TEMPORAL OBJECT RELATIONS

The two common time entities of time points and time intervals can be used to permit formal reasoning about time and temporal relationships.

Definition: *time point* - a real number that represents the occurrence time of an instantaneous event and is an indivisible entity. Defined with respect to a reference that should be the same for all time points.

Definition: let t_α and t_β be two time points so that binary relations can be specified between them:

- $t_\alpha < t_\beta$ (i.e. t_α before t_β) and its inverse $t_\alpha > t_\beta$
- $t_\alpha = t_\beta$

Definition: a *convex time interval* is a contiguous period that specifies a range of time points such that:

$$< t_\alpha, t_\beta > \;\equiv\; \{ t\text{:}\;\; t_\alpha \leq t \leq t_\beta \}$$

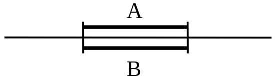
Definition: the *duration* of a convex time interval $< t_\alpha, t_\beta >$ is the time:

$$< t_\alpha, t_\beta > \;\equiv\; | t_\beta - t_\alpha |$$

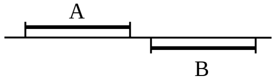
Definition: a *non-convex time interval* is a set of sub-intervals expressed as a union of disjoint convex intervals.

On the basis of these definitions it is possible to define a number of *convex interval relations* using the intervals, A = $< t^A_\alpha,\ t^A_\beta >$, B = $< t^B_\alpha,\ t^B_\beta >$:

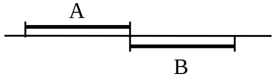
- *equal* (A = B) $\Rightarrow$ $(t^A_\alpha = t^B_\alpha) \wedge (t^A_\beta = t^B_\beta)$



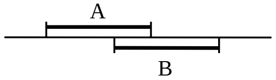
- *precede* (A ≺ B) $\Rightarrow$ $t^A_\beta < t^B_\alpha$ and inverse *succeed* (B ≻ A)



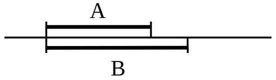
- *meet* (A ⇑ B) $\Rightarrow$ $t^A_\beta = t^B_\alpha$ and inverse *met-by* (B ⇓ A)



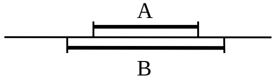
- *overlap* (A ∅ B) $\Rightarrow$ $t^A_\alpha < t^B_\alpha < t^A_\beta < t^B_\beta$ and inverse *overlapped-by* (B $\varnothing^u$ A)



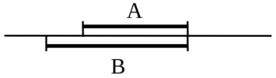
- *start* (A ↑ B) $\Rightarrow$ $t^B_\alpha = t^A_\alpha < t^A_\beta < t^B_\beta$ and inverse *started-by* (B $\uparrow^u$ A)



- *during* (A << B) $\Rightarrow$ $t^B_\alpha < t^A_\alpha < t^A_\beta < t^B_\beta$ and inverse *contains* (B >> A)



- *end* (A ↓ B) $\Rightarrow$ $t^B_\alpha < t^A_\alpha < t^A_\beta = t^B_\beta$ and inverse *ended-by* (B $\downarrow^u$ A)



These relations can be combined to express other relations, for example, the *disjoint* relation (><), i.e:

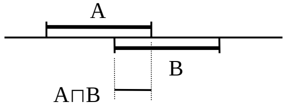
A >< B = (A ≺ B) ∨ (A ≻ B)

and all the containment possibilities of interval A in interval B:

A ↑ B ∨ (A << B) ∨ (A ↓ B)

Definition: the *interval intersection* of convex time intervals A and B is defined for ¬(A ≺ B) ∧ ¬ (A ≻ B) as:

$$A \sqcap B \equiv < \max (t^A_\alpha,\ t^B_\alpha), \min (t^A_\beta,\ t^B_\beta) >$$

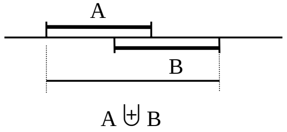


and for (A ≺ B) ∨ (A ≻ B) it is ∅

The *intersection* of two convex intervals that meet each other is a non-null convex interval of zero duration, i.e. a time point.

Definition: the *cover* of convex time intervals A and B is a convex interval defined as:

$$A \uplus B \equiv < \min (t^A_\alpha,\ t^B_\alpha), \max (t^A_\beta,\ t^B_\beta) >$$

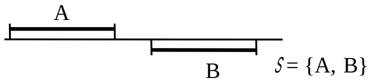


Note that the cover is a symmetric and commutative operation, so that it can be applied to a set of more than two intervals:

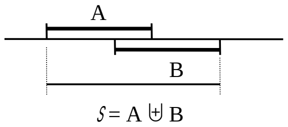
$$\biguplus_{i=1}^{n} \left\{C_i\right\} = C_1 \uplus C_2 \uplus \Lambda \uplus C_n$$

Definition: the set of *maximal convex sub-intervals* of convex time intervals A and B is defined as $\mathcal{S}$ ({A}, {B}), such that:

- A ⊓ B = ∅ ⇒ $\mathcal{S}$ = {A, B}

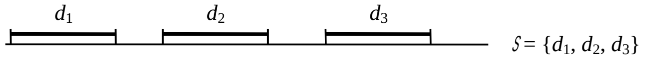


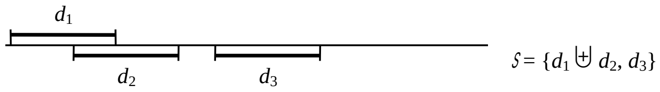
- A ⊓ B ≠ ∅ ⇒ $\mathcal{S}$ = {A ⊎ B}



The use of relations on event times and intervals is somewhat artificial due to the varying granularity of time knowledge (in practice), and the fact that intervals are not necessarily continuous but contain 'gaps'. To address this issue, non-convex intervals are required.

The set of maximal convex sub-intervals of a non-convex time interval $\mathcal{D}$ is the set of maximal convex sub-intervals of all its convex members $\{d_i\}$:





Definition: the *set union* of non-convex time intervals C and D is a non-convex interval made up of the set of members in $\mathcal{S}$({C}) and the set of members in $\mathcal{S}$({D}):

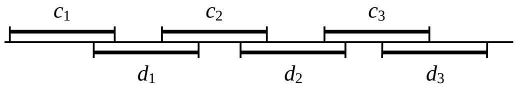
$$\{C\} \cup \{D\} \equiv \mathcal{S}(\{C\}) \cup \mathcal{S}(\{D\})$$

Also note that that the *cover* of the set union is:

$$\uplus\ (\{C\} \bigcup \{D\}) = < \min\,(t_\alpha^C,\ t_\alpha^D),\ \max\,(t_\beta^C,\ t_\beta^D) >$$

The set union is a symmetric and commutative operation, a property that allows the operation to be defined on a set of more than two intervals:

$$\bigcup_{i=1}^{n} \left\{C_i\right\} = C_1 \cup C_2 \cup \Lambda \cup C_n$$



Definition: the *interval union* of non-convex time intervals C and D is a non-convex interval denoted by:

$$\{C\} \sqcup \{D\} \equiv \mathcal{S}(\{C\} \cup \{D\})$$

We can note that the interval union cover is equal to the set union cover, and that the interval union is commutative:

$$\bigsqcup_{i=1}^{n} \left\{C_i\right\} = C_1 \sqcup C_2 \sqcup \cdots \sqcup C_n$$

FORMAL CONSTRAINT SPECIFICATION

Constraint specification can be expressed formally using the temporal object relations defined earlier. Let the incoming constraint imposed on the invoker by a remote object be TC_{in} and let the constraint imposed by the invoker on the remote object (invokee) be TC_{out}.

Let P_{in} be the invoker's computation time interval associated with the constraint TC_{in} and all constraints are referenced to global time.

To produce a specification for TC_{out} as a function of TC_{in} the following are needed:

- $\mathfrak{R}_{\mathrm{in}}$ and $\mathfrak{R}_{\mathrm{out}}$ are two convex interval relations
- a convex time interval TC_{R} with constant duration $\|TC_{\mathrm{R}}\|$
- the composite relation $TC_{\mathrm{in}}\ \mathfrak{R}_{\mathrm{in}}\ TC_{\mathrm{R}}\ \mathfrak{R}_{\mathrm{out}}\ TC_{\mathrm{out}}$

Examples: Let the constraint $TC_{\text{in}} = <t^{in}_{\alpha},\ t^{in}_{\beta}>$ and let the service to be invoked be required to succeed TC_{in} by at least γ time units. Thus in this case we have:

- $\Re_{\text{in}} \equiv \prec$ and $\Re_{\text{out}} \equiv \prec$
- $\|TC_{\text{R}}\| = \gamma$
- $TC_{\text{in}} \prec TC_{\text{R}} \prec TC_{\text{out}}$

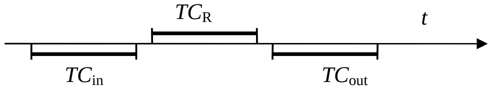
From these relations the following inferences can be drawn:

- $TC_{\text{in}} \prec TC_{\text{R}} \;\Rightarrow\; t^{in}_{\beta} < t^{R}_{\alpha}$
- $\|TC_{\text{R}}\| = \gamma \;\Rightarrow\; t^{R}_{\beta} = t^{R}_{\alpha} + \gamma$

$$> \quad t^{R}_{\beta} > t^{in}_{\beta} + \gamma$$

- $TC_{\text{R}} \prec TC_{\text{out}} \;\Rightarrow\; t^{R}_{\beta} < t^{out}_{\alpha} \;\Rightarrow\; t^{out}_{\alpha} > t^{in}_{\beta} + \gamma$

  i.e. the service to be invoked succeeds TC_{in} by more than γ time units.



Example: Let the service to be invoked complete in less than γ time units after TC_{in} has completed. Thus in this case we have:

- $\Re_{\text{in}} \equiv \varnothing$ and $\Re_{\text{out}} \equiv \succ \vee \Downarrow \vee \varnothing^{\text{u}} \vee \uparrow^{\text{u}} \vee \gg \vee \downarrow^{\text{u}}$
- $\|TC_{\text{R}}\| = \gamma$
- $TC_{\text{in}}\ \Re_{\text{in}}\ TC_{\text{R}}\ \Re_{\text{out}}\ TC_{\text{out}}$

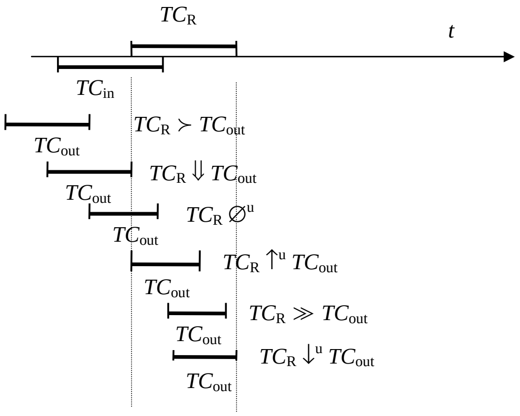
From these relations the following inferences can be drawn:

- $TC_{\text{in}} \varnothing TC_{\text{R}} \;\Rightarrow\; t^{R}_{\alpha} < t^{in}_{\beta}$
- $\|TC_{\text{R}}\| = \gamma \;\Rightarrow\; t^{R}_{\beta} = t^{R}_{\alpha} + \gamma$

$$> \quad t^{R}_{\beta} < t^{in}_{\beta} + \gamma$$

- $TC_{\text{R}}\ \Re_{\text{out}}\ TC_{\text{out}} \;\Rightarrow\; t^{out}_{\beta} \le t^{R}_{\beta} \;\Rightarrow\; t^{out}_{\beta} < t^{in}_{\beta} + \gamma$

  i.e. the service to be invoked completes in less than γ time units after TC_{in} completes.



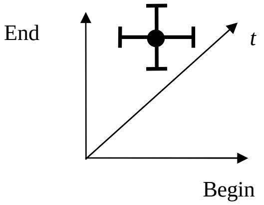
The temporal object relations are useful for specifying and analysing temporal properties. They can also be combined with resource allocation and schedulability considerations to support time-based object resource management.

TIME CONSTRAINT PROJECTION AND PROPAGATION

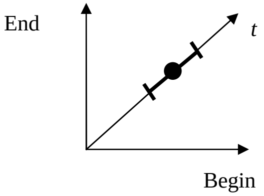
Objects must be able to view the temporal properties of other executing objects since they may require services from another object as part of a complete task.   The same requirement is posed for fault-tolerance redundancy where remote invocations of server objects may be required. This demands that time constraints are also satisfied in a distributed execution environment → time constraint projection is required for remote objects.

Constraint Projection

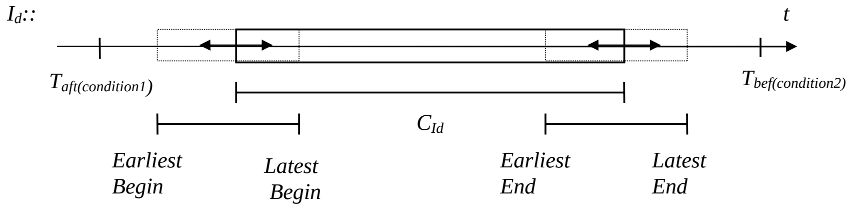
Each time constraint represents a *set of possible occurrences* in which the task beginning, task end and duration are constrained. This information can be most easily interpreted on a 2-D occurrence timing diagram, i.e:



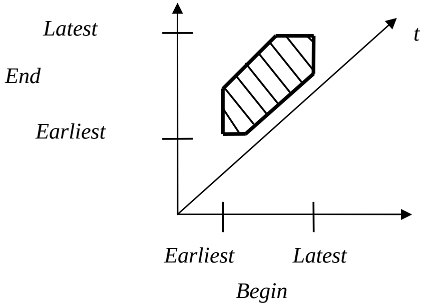
The diagonal represents the axis of time points since Begin(time) = End (time). An additional variance is introduced by time-knowledge uncertainty due to local clock variation → any time point creates an interval in the begin-end time plane, i.e:



An occurrence interval for object *Id* can be defined between the *Earliest Begin Time* (after $T_{aft(condition1)}$) and the *Lastest End Time* (before $T_{bef(condition2)}$) with a computation duration at least equal to C_{Id}.



This can more clearly be represented on the 2-D begin-end plane. Each point that lies within the window satisfying the earliest and latest begin time, earliest and latest end time and the range of permissible object execution durations, is referred to as the *time constraint laxity window*, e.g:



Periodic time constraints, which consist of finite convex subintervals (and combine to form non-finite non-convex intervals) are not modelled by this method. A simplification that is commonly made to apply this approach is to restrict analysis to a limited 'local' region.

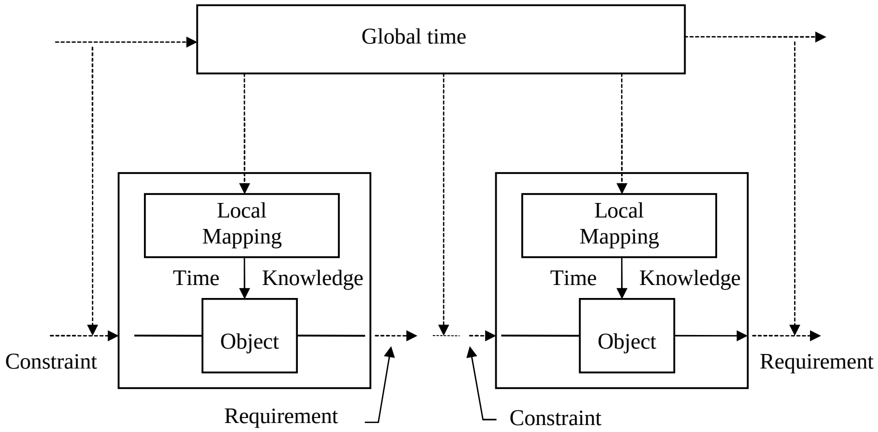
Where we have two periodic objects with periods of T_1 and T_2 time units respectively (where T_1, $T_2 > 1$) then an interval of duration T_1T_2 will contain all possible relations between the subintervals → this can be used as the local region in which to restrict analysis.

Constraint Propagation

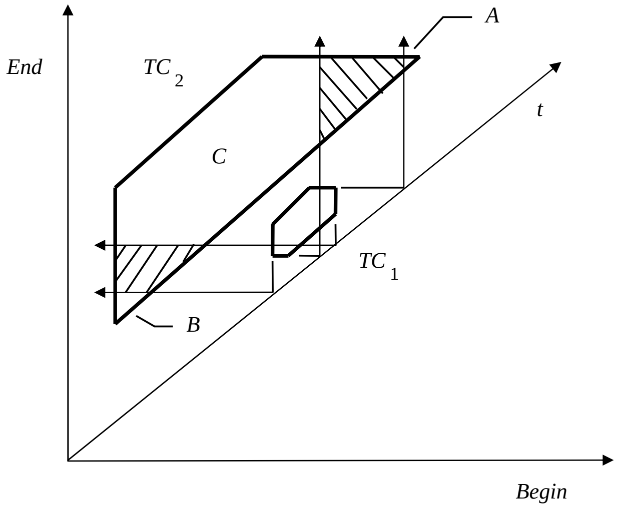
The preceding section considers the handling of uncertainty in describing a global time point in terms of local clocks, but uncertainty can also arise in describing a time point as observed by a remote clock in terms of local time knowledge.

A temporal relationship between two occurrences can be expressed in terms observed by either occurrence, e.g. when a remote client object imposes a time constraint on a server the time reference is either that of the client or the server. Because this requires that the server node has knowledge of the time uncertainties of the remote node, it is not a practical approach in distributed systems.

A better approach is to rely only on local knowledge of uncertainties and work from a common reference for the client and server. Thus the server object interprets the imposed constraint of TC_i through the mapping $TC_i \nrightarrow TC_i'$ with respect to a global time, i.e:



Given this framework for propagating time constraints from invoking objects to become requirements on invoked objects, projection of one time constraint (TC_1) on to another (TC_2) can be looked at in general:



- The latest possible end (TC_1) is projected onto the begin-axis of TC_2 creating a small occurrence window *A*. In *A* the relation $TC_1 \prec TC_2$ holds, i.e. begin(TC_2) > $\text{end}_{\max}(TC_1)$.

- The earliest possible begin (TC_1) is projected onto the end-axis of TC_2 creating a small occurrence window *B*. In *B* the relation $TC_2 \prec TC_1$ holds, i.e. end(TC_2) < $\text{begin}_{\min}(TC_1)$.

- In the region *C*, the intersection of TC_1 and TC_2 is non-null, i.e: end(TC_2) > $\text{begin}_{\max}(TC_1)$ and begin(TC_2) < $\text{end}_{\min}(TC_1)$.

- The other two cross-hatched regions indicate where the temporal relation between the two time constraints requires accurate knowledge of the actual begin and end points, i.e:
  ▧ → $\text{end}_{\max}(TC_1) > \text{begin}(TC_2) > \text{end}_{\min}(TC_1)$
  ▨ → $\text{begin}_{\max}(TC_1) > \text{end}(TC_2) > \text{begin}_{\min}(TC_1)$

Time constraint laxity window contraction with time uncertainty

Assume the time servers use a linear clock interpolation so that the service for a *get-time* request at node p is:

$$T_p(t) = a_p(t)C_p(t) + b_p(t), \qquad t \geq t_p^{(0)}$$

where $C_p(t)$ is the local clock time which synchronizes periodically at least every τ time units with other system clocks. The synchronization times are denoted by the sequence $t_p^{(i)}$ and the bounds on the correct knowledge of $a_p(t)$ and $b_p(t)$ sets the scale and offset in which the time constraint projection is propagated.

Define the terms to express the uncertainty in the local knowledge of a time constraint - let this be TC_i:

$$\Delta a_p = \max \mid 1 - a_p(\mathrm{end}_{\max}(TC_i)) \mid$$
$$\Delta b_p = \max \mid b_p(\mathrm{end}_{\max}(TC_i)) \mid$$
$$\delta a_p = \max \mid 1 - a_p(\mathrm{begin}_{\min}(TC_i)) \mid$$
$$\delta b_p = \max \mid b_p(\mathrm{begin}_{\min}(TC_i)) \mid$$

Suppose the time constraint imposed in terms of local time is TC_i', so that at a node p, an imposed time constraint TC_i maps to the following local bounds on TC_i':

$$\mathrm{begin}_{\min}(TC_i') = \mathrm{begin}_{\min}(TC_i) + \delta a_p\, \mathrm{begin}_{\min}(TC_i) + \delta b_p$$
$$\mathrm{begin}_{\max}(TC_i') = \mathrm{begin}_{\max}(TC_i) - \delta a_p\, \mathrm{begin}_{\max}(TC_i) - \delta b_p$$
$$\mathrm{end}_{\min}(TC_i') = \mathrm{end}_{\min}(TC_i) + \Delta a_p\, \mathrm{end}_{\min}(TC_i) + \Delta b_p$$
$$\mathrm{end}_{\max}(TC_i') = \mathrm{end}_{\max}(TC_i) - \Delta a_p\, \mathrm{end}_{\max}(TC_i) - \Delta b_p$$

The result of the clock variations which cause variations in the local bounds on TC_i', is that the constraint laxity window is reduced in size in all dimensions. The projection of a globally defined time constraint to a locally defined time constraint is denoted by the mapping: $TC_i \mapsto TC_i'$