if $\mathcal{M}' \preceq \mathcal{M}$ and $\mathcal{M}'$ is consistent, then so is $\mathcal{M}$, and
- if $\mathcal{M}' \precapprox \mathcal{M}$, then $\mathcal{M}'$ is consistent iff $\mathcal{M}$ is consistent.

Initializable

A DTIOM $\mathcal{M}$ is said to *be initializable* if, for every $0 \leq t \leq \delta_{\mathcal{M}}$, $(l^\star, v^\star + t) \vDash Inv(l^\star)$.

That is, a DTIOM is initializable if it can stay in the initial state until the first tick of the clock.

Independent progress

A DTIOM $\mathcal{M}$ is said to *make independent progress* if, for every reachable state (l, v), there is an edge (l, A, C, R, l') such that:

- $A \subseteq Act^O_{\mathcal{M}} \cup Act^\tau_{\mathcal{M}}$
- $v + \delta_{\mathcal{M}} \vDash C$
- for all $0 \leq t \leq \delta_{\mathcal{M}}$, $(v + \delta_{\mathcal{M}})^{\mathbf{R}} + t \vDash Inv(l')$

That is, if the DTIOM is able to make a transition from any reachable state without forcing the environment to provide any input.

Theorem

Any initializable DTIOM that makes independent progress is consistent.

Can this be checked in a compositional way?

The fact that two DTIOMs $\mathcal{M}_1$ and $\mathcal{M}_2$ are such that δ_1 and δ_2 are commensurate simply means that we can find a clock granularity in which we can accommodate the transitions that the two DTIOMs perform: by itself, this does not ensure that the two DTIOMs can jointly execute their input/output synchronisation pairs.

> *For example, if $\delta_1 = 2$ and $\delta_2 = 3$ and $\mathcal{M}_2$ only performs non-empty actions at odd multiples of 3, the two machines will not be able to agree on their input/output synchronisation pairs.*

For the DTIOMs to actually be able to interact with each other it is necessary that their input/output synchronisation pairs can be performed on a common multiple of δ_1 and δ_2.

Cooperative

A DTIOM $\mathcal{M}$ is said to be *cooperative in relation to* $Q \subseteq Act_{\mathcal{M}}$ and a multiple δ of $\delta_{\mathcal{M}}$ if the following holds for every (l, v) reachable at a time T such that $(T + \delta_{\mathcal{M}})$ is not a multiple of δ:

- for every edge $(l, A, C, R, l') \in E_{\mathcal{M}}$ such that $v + \delta_{\mathcal{M}} \models C$ and $(v + \delta_{\mathcal{M}})^{\mathbf{R}} + t \models Inv_{\mathcal{M}}(l')$ for all $0 \leq t \leq \delta_{\mathcal{M}}$ — i.e., the machine makes a transition at a time that is not a multiple of δ
- there exists an edge $(l, A \backslash Q, C', R', l'')$ such that $v + \delta_{\mathcal{M}} \models C'$ and, for all $0 \leq t \leq \delta_{\mathcal{M}}$, $(v + \delta_{\mathcal{M}})^{\mathbf{R}'} + t \models Inv_{\mathcal{M}}(l'')$ — i.e., the machine can make an alternative transition that does not perform any actions in Q.

Essentially, being cooperative in relation to Q and δ means that the machine will not force transitions that perform actions in Q at times that are not multiples of δ.

DP-enabled

A DTIOM $\mathcal{M}$ is said to be *DP-enabled in relation to* $J \subseteq Act^I_{\mathcal{M}}$ and δ multiple of $\delta_{\mathcal{M}}$ if the following property holds for every $B \subseteq J$ and state (l, v) reachable at a time T such that $(T + \delta_{\mathcal{M}})$ is a multiple of δ:

> for every edge $(l, A, C, R, l') \in E_{\mathcal{M}}$ such that $v + \delta_{\mathcal{M}} \vDash C$ and, for all $0 \leq t \leq \delta_{\mathcal{M}}$, $(v + \delta_{\mathcal{M}})^{\mathbf{R}} + t \vDash Inv_{\mathcal{M}}(l')$ — i.e., the machine can make a transition
>
> there exists an edge $(l, B \cup (A \backslash J), C', R', l'')$ such that $v + \delta_{\mathcal{M}} \vDash C'$ and, for all $0 \leq t \leq \delta_{\mathcal{M}}$, $(v + \delta_{\mathcal{M}})^{\mathbf{R}'} + t \vDash Inv_{\mathcal{M}}(l'')$ — i.e., the machine can make an alternative transition that accepts instead B as inputs and still performs the same outputs (and inputs outside J).

That is, a DTIOM is DP-enabled in relation to a set of inputs J and a multiple δ of its clock granularity if, whenever it leaves a reachable state at a multiple of δ, it can do so by accepting any subset of J, and if its outputs are independent of the inputs in J that it receives.

Theorem

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be δ-compatible DTIOMs that can make independent progress. If, for some δ' multiple of δ_1 and δ_2,

- $\mathcal{M}_1$ is DP-enabled in relation to $Act_1^I \cap Act_2^O$ and δ',
- $\mathcal{M}_2$ is DP-enabled in relation to $Act_2^I \cap Act_1^O$ and δ',
- both $\mathcal{M}_1$ and $\mathcal{M}_2$ are δ'-cooperative in relation to $Act_1 \cap Act_2$,

then $\mathcal{M}_1 \parallel_\delta \mathcal{M}_2$ is initializable and makes independent progress (and, hence, is consistent).

To conclude

- Timed machines do not really provide a component algebra for dynamic and heterogeneous timed systems.
- We developed a component algebra based on networks of timed machines, i.e., the components are networks, not individual machines.
- The composition operator is dynamic, i.e., it takes into account the time and the behaviour until that time of the networks be composed.
- In this network algebra we investigated important properties for run-time interconnection, including consistency and feasibility.
- We also investigated how consistency and feasibility can be proved compositionally, and at design time.

References

J. L. Fiadeiro, A. Lopes, B. Delahaye, and A. Legay (2016)
Dynamic Networks of Heterogeneous Timed Machines
Mathematical Structures in Computer Science, in print

J. L. Fiadeiro and A. Lopes (2016)
Heterogeneous and Asynchronous Networks of Timed Systems
Theoretical Computer Science, in print

B. Delahaye, J. L. Fiadeiro, A. Legay, and A. Lopes (2014)
Heterogeneous timed machines
LNCS 8687, 115-132, 2014.

B. Delahaye, J. L. Fiadeiro, A. Legay, and A. Lopes (2013)
A timed component algebra for services
LNCS 7892, 242-257, 2013.

J. L. Fiadeiro and A. Lopes (2013)
An interface theory for service-oriented design
Theoretical Computer Science, 503:1-30, 2013.