Constraint-based Modelling of Discrete Event Dynamic Systems

Gérard Verfaillie, Cédric Pralet, and Michel Lemaître
ONERA, Toulouse, France

Presentation outline

1. **Motivations**;
2. **Modeling principles**;
3. **Timeline-based representation**;
4. **Constraints** on **timelines**;
5. **Subsumed frameworks**;
6. Remaining **work** and possible **extensions**.

Motivations (1/3)

Scientific communities

- with **various motivations**: programming, simulation, validation, diagnosis, situation tracking, planning, scheduling, decision-making ...
- using **various formalisms**: automata, Petri nets, synchronous languages, temporal logic, situation calculus, STRIPS, Markov chains, Markov decision processes ...
- to reason on the **same objects**: **discrete event dynamic systems**.

$\longrightarrow$ **Would a common modeling framework be possible?**

Motivations (2/3)

Autonomous systems with needs in terms of situation tracking, decision-making, validation, . . . all of them working preferably on a **common model** of the **system dynamics** expressed in a **common framework**.

$\longrightarrow$ **A common modeling framework seems to be necessary.**

Motivations (3/3)

Constraints:

- very **flexible modeling framework**: propositional logic extended to non boolean variables, either numeric or symbolic, either discrete or continuous, and to any kind of constraint;
- numerous **generic algorithms**: local and global constraint propagation, tree search, greedy search, local search, dynamic programming, . . .

→ **A constraint-based framework for discrete event dynamic systems?**

Able for example to model the same way **planning** and **scheduling problems** and to manage them really together.

Modeling principles

1. **Continuous** time: $\mathbb{R}$;

2. **State** and **Event variables**;

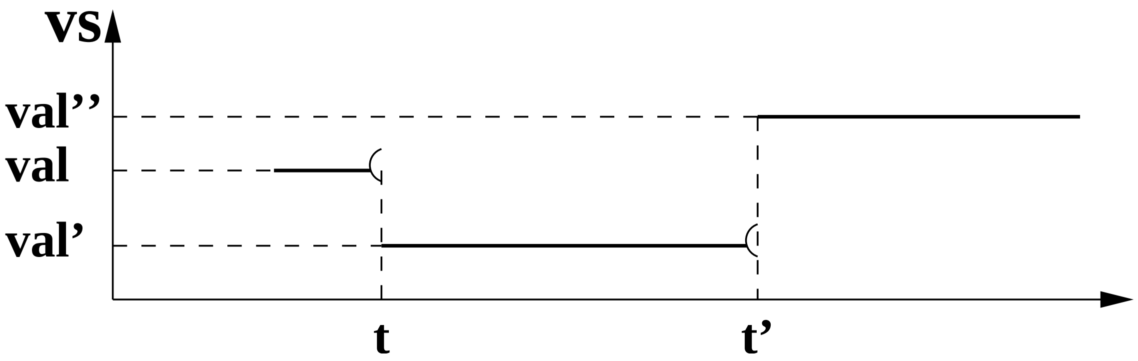
3. **Instants** of state change or event $=$ **discrete** subset of $\mathbb{R}$.

State variables

To model the **permanent features** of the system, which evolve with time.

Examples: orbital position, energy level, observation instrument mode, movement in progress, ...

Assumption of a **stepwise** evolution. **Continuous changes** are not modeled.



By convention, if t and t' are two **successive** instants of change for a state variable vs, the value of vs over the **semi-open** interval $[t, t'[$ is the value it took at time t (val').

Event variables

To model **instantaneous phenomena**.

Examples: arrival of an observation request, data downloading decision, . . .

Default value $\perp$ to model the **absence of event**.

By convention, if t and t' are two **successive** instants of event
for an event variable ve, the value of ve over the **open** interval $]t, t'[$ is $\perp$.

Assumptions

Neither **correlation**, nor **causality** assumptions between state changes and events.

Neither **Markovian**, nor **stationary** assumptions.

$\longrightarrow$ To offer a very generic framework, with a great **freedom** in terms of modeling.

A timeline-based representation

Timeline $tl = \langle v, d, I, t_I, v_I \rangle$ with:

1. v a **state** or **event variable**;

2. d its **domain of values**;

3. I a **countable sequence of instants**, finite or not;
   let be $I = [1, \ldots i, \ldots]$, $I^+ = [0, 1, \ldots i, \ldots]$, and $I^- = [2, \ldots i, \ldots]$;

4. t_I a **sequence of temporal variables**, with values in $\mathbb{R}$;
   $\forall i \in I$, $t_i =$ temporal position of instant i;
   totally ordered instants: $\forall i \in I^-$, $t_{i-1} \leq t_i$;

5. v_I a **sequence of atemporal variables**, with values in d;
   $\forall i \in I^+$, $v_i =$ value of v at instant i;
   $\forall i \in I^-$, $(t_i = t_{i-1}) \rightarrow (v_i = v_{i-1})$;
   $v_0 =$ initial value of v, equal to $\perp$ for an event variable.

Tabular representation of a timeline

	0	1	...	i	...
t		t_1	...	t_i	...
v	v_0	v_1	...	v_i	...

In a timeline, the t_i's and v_i's are either **constant** or **variable**.

$\longrightarrow$ Timeline $=$ **array** of variables and constants, either temporal or atemporal.

Warning !

Do not mistake:

1. **state** and **event variables**

2. for **temporal** and **atemporal variables**.

Only the latter are really **mathematical variables**.

The former are **functions** of time.

We keep the term **variable** for the latter.

We use the term **timeline** for the former.

Constraint network on timelines

Constraint network on timelines $CNT = \langle TL, C \rangle$ with:

1. TL **finite set of timelines**
   sharing the same sequences I and t_I (in the current version);

2. C **finite set of constraints** on TL.

Let Var be the set of variables (temporal and atemporal) in all the timelines in TL.

Constraint on timelines

Constraint on timelines $c = \langle qt, cd, sc, df \rangle$ with:

1. qt **finite sequence of quantifiers** $[q_1, \ldots q_j, \ldots q_m]$, with $q_j \in \{\forall, \exists\}$;

2. cd **finite sequence of conditions** $[c_1, \ldots c_j, \ldots c_m]$, with c_j boolean function over I^j;

3. sc **function** associating with each sequence $[i_1, \ldots i_j, \ldots i_m] \in I^m$ the **scope** of a basic constraint $sc(i_1, \ldots i_m)$, that is a **finite sequence of variables** in Var;

4. df **function** associating with each sequence $[i_1, \ldots i_j, \ldots i_m] \in I^m$ the **definition** of a basic constraint $df(i_1, \ldots i_m)$, that is a **boolean function** over the Cartesian product of the domains of the variables in $sc(i_1, \ldots i_m)$.

If $m = 0$, sc is the **scope** of a basic constraint and df the associated **definition**.

Basic constraint $=$ CSP constraint.

Example of constraint

Let us assume a **state variable** v whose value changes at each instant from the third one, if this instant occurs between times 5 and 10.

Constraint: $\forall i \in I\,,\ ((i \geq 3) \wedge (5 \leq t_i \leq 10)) \rightarrow (v_i \neq v_{i-1})$.

$c = \langle qt, cd, sc, df \rangle$ with:

1. $qt = [\forall]$;

2. $cd = [(i \geq 3)]$;

3. $\forall i \in I,\ sc(i) = [t_i, v_i, v_{i-1}]$;

4. $\forall i \in I,\ df(i) \equiv ((5 \leq t_i \leq 10) \rightarrow (v_i \neq v_{i-1}))$.

Constraint satisfaction

Let be a **finite completely assigned** CNT (assignment A of Var).

Satisfaction of a constraint $c = \langle qt, cd, sc, df \rangle$ by $A =$
satisfaction of $\langle \emptyset, qt, cd, sc, df \rangle$ by A.

Satisfaction of $\langle Is, qt, cd, sc, df \rangle$ by $A =$

- if $qt = cd = \emptyset$: $(df(Is))(A_{\downarrow sc(Is)}) = true$
  usual definition of constraint satisfaction;

- if $qt = [q] \cup qt'$ and $cd = [c] \cup cd'$:
  - if $q = \forall$: $\forall i \in I$ such that $c(Is \cup [i])$, A satisfies $\langle Is \cup [i], qt', cd', sc, df \rangle$
    $\longrightarrow$ **conjonction** of constraints;
  - if $q = \exists$: $\exists i \in I$ such that $c(Is \cup [i])$ and A satisfies $\langle Is \cup [i], qt', cd', sc, df \rangle$
    $\longrightarrow$ **disjonction** of constraints.

Complexity of constraint checking

Let be:

1. md the **maximum domain size**;

2. ma the **maximum constraint arity** among the induced basic constraints;

3. ml the **maximum quantifier sequence length** among the non basic constraints;

4. l the **maximum timeline length**.

Temporal complexity : $O(l^{ml} \cdot c(ma, md))$.

Specific constraints

1. **pure temporal** constraints;
2. **instantaneous state** constraints;
3. **instantaneous event** constraints;
4. **instantaneous transition** constraints;
5. **non instantaneous transition** constraints.

Example of an instantaneous state constraint

A **robot** with **two instruments** which cannot active at the same time.

Two **boolean state timelines** $is1$ and $is2$.

A **universally quantified instantaneous state constraint** $\langle qt, cd, sc, df \rangle$:

1. $qt = [\forall]$;

2. $cd = [true]$;

3. $\forall i \in I$, $sc(i) = [is1_i, is2_i]$;

4. $\forall i \in I$, $df(i) \equiv \neg(is1_i \wedge is2_i)$.

$$\forall i \in I \, , \; \neg(is1_i \wedge is2_i).$$

Example of an instantaneous state constraint

A **robot** which must be at a given **location** lo_G by **time** t_G.

A **state timeline** lo (current location).

An **existentially quantified instantaneous state constraint** $\langle qt, cd, sc, df \rangle$:

1. $qt = [\exists]$;

2. $cd = [true]$;

3. $\forall i \in I$, $sc(i) = [lo_i, t_i]$;

4. $\forall i \in I$, $df(i) \equiv ((lo_i = lo_G) \wedge (t_i \leq t_G))$.

$$\exists i \in I \, , \, ((lo_i = lo_G) \wedge (t_i \leq t_G)).$$

Example of an instantaneous event constraint

A **robot** with a set A of **possible actions**, such two actions cannot be triggered at the same time and each action $a \in A$ requires a level $e(a)$ of energy.

An **event timeline** ca (current action) of domain $A \cup \{\perp\}$.

A **state timeline** ce (current level of energy).

A **universally quantified instantaneous event constraint** $\langle qt, cd, sc, df \rangle$:

1. $qt = [\forall]$;

2. $cd = [true]$;

3. $\forall i \in I, \; sc(i) = [ca_i, ce_{i-1}]$;

4. $\forall i \in I, \; df(i) \equiv ((ca_i \neq \perp) \rightarrow (ce_{i-1} \geq e(ca_i)))$.

$$\forall i \in I \, , \; ((ca_i \neq \perp) \rightarrow (ce_{i-1} \geq e(ca_i))).$$

Example of an instantaneous transition constraint

A **impulse switch**.

A **boolean state timeline** sp (current switch position).

A **boolean state timeline** st (current switch state, stuck or not).

A **boolean event timeline** im (current impulse).

A **universally quantified instantaneous transition constraint** $\langle qt, cd, sc, df \rangle$:

1. $qt = [\forall]$;
2. $cd = [true]$;
3. $\forall i \in I,\ sc(i) = [sp_i, sp_{i-1}, st_i, im_i]$;
4. $\forall i \in I,\ df(i) \equiv ((sp_i \neq sp_{i-1}) \leftrightarrow (\neg st_i \wedge im_i))$.

$$\forall i \in I\ ,\ ((sp_i \neq sp_{i-1}) \leftrightarrow (\neg st_i \wedge im_i)).$$

Example of a non instantaneous transition constraint

A **robot** with a set A of **possible actions**, such two actions cannot be active at the same time and the duration of each action $a \in A$ belongs to an interval $[dmin(a), dmax(a)]$.

A **state timeline** ca (current action) of domain $A \cup \{\perp\}$.

A **universally quantified non instantaneous transition constraint**:

1. $qt = [\forall, \exists]$;
2. $cd = [true, i_1 < i_2]$;
3. $\forall [i_1, i_2] \in I^2$, $sc(i_1, i_2) = [ca_i \, , \, (i_1 - 1) \leq i \leq i_2] \cup [t_{i_1}, t_{i_2}]$;
4. $\forall [i_1, i_2] \in I^2$, $df(i_1, i_2) \equiv (((ca_{i_1-1} \neq a) \wedge (ca_{i_1} = a)) \rightarrow ((\wedge_{i_1<i<i_2}(ca_i = a)) \wedge (ca_{i_2} \neq a) \wedge (dmin(a) \leq (t_{i_2} - t_{i_1}) \leq dmax(a)))$.

$$\forall i_1 \in I \, , \, (((ca_{i_1-1} \neq a) \wedge (ca_{i_1} = a)) \rightarrow (\exists i_2 \in I \, , \, ((i_1 < i_2) \wedge (\wedge_{i_1<i<i_2}(ca_i = a)) \wedge (ca_{i_2} \neq a) \wedge (dmin(a) \leq (t_{i_2} - t_{i_1}) \leq dmax(a))))).$$

Some subsumed frameworks

1. **Automata**;
2. **Petri nets**;
3. **STRIPS planning**;
4. **Job-shop scheduling**.

Some possible requests

At least with **finite timelines** (finite sequence of instants).

1. **Situation tracking**: to find all the solutions of the CSP
   model + observations;

2. **Validation**: to prove the inconsistency of the CSP
   model + negation of the properties;

3. **Decision**: to distinguish between **controllable** and **uncontrollable** variables:
   (a) **optimistic** version: to find all the solutions of the CSP
   model + objectives;
   (b) **pessimistic** version: to find all the solutions of the QCSP
   model + objectives with controllable variables **existentially** quantified and uncontrollable variables **universally** quantified.

What remains to be done

1. to check the ability of the proposed framework to **subsume** other existing frameworks, such as **temporal automata**, **temporal logic**, or **situation calculus**;

2. to assess its ability to **model** compactly **dynamic systems**, **requests**, and **problems**;

3. to define specific efficient **propagation** and **search algorithms**, mainly at the interaction between **temporal** and **atemporal** variables;

4. to study **requests**, **problems**, and **algorithms** in the case of **infinite** timelines;

5. to define a more **user-friendly constraint expression language**.

Some possible extensions

1. to relax the assumption of **totally ordered instants** (done);

2. to go from **hard** constraints to **soft** ones (local functions not necessarily boolean) to model **plausibility** and **utility** degrees (see the **PFU** framework);

3. for state variables, to go from **stepwise** functions of time to other functions, such as for example **piecewise linear** ones.

PFU = **Plausibilities, Feasibilities, and Utilities**, Pralet-Verfaillie-Schiex, JAIR 2007.

To remember

CSP = **variables** + **domains** + **constraints**

Constraint network on timelines =

state and **event variables** + **domains** + **time** (timelines) + **constraints** =

atemporal and **temporal variables** + **domains** + **constraints**

For more details

Constraint-Based Modelling of Discrete Event Dynamic Systems

Gérard Verfaillie, Cédric Pralet, and Michel Lemaître

CP/ICAPS 2007 Workshop on **Constraint Satisfaction Techniques for Planning and Scheduling Problems**