PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT. © ICOT, 1992

Dealing with Time Granularity in the Event Calculus

Angelo Montanari (+)(*), Enrico Maim (-), Emanuele Ciapessoni (+), Elena Ratto (+)

(+) CISE, Milano, Italy (-) SYSECA, Paris, France
(*) Current Affiliation: University of Udine, Udine, Italy

Abstract

The paper presents a formalization of the notion of time granularity in a logic-based approach to knowledge representation and reasoning. The work is based on the Event Calculus [Kowalski,86], a formalism for reasoning about time, events and properties using first-order logic augmented with negation as failure. In the paper, it is extended to include the concept of time granularity. With respect to the representation, the paper defines the basic notions of temporal universe, temporal decomposition and coarse grain equivalence. Then, it specifies how to locate events and properties in the temporal universe and how to pair event and temporal decompositions. With respect to the reasoning mechanisms, the paper defines two alternative modalities of performing temporal projection, namely upward and downward projections, that make it possible to switch among coarser and finer granularities.

1 Introduction

The paper presents a formalization of the notion of time granularity in a logic-based approach to knowledge representation and reasoning. The work is based on the Event Calculus, a formalism for reasoning about time, events and properties using first-order logic augmented with negation as failure [Kowalski and Sergot 1986]. In the paper, it is extended to include the concept of time granularity. Informally, granularity can be defined as the resolution power of a representation. In general, each level of abstraction at which knowledge can be represented is characterized by a proper granularity. Providing a formalism with the concept of granularity allows it to embed different levels of knowledge in a representation. In such a way, each reasoning task can refer to the representational level that abstracts from the domain only those aspects relevant to the actual goal. We are interested in time granularity. With respect to the *expressive power*, it allows one to maintain the representations of the dynamics of different processes of the domain that evolve according to different time constants as separate as possible [Corsetti *et al.* 1990]. It also allows one to model the dynamics of a process with respect to different time scales. In such a case time granularity has to be paired with other refinement mechanisms such as process decomposition [Allen 1984], [Kautz and Allen 1986], [Corsetti *et al.* 1991a], [Evans 1990]. Finally, time granularity increases both the temporal distinctions that a language can make and the distinctions that it can leave unspecified. This means that considering two events as simultaneous or temporally distinct, or two time dependent relations as temporally overlapped or disjoint depends on the granularity one refers to. With respect to the *computational power*, it supports different grains of reasoning to deal with incomplete and uncertain knowledge [Allen 1983], [Dean and Boddy 1988]. It also allows one to tailor the visibility of the knowledge base and the reasoning process to the needs of the actual task [Fum *et al.* 1989]. Secondly, it allows one to alternate among different time granularities during the execution of a task in order to solve each incoming problem at a time granularity as coarse as possible [Dean *et al.* 1988]. An example of a limited use of time granularity to expedite the search of large temporal databases is provided by [Dean 1989]. Finally, it allows one to solve a problem at a time granularity coarser than the required one to

Authors' addresses: Angelo Montanari, University of Udine, Mathematics and Computer Science Department, Via Zanon, 6 33100 Udine, ITALY, email: montanari@uduniv.cineca.it; Enrico Maim, SYSECA Temps Reel, Constraint Resolution Research Group, 315 Bureaux de la Colline, 92213 Saint-Cloud Cedex, FRANCE, email: enrico_maim@eurokom.ie; Emanuele Ciapessoni and Elena Ratto, CISE, Artificial Intelligence Section, Division of Systems and Models, Via Reggio Emilia, 39, Segrate (Milano), ITALY, email: kant@sia.cise.it and elena@sia.cise.it. Most work of the first author was done while he was employed in CISE.

cope with the complexity of temporal reasoning. Such a simplification speeds up the reasoning, but implies a relaxation of the precision of the solution. The ratio between the time granularities provides a measurement of the approximation of the achieved result.

In despite of the widespread recognition of its relevance for knowledge representation and reasoning, there is a lack of a systematic framework for temporal granularity. The main references are the paper of Hobbs [1985] on the general concept of granularity and the works of Plaisted [1981], Giunchiglia and Walsh [1989] on abstract theorem proving. Hobbs defines a concept of granularity that supports the construction of simple theories out of more complex ones. He formally introduces the basic notions of relevant predicate set, indistinguishability relations, simplification, idealization and articulation. Such notions are extended and refined by Greer and McCalla [1989], which identify two orthogonal dimensions along which granularity can be interpreted, namely abstraction and aggregation. However, the one and the others reserve little or no attention to time granularity. In particular, Hobbs only sketches out a rather restrictive mapping of continuous time into discrete times using the situation calculus formalism. Conversely, a set-theoretic formalization of time granularity is provided by Clifford and Rao [1988], but they do not attempt to relate the truth value of assertions to time granularity. Finally, Galton [1987] and Shoham [1988] give significant categorizations of assertions based on their temporal properties. These categorizations are strictly related to the concept of time granularity even if it is not explicitly considered.

A first attempt to introduce the notion of time granularity in the Event Calculus is reported in [Evans 1990]. Evans defines a macro-events calculus for dealing with time granularity whose limitations are discussed in section 4.1. Our paper proposes a framework to represent and reason about time granularity in the Event Calculus that generalizes these previous results. It significantly benefits by the work done to formalize the concept of time granularity in TRIO, a logic formalism for specifying realtime systems [Corsetti *et al.* 1991b], [Corsetti *et al.* 1991c], [Montanari *et al.* 1991], and [Ciapessoni *et al.* 1992]. [Maim 1991] and [Maim 1992a] present an alternative approach where the granularity problem is seen as an issue of dealing with ranges and intervals in constraint-based reasoning.

The paper is organized as follows: section 2 presents the original Event Calculus together with its basic extensions, namely types, macro-events and continuous change; section 3 focuses on the representation of time granularity; section 4 details the modalities of reasoning about time granularity.

2 The Event Calculus

The Event Calculus proposes a general approach to represent and reason about events and their effects in a logic framework [Kowalski and Sergot 1986], [EQUATOR 1991]. From a description of events that occur in the real world, it allows one to derive various relationships and the time periods for which they hold. It also embodies a notion of *default persistence*, that is, relationships are assumed to persist until an event occurs which terminates them. As an example, if we know that an aircraft enters a given sector at 10:00hrs and leaves at 10:20hrs, the Event Calculus allows us to infer that it is in that sector at 10:15hrs. More precisely, the Event Calculus takes the notions of event, property, time-point and time-interval as primitives and defines a model of change in which *events* happen at *time-points* and initiate and/or terminate *time-intervals* over which some *property* holds. So, for instance, the events of entering and leaving the sector initiate and terminate the aircraft's property of being in the sector, respectively. Time-points are unique points in time at which events take place instantaneously. In the previous example, the event of entering the sector occurs at 10:00hrs, while the event of leaving the sector occurs at 10:20hrs. They can be specified at different degree of explicitness, e.g. "91/5/24:10:00hrs" to include the full date or just "10:00hrs", but belong to a unique domain. Time-intervals are represented by means of tuples of two time-points. With the same example, we can deduce that the aircraft is in the sector during the time-interval starting at 10hrs and ending at 10:20hrs.
Formally, Event Calculus represents domain knowledge by means of *initiates* and *terminates* predicates that express the effects of events on properties[1]:

$initiates(Event, Property)$
$terminates(Event, Property)$

In such a way, domain relations are intensionally defined in terms of events and properties *types* [EQUATOR 1991]. Weak forms of the *initiates* and *terminates* predicates, namely *weak-initiates* and *weak-terminates*, have been introduced in [Sergot 1990]. The predicate *weak-terminates* states that a given event terminates a given property unless this property has been already terminated. In a similar way, the predicate *weak-initiates* states that a given event initiates a given property unless this property has been already initiated.
Instances of events and properties are obtained by attaching a time-point *(event, time-point)* and a time-

[1] We adopt the variable convention of the original Event Calculus where constants are distinguished from variables by being denoted by names beginning with upper-case characters.

interval *(property, time-interval)* to event and property types, respectively.

The first Event Calculus axiom we introduce is the *Mholds-for*. It allows us to state that the property p holds maximally (i.e. there is no larger time-interval for which it also holds) over $\langle start, end \rangle$ if an event e occurs at the time *start* which initiates p, and an event e' occurs at time *end* which terminates p, provided there is no known interruption in between:

Mholds-for(p, ⟨start,end⟩) ←
happens_at(e,start) ∧ initiates(e,p) ∧
happens_at(e',end) ∧ terminates(e',p) ∧
end > start ∧ not broken-during(p, ⟨start,end⟩)

In the above axiom, the negation involving the *broken* predicate is interpreted using negation-as-failure. This means that properties are assumed to hold uninterrupted over an interval of time on the basis of failure to determine an interrupting event. Should we later record a terminating event within this interval, we can no longer conclude that the property holds over the interval. This gives us the non-monotonic character of the Event Calculus which deals with default persistence[2].

The predicate *broken-during* is defined as follows:

broken-during(p, ⟨start,end⟩) ←
happens_at(e,t) ∧ start < t ∧
end > t ∧ terminates(e,p)

This states that a given property p ceases to hold at some point during the time-interval ⟨start,end⟩ if there is an event which terminates p at a time t within ⟨start,end⟩.

Event Calculus also defines an *Iholds-for* predicate in terms of *Mholds-for* to state that a property holds over each time-interval included in the maximal one:

Iholds-for(p, ⟨start,end⟩) ←
Mholds-for(p, ⟨a,b⟩) ∧ start ≥ a ∧ end ≤ b

Finally, Event Calculus defines the *holds-at* predicate which is similar to *Iholds-for* except that it relates a property to a time-point rather than a time-interval:

holds-at(p,t) ←
Mholds-for(p, ⟨start,end⟩) ∧
t > start ∧ t < end

In particular, the *holds-at* predicate states that a property is not valid at the time points at which occur the events that initiate and terminate it. This negative conclusion about the validity of properties at the left and right ends of time-intervals properly stands for ignorance. Time granularity will allow us to refine descriptions with respect to finer temporal domains.

[2] To deal with default persistence, [Maim 1992b] presents an approach to constructive negation in constraint-based reasoning.

2.1 Macro-events to Model Discrete Processes

To model discrete processes the basic Event Calculus has been extended with an event decomposition mechanism that allows us to refine event representations [Evans 1990], [EQUATOR 1991]. Evans introduced the notion of *macro-event*, which is a finite event decomposed into a number of sub-events. The connections between a macro-event and its components are formalized in the Event Calculus as follows:

happens_at(e,t) ←
happens_at(e1,t1) ∧ part_of(e1,e) ∧
happens_at(e2,t2) ∧ part_of(e2,e) ∧
happens_at(e3,t3) ∧ part_of(e3,e) ∧
happens_at(e4,t4) ∧ part_of(e4,e)

where the predicate *part_of* is defined by means of appropriate domain axioms.

This axiom allows us to derive the occurrence of a macro-event from the occurrences of its sub-events. It can also be used to abduce the occurrence of sub-events from the occurrence of the macro-event.

2.2 Continuous Change to Model Continuous Processes

The basic Event Calculus is well-equipped to represent discrete processes, but is not so good for representing continuous processes, i.e. processes characterized by a continuous variation in a quantity such as the height of a falling object or the angular position of a crankshaft. Modelling a continuous process in terms of its temporal snapshots, in fact, can be seen a particular case of event decomposition, but cannot be directly done by means of macro-events. To model continuous processes, Event Calculus has been extended with the idea of the *trajectory* of a continuously changing property through a space of values [Shanahan 1990], [Shanahan 1991], [EQUATOR 1991]. Shanahan introduced the notion of 'dynamic' properties, like *moving* of a train. When such a property holds, another property is continuously changing, such as *position* of the train. Continuously changing properties are modelled as *trajectories*. Formally, the *holds-at* axiom which gives value to a continuously changing property is:

holds_at (p,t2) ←
happens_at(e,t1) ∧ initiates(e,q) ∧ t1 < t2 ∧
not broken_during(q, ⟨t1, t2⟩)∧
trajectory(q,t1,p,t2)

In this axiom, the continuously changing property p can be assigned a given value at a time point t2 if an instance of the relevant dynamic property q is initiated at a time point t1 (before t2) and not broken at some point between t1 and t2. The predicate *trajectory* describes the functional relationship between the continuously changing property and the time that has elapsed since it started to change. It can be seen as a path plotted against time through the corresponding quantity space. The formula $trajectory(q, t1, p, t2)$ represents that property p holds at time $t2$ on the trajectory of the period of continuous change represented by q which start at time $t1$. Such a property p holds only instantaneously and represents that some quantity varying continuously has a particular value. Its definition is *domain specific*. That is, a set of *trajectory* clauses is also part of the description of the domain, along with the domain's *initiates* and *terminates* clauses.

For example, suppose that the angular position of a crankshaft increasing linearly with time whilst the shaft is rotating. If ω is the angular velocity of the crankshaft, we have the following domain axiom:

$$trajectory(rotating, t1, angle(a2), t2) \leftarrow holds\_at(angle(a1), t1) \wedge a2 = \omega(t2 - t1) + a1$$

3 Representing Time Granularity

This section first introduces the notion of temporal universe as a set of related, differently grained temporal domains. Such a notion supports the definition of the relations of indistinguishability and distinguishability among the time-points of the domains. Then, it precisely states the linkage between events and properties, and time granularity.

3.1 The Temporal Universe

Providing a representation with time granularity requires introducing a finite set of disjoint *temporal domains* that constitutes the *temporal universe* of the representation:

$$T = \bigcup_{i=1,..,n} T_i$$

The set $\{T_1, T_2, .., T_n\}$ is totally ordered on the basis of the degree of fineness (coarseness) of its elements and, for each i, with $1 \leq i < n$, T_{i+1} is said of a *finer granularity* than T_i. Each domain is *discrete* with the possible exception of the finest domain that may be dense. For the sake of simplicity, we assume that each domain is denumerable. The temporal universe includes at most one dense domain because each dense domain is already at the finest level of granularity, since it allows any degree of precision in measuring time distances. As a consequence, for dense domains we must distinguish granularity from metric, while for discrete domains we can define granularity in terms of set cardinality and assimilate it to a natural notion of metric[3]. For the sake of simplicity, we assume that each domain is denumerable.

For each pair of domains T_i, T_{i+1}, a mapping is defined that maps each time-point of T_i into a time-interval of T_{i+1} (*totality*). mathematical expressions we use the $succ_i(t)$ denotes the It maps contiguous time-points into contiguous, disjoint time-intervals (*contiguity*) preserving the ordering of the domains (*order preserving*) . Moreover, the union set of the time-intervals of T_{i+1} belonging to its range is equal to T_{i+1} (*coverage*). Finally, we assume that the length of time-intervals into which it maps the time-points of T_i is constant (*homogeneity*). This constant, denoted by $\Delta_{i,i+1}$, defines the *conversion factor* between T_i and T_{i+1} which provides a relative measurement of the granularity of T_i and T_{i+1} with respect to each other. A general mapping between T_i and T_j, with T_i coarser than T_j, can be easily obtained by a suitable composition of a number of elementary mappings. It is formally defined in a recursive way in [Corsetti *et al.* 1991a], where it is also shown that the properties of totality, contiguity, order preserving, coverage and homogeneity are preserved.

In general, there are several ways to define these mappings each one satisfying the required properties. According to the intended meaning of the mappings as *decomposition functions*, each time-point of T_i is mapped into the set of time-points of T_{i+1} that compose it. Nevertheless, we are faced by a number of alternative possibilities in settling the reference time-point of each domain. Choosing the one or the other is merely a matter of convention, but it determines the actual form of the mappings. In the following, assume that, for each pair T_i, T_j, the relevant function maps the reference time-point of T_i into a time-interval of T_j whose first element is the reference time-point of T_j (reference time-points alignment assumption).

To include the notion of temporal universe in the Event Calculus, we introduce the predicate *value-metric* which splits each time-point (1st argument) into a metric (2nd argument) and a value (3rd argument) components. Moreover, we express metrics as a subset of integer. Let us consider a temporal universe consisting of hours, minutes and seconds, and assign by convention the metric 1 to the domain of

[3] "Mapping, say, a set of reals into another set of reals would only mean changing the unit of measure with no semantic effect. Just in the same way one could decide to describe geometric facts by using, say, Kmeters and centimetres. However, if Kmeters are measured by real numbers, the same level of precision as with centimetres can be achieved. Instead, the key point in time granularity is that saying that something holds for all days in a given interval does not imply that it holds every second within the 'same' interval" [Corsetti *et al.* 1991c].

seconds (in general, metric 1 is assigned to the finest domain), the metric 60 to the domain of minutes)1 minute corresponds to 60 seconds) and the metric 3600 to the domain of hours (1 hour corresponds to 3600 seconds). As an example, *value-metric(2hrs30m,60,150)* since there are 60 minutes in an hour. Using the predicate *value-metric*, decomposition functions can be defined as follows:

$$\begin{aligned}&fine\_grain\_of(<t1,t2>,t)\leftarrow\\&\quad value\_metric(t,m,v)\wedge\\&\quad value\_metric(t1,m1,v1)\wedge\\&\quad value\_metric(t2,m1,v2)\wedge m1\leq m\wedge\\&\quad v1=v*(m/m1)\wedge v2=(v+1)*(m/m1)-1\end{aligned}$$

Given a pair of domains T_i, T_j, with T_i coarser grain of T_j, for each time-point t_j of T_j we also define as its *coarse grain equivalent* on T_i the time-point t_i of T_i such that t_j belongs to the time-interval obtained by applying the corresponding decomposition function to t_i. The uniquity of the coarse grain equivalents can be easily deduced from the definition of the decomposition functions. Coarse grain equivalent functions can also be defined using the predicate *value-metric* as follows:

$$\begin{aligned}&coarse\_grain\_of(t2,t1)\leftarrow\\&\quad value\_metric(t1,m1,v1)\wedge\\&\quad value\_metric(t2,m2,v2)\wedge\\&\quad m1\leq m2\wedge v2=(v1*m1)//m2\end{aligned}$$

where $(v1*m1)//m2$ denotes the integer division of $(v1*m1)$ by $m2$.
The relationships of temporal ordering can be generalized to make it possible to compare two time-points belonging to different temporal domains as follows:

$$\begin{aligned}&is\_after(t2,t1)\leftarrow\\&\quad value\_metric(t1,m,v1)\wedge\\&\quad coarse\_grain\_of(t,t2)\wedge\\&\quad value\_metric(t,m,v)\wedge\\&\quad v1<v\end{aligned}$$

$$\begin{aligned}&is\_after(t2,t1)\leftarrow\\&\quad value\_metric(t2,m,v2)\wedge\\&\quad coarse\_grain\_of(t,t1)\wedge\\&\quad value\_metric(t,m,v)\wedge\\&\quad v<v2\end{aligned}$$

The *is_before* predicate can be easily defined in a similar way.

The coarse grain equivalent and the decomposition functions can be viewed as forms of simplification and articulation along the dimension of temporal aggregation, i.e. shifts in focus through part-whole relationships among time-points, respectively. They define distinguishability and undistinguishability relations between any pair of time-points with respect to each domain of the temporal universe.

3.2 Events and Properties in the Temporal Universe

Let us now locate events and properties in the temporal universe. The idea is to directly associate a time granularity with events and to derive the granularity of properties on the basis of the *initiates* and *terminate* relations.
First of all, we give a characterization of events with respect to the temporal universe. With respect to *a given domain*, we distinguish *instantaneous events*, that happen at a time-point, and *events with duration*, that take place over a nonpoint time-interval. Such a distinction among events is a *relative* one, so, for example, passing from a given domain to a finer (coarser) one an instantaneous (with duration) event may become an event with duration (instantaneous).
With respect to *the temporal universe*, we distinguish finite and infinitesimal events. An event is said *finite* if there exists a domain with respect to which it has duration. A finite event thus identifies an implicit level of time granularity: at this level and coarser ones, it is an instantaneous event; at finer levels it is of finite duration. We define such a threshold the *intrinsic time granularity* of the event. An event is said *infinitesimal* if it is instantaneous with respect to every domain[4]. Infinitesimal events are needed for dealing with continuous change [Shanahan 1990]. Let us consider, for instance, a process of continuous change such as sink filling with water. We might associate the occurrence of an event with each new level reached by the filling fluid. If we did this, then there would be no limit to how fine we might choose our temporal grain in order for the events to remain instantaneous. Thus taking this approach we have a need for infinitesimal events. Differently from the previous one, such a distinction among events is an *absolute* one.
To be able to deal with instantaneous events only, we impose that every event is associated with a domain whose granularity is equal to or coarser than the intrinsic one of the event. In such a way, Event Calculus axioms can be still used to reason *within domains*. On the contrary, they are insufficient by themselves to deal with events associated with different domains (*differently grained events*). However, reasoning *across domains* can be brought back to reasoning within domains provided that there exist some rules to relate differently grained events to the same domain. The idea is to integrate macro-events (section 3.3) and continuous change (section 3.4) mechanisms with time granularity, and to define general temporal projection

[4]The absolute instantaneousness of infinitesimal events copes with the same representational problems that suggested to Hayes and Allen the introduction of short time periods (moments) in Allen's Interval Logic [Hayes and Allen 1987].

rules (section 4) that are used by default when neither macro-events nor continuous change decompositions are explicitly given.

3.3 Refining Macro-Events

We define a *unifying framework* for the packaging of events and the granularity of time to describe the temporal relationships between a macro-event and its components. We require that the intrinsic time granularity of a macro-event is coarser than the ones of its sub-events and that its occurrence time is a coarse grain equivalent of the occurrence time of all its sub-events. We also define a number of general operators, called *macro-event constructors*, for specifying temporal relationships among sub-events [EQUATOR 1991](we use the infix notation for macro-event constructors for the sake of simplicity)[5]:

;	sequence
;delay(min,max)	minimum and maximum delay between two events
|	alternative
||	parallelism
$*n$	sequential repetition (n is optional)
$**n$	parallel repetition (n is optional)
[ ]	composition

Let us report here the Event Calculus axiomatization of the basic operators expressing sequence, alternative, and parallelism.

$$\begin{aligned}&happens\_at(e1;e2,t)\leftarrow\\&\quad happens\_at(e1,t1)\wedge happens\_at(e2,t2)\wedge\\&\quad coarse\_grain\_of(t,t1)\wedge coarse\_grain\_of(t,t2)\\&\quad is\_after(t2,t1)\end{aligned}$$

$$\begin{aligned}&happens\_at(e1|e2,t)\leftarrow\\&\quad happens\_at(e1,t)\wedge not\ happens\_at(e2,t)\end{aligned}$$

$$\begin{aligned}&happens\_at(e1|e2,t)\leftarrow\\&\quad not\ happens\_at(e1,t)\wedge happens\_at(e2,t)\end{aligned}$$

$$\begin{aligned}&happens\_at(e1||e2,t)\leftarrow\\&\quad happens\_at(e1,t)\wedge happens\_at(e2,t)\end{aligned}$$

In general, domain axioms include definition of macro-events in terms of a suitable composition of sub-events. An example of these domain axioms is:

$$\begin{aligned}&happens\_at(e,t)\leftarrow\\&\quad happens\_at([e1;[e2||e3]],t)\wedge\\&\quad part\_of(e1,e)\wedge part\_of(e2,e)\wedge part\_of(e3,e)\end{aligned}$$

[5]Dealing with the repetition operators may require the addition of a domain composed of a single time point to the temporal universe (absolutely coarsest domain).

The operator expressing sequence deserves further consideration. It allows us to deduce the occurrence of a macro-event at a time-point of a domain coarser than the domain(s) the occurrence times of its component events (possibly macro-events in their turn) belong to. Such a time-point is a coarse grain equivalent of both the occurrence times of components. Then, the rule for sequential macro-events first executes a comparison of time-points with respect to the finer domain and then it abstracts them into a time-point of the coarser one. The presence of this switching to a coarser domain makes the definition of sequential macro-events *incomplete*. Consider the following example. Given the occurrences of three events $e1$, $e2$ and $e3$ at time-points $2hrs15m$, $2hr42m$ and $2hrs50m$, respectively, we are not able to deduce the occurrence of a sequential event $e1;[e2;e3]$ at time-point $2hrs$ when the temporal universe is {..., hours, minutes, ...}. In fact, there is no way of strictly ordering $e1$ and the macro-event into which $e2$ and $e3$ can be abstracted, because the occurrence time of the macro-event is a coarse grain equivalent of the occurrence time of $e1$. To make it possible to derive the occurrence the macro-event $e1;[e2;e3]$, the temporal universe has to be extended with the domain of 30-minutes (similar considerations hold for the macro-event $[e1;e2];e3$). However, it is easy to find another sequence that cannot be abstracted into a sequential macro-event with respect to the extended temporal universe too. Such an incompleteness is due to the fact that mappings between temporal domains are fixed once and for all and then is inherent to the upward temporal projection involved in macro-event derivation rules (section 4.1).

3.4 Refining Continuous Change

The original approach to continuous change makes the assumptions that the parameters of the trajectory function are set not after $t1$ and are not reset between $t1$ and $t2$. In general, these assumptions are too restrictive. Mechanisms are requested for resetting the parameters of the trajectory function. This allows it to be initiated with parameters values at the start of the property, but also allows the parameters to be changed during the interval of validity of the property. In this way, the trajectory may model 'non-linearities' (e.g. a change in the rate of a linear increase of a temperature) without interrupting the relevant dynamic property (e.g. by splitting a 'temperature rising' property when the rate of rise changes).

To take into account the resetting of parameters the original axiom can be replaced by the following one:

$holds\_at(p,t) \leftarrow$
$value\_metric(t, m, v) \wedge$
$happens\_at(e, t1) \wedge value\_metric(t1, m, v1) \wedge$
$v1 < v \wedge initiates(e, q) \wedge$
$not\ broken\_during(q, \langle t1, t \rangle) \wedge$
$happens\_at(e', t2) \wedge value\_metric(t2, m, v2) \wedge$
$v2 < v \wedge initiates(e', par) \wedge$
$not\ broken\_during(par, \langle t2, t \rangle) \wedge$
$max(t1, t2, ti) \wedge trajectory(q, par, ti, p, t)$

A continuously changing property p can be assigned a given value at a time point t if an instance of the relevant dynamic property q is initiated at a time point t1 (before t) and not broken at some point between t1 and t, the relevant parameter par is (re)set at a time point t2 (before t) and not broken at some point between t2 and t, and the initial value of p is calculated (by the *trajectory* predicate) at the time point ti which is the maximum between t1 and t2 and the 'max' predicate has the obvious definition. The crankshaft example of section 2.2 must be rewritten according to the revisited axiom as:

$trajectory(rotating, velocity(\omega), ti, angle(a), t) \leftarrow$
$value\_metric(ti, m, vi) \wedge value\_metric(t, m, v) \wedge$
$holds - at(angle(ai), ti) \wedge a = \omega(v - vi) + ai$

The indirect recursion on the predicate *trajectory* (or, equivalently, on the predicate *holds_at*) stops when the initial values of the configuration variables, e.g. the angular position, are reached. They can be explicitly asserted or derived from the occurrence of independent events.

The application of the refined axiom for continuous change is not restricted to discrete resetting of parameters; it can be used to deal with continuously changing parameters too. In such a case, the occurrence of the *continuous events* of resetting can be derived from the continuous change of the configuration variables by means of appropriate domain axioms.

Continuous events can be either acquired by the external environment or computed according to explicit laws. In both cases, we generally need to plot them at regular time intervals to make the model computable. Choosing the width of the time interval is equivalent to choosing the time granularity at which describing the process. Then, a change in the frequency of plotting is equivalent to the switching of a continuous process from one time granularity to another.

4 Reasoning with Time Granularity

We distinguished two basic modalities of relating differently grained events, namely upward and downward temporal projections. Upward (downward) projection determines the temporal relations set up by two events e_i and e_j which occur at the time-points $t_i \in T_i$ and $t_j \in T_j$, respectively, with T_i coarser than T_j, by upward (downward) projecting e_j (e_i) on T_i (T_j).

4.1 'Naive' Upward Projection

The 'naive' upward projection is a quite straightforward approach to abstractive temporal reasoning. It states that the upward projection of an event e that occurs at a time-point t of a domain T_j on a domain T_i, coarser than T_j, is accomplished by simply replacing t with its coarse grain equivalent on T_i [Evans 1990]. Then the temporal ordering and distance between two events e_i and e_j which occur at the time-points $t_i \in T_i$ and $t_j \in T_j$, respectively, are determined on the basis of the relation between t_i and the coarse grain equivalent of t_j on T_i. Moreover, if e_i (e_j) precedes e_j (e_i) then the properties initiated by e_i (e_j) and terminated by e_j (e_i) hold over the time-interval of T_i identified by t_i and the coarse grain equivalent of t_j. To formalize upward projection in the Event Calculus, we first extend the definition of the occurrence time of an event as follows:

$happens\_at(e, t1) \leftarrow$
$happens\_at(e, t2) \wedge coarse - grain - of(t1, t2)$

In this way, each event is endowed with several occurrence times belonging to different domains, i.e. the time-point at which it originally occurs and all the coarse grain equivalents of such a point. Combined with the macro-event derivation rules, upward projection allows us to deduce the occurrence of parallel and alternative macro-events at time-points of domains coarser than the domains at which occur their components.

Upward projection can be seen as a *simplification* rule [Hobbs 1985], because it allows us to derive a relation of temporal indistinguishability, i.e. simultaneity, among events from the relation of indistinguishability among time-points defined by coarse grain equivalent functions.

Then, the *Mholds-for* predicate is redefined to constrain the starting and the ending time-points of the time-interval to belong to the same domain:

$Mholds\text{-}for(p, \langle start, end \rangle) \leftarrow$
$happens\_at(e, start) \wedge initiates(e, p) \wedge$
$value\text{-}metric(start, m, vs) \wedge$
$happens\_at(e', end) \wedge terminates(e', p) \wedge$
$value\text{-}metric(end, m, ve) \wedge$
$vs < ve \wedge not\ broken\text{-}during(p, \langle start, end \rangle)$

together with a similar axiom for the predicate *broken-during*.

In such a way, the predicate *Mholds-for* identifies several time-intervals of differents domains over which the properties initiated and terminated by two differently grained events hold.

In despite of its apparent simplicity upward projection involves a number of semantic assumptions. The most relevant one is related to its application to contradictory events, i.e. events that cannot occur simultaneously. We formally define two events as contradictory if they initiate or terminate incompatible properties. The definition of the relation of incompatibility among properties depends on domain-specific knowledge [Kowalski and Sergot 1986].

Upward projection maintains the weak temporal ordering between events, but it does not always preserve the strict one. Then the logical consistency of the upward projection cannot be guaranteed in the general case, because it may enforce contradictory events to occur at the same time-point in a coarser domain. As a consequence, if two differently grained events are contradictory the coarse grain equivalent of the occurrence time of the fine grained event must be different from the occurrence time of the coarse-grained event. This is guaranteed by the following integrity constraint[6]:

$$\leftarrow happens\_at(e1, t) \wedge happens\_at(e2, t) \wedge contradictory(e1, e2)$$

Moreover, upward projection may change the ratio between the width of time-intervals. That is, given two domains T_i and T_j, with T_i coarser than T_i, the coarse grain equivalents on T_i of two pairs of time-points of T_j which are at the same temporal distance may be at a different one, while the coarse grain equivalents on T_i of two pair of time-points of T_j that are at a different temporal distance may be at the same one.

Such a weakness of the 'naive' upward projection will be overcome refining upward projection according to the downward projection schema we are going to define.

4.2 Downward Projection

The downward projection of an event e that occurs at a time-point t of a domain T_i on a domain T_j finer than T_i is accomplished by applying the following decomposition scheme: for each event e that occurs at a time-point t of T_i there exist two infinitesimal events e_i and e_f that occur at the time-points t_i and t_f of T_j, respectively, and such that (i) $t_i \leq t_f$; (ii) t is the coarse grain equivalent on T_i of both t_i and t_f; (iii) for each property p such that p is terminated by e, there exist an event e_p that occurs at t_p of T_j such that e_p terminates p and $t_i \leq t_p \leq t_f$; (iv) for each property q such that q is initiated by e, there exist an event e_q that occurs at t_q of T_j such that e_q initiates q and $t_i \leq t_q \leq t_f$; (v) the (type of the) event e becomes a dynamic property that is initiated by e_i and terminated by e_f with respect to T_j. Because of an event is defined by the properties that it initiates and terminates, such rules provide the definition of the component events e_i, e_f, e_p and e_q[7].

Downward projection can be seen as an *articulation* rule [Hobbs 1985]. From the relation of distinguishability among time-points of the finer domain introduced by the decomposition function, in fact, it derives a relation of temporal distinguishability among the subevents of a given finite event.

Let us formalize this scheme in the Event Calculus. First of all, we define two functions *begin* and *end* that map a given instance of a macro-event into its initiating and terminating events, respectively. The occurrence of such events can be deduced from the occurrence of the macro-event by means of the following axioms:

$$happens\_at(begin(e,t), time(begin(e,t))) \leftarrow happens\_at(e,t) \wedge coarse-grain-of(t, time(begin(e,t)))$$

$$happens\_at(end(e,t), time(end(e,t))) \leftarrow happens\_at(e,t) \wedge coarse-grain-of(t, time(end(e,t)))$$

together with (condition (ii)):

$$coarse-grain-of(t, time(begin(e,t)))$$
$$coarse-grain-of(t, time(end(e,t)))$$

where $time(begin(e,t))$ and $time(end(e,t))$ denote the occurrence times of $begin(e,t)$ and $end(e,t)$, respectively.

Condition (i) is expressed by the following integrity constraint:

$$\leftarrow is\_after(time(begin(e,t)), time(end(e,t)))$$

Let us now represent e_p and e_q by means of two functions *term* and *in*. For each property p (q), *term* (*in*) maps each instance of a given macro-event into the component event that terminates (initiates) such a property. Using these functions, conditions (iii) and (iv) are codified by the following axioms:

[6] This solution can be generalized by making contradiction dependent on granularities or even on time instants. In such a way, simultaneous occurrence of two events can be classified as contradictory in certain domains, or even in certain time instants of them, only.
The relevant integrity constraint becomes:

$$\leftarrow happens\_at(e1, t) \wedge happens\_at(e2, t) \wedge contradictory(e1, e2, t)$$

[7] When t_i and t_f coincide, the events e_i, e_p, e_q and e_f are merged into the original single event e. This is always the case of the downward projection of infinitesimal events. For instance, the infinitesimal event of switching on the light remains instantaneous with respect to all the domains of the temporal universe composed of {Day, Hour, Minute}.

$terminates(term(e,t,p),p) \leftarrow terminates(e,p)$
$initiates(in(e,t,q),q) \leftarrow initiates(e,q)$

together with:

$\leftarrow is\_after(time(begin(e,t)), time(term(e,t,p))) \vee$
$is\_before(time(end(e,t)), time(term(e,t,p)))$

$\leftarrow is\_after(time(begin(e,t)), time(in(e,t,q))) \vee$
$is\_before(time(end(e,t)), time(in(e,t,q)))$

for each property p and q.
Finally, condition (v) is expressed by the following axioms:

initiates(begin(e,t),e)
terminates(end(e,t),e)

They allow us to state that the property e holds over $\langle time(begin(e,t)), time(end(e,t))\rangle$ by means of the *Mholds-for* axiom. These last axioms provide each temporal object with a twofold event/property characterization. That is, (the type of) an event e, associated with a given domain, may become a dynamic property with respect to a finer domain, and vice versa.
Let us consider, as an example, the event of flying from Milan to Venice. With respect to the domain T_H of *hours* it can be modeled as an instantaneous event that occurs at a time-point t of T_H. Such an event *terminates* the property of *being in Milan* and *initiates* the property of *being in Venice*. With respect to the domain T_M of *minutes*, it can be decomposed into a pair of infinitesimal events $flying_i$ and $flying_f$ that occur at the time-points t_i and t_f of T_M, respectively, with $t_i \leq t_f$, and such that t is the coarse grain equivalent of both. Moreover, $flying_i$ *terminates* the property of *being in Milan* and *initiates* the property of *flying*, while $flying_f$ *terminates* the property of *flying* and *initiates* the property of *being in Venice*.

4.3 'Revised' Upward Projection

The event/property duality introduced by downward projection suggests an extension of the upward projection rules to cope with contradictory events without restrictions. When the coarse grain equivalents of two contradictory events coincide the downward projection schema suggests to merge and replace the events by a *macro-event* corresponding to the conjunction of the properties initiated by the first one and terminated by the second one. Moreover, such a macro-event terminates (initiates) all the properties terminated (initiated) by its first (second) component and every property terminated (initiated) by the second (first) component which is not initiated (terminated) by the first (second) component.
Let us consider, as an example, the events of *leaving station A* and *arriving at station B* of a train. The first one *terminates* the property of the train of *being at station A* and *initiates* the property of *moving*, while the second one *terminates* the property of *moving* and *initiates* the property of *being at station B*. Let be T a domain with respect to which the two events are simultaneous. According to the revised upward projection rules they are merged and replaced by the event of *moving* that *terminates* the property of *being at station A* and *initiates* the property of *being at station B*.

The actual structure of the corresponding macro-event can be given in terms of a *suitable composition* of the component events using macro-event constructors. Consider two contradictory events $e1$ and $e2$. If their temporal ordering is known and meaningful, e.g. $e1$ precedes $e2$, then the corresponding macro-event e is a sequential one, that is, $e1; e2$; if their temporal ordering is meaningless (their global effect does not change even if their ordering changes), and possibly unknown, then the corresponding macro-event is a parallel one, that is $e1||e2$; if their temporal ordering is meaningful and unknown, then the corresponding macro-event is $[e1||e2]|[[e1;e2]|[e2;e1]]$; and so on.
The last one is the case, for instance, of events of rotation around orthogonal axes in the three dimensional space which are not commutative, that is, the final configuration of the rotating system depends on the ordering of their occurrences.

5 Conclusion

The paper made a proposal for embedding the notion of time granularity into a logic-based representation language. Firstly, it enumerated a number of notational and computational reasons that motivate the introduction of time granularity and briefly surveied and discussed the existing relevant literature. Successively, it extended the Event Calculus to deal with time granularity by introducing the concepts of temporal universe, finite and infinitesimal events, macro-event, and continuously changing events and properties. Finally, it provided Event Calculus with the axioms supporting upward and downward temporal projection.

Acknowledgements

We would like to thank Chris Evans of Goldsmiths' College, University of London, and Murray Shanahan of Imperial College for the useful discussions we had with them.
The research for this paper was partially funded by the European Community ESPRIT Program, EQUATOR Project no. 2409 [EQUATOR 1991]. Collaborating organizations are CENA (France), CISE (Italy), EPFL (Switzerland), ERIA (Spain), ETRA (Spain), Ferranti

Computer Systems Ltd. (UK), Imperial College (UK), LABEN (Italy), Politecnico di Milano (Italy), SWIFT (Belgium), SYSECA (France), UCL (UK). The work of CISE was partially funded by the Automatica Research Center (CRA) of the Electricity Board of Italy (ENEL) within the VASTA project too.

References

[Allen 1983] Allen, J., *Maintaining Knowledge about Temporal Intervals*; Communications of the ACM, 26, 11, 1983.

[Allen 1984] Allen, J., *Toward a General Theory of Action and Time*, Artificial Intelligence, Vol. 23, No. 2, July 1984.

[Clifford and Rao 1988] Clifford,J., Rao,A., *A Simple, General Structure for Temporal Domains* in Temporal Aspects in Information Systems, Rolland, C., Bodart, F., Leonard, M. (Eds.), IFIP 1988.

[Ciapessoni *et al.* 1992] Ciapessoni,E., Corsetti, E., Montanari,A., San Pietro,P., *Embedding Time Granularity in a Logical Specification Language for Synchronous Real-Time Systems*; submitted to Science of Computer Programming, North-Holland, January 1992.

[Corsetti *et al.* 1990] Corsetti, E., Montanari, A., Ratto, E., *A Methodology for Real-Time System Specifications based on Knowledge Representation*; in Computational Intelligence, III, N. Cercone, F. Gardin, G. Valle (Eds.), North-Holland, Proc. of the International Symposium, Milan, Italy, 24-28 September, 1990.

[Corsetti *et al.* 1991a] Corsetti, E., Montanari, A., Ratto, E., *Dealing with Different Time Granularities in Formal Specification of Real-Time Systems*; The Journal of Real-Time Systems, Vol. III, Issue 2, June 1991.

[Corsetti *et al.* 1991b] Corsetti, E., Montanari, A., Ratto, E., *Time Granularity in Logical Specifications*, Proc. 6th Italian Conference on Logic Programming, Pisa, Italy, June 1991.

[Corsetti *et al.* 1991c] Corsetti, E., Crivelli, E., Mandrioli, D., Montanari, A., Morzenti, A., San Pietro, P., Ratto, E., *Dealing with Different Time Scales in Formal Specifications*; Proc. 6th International Workshop on Software Specification and Design, Como, Italy, October 1991.

[Dean and Boddy 1988] Dean, T., Boddy, M., *Reasoning about partially ordered events*; Artificial Intelligence, 36, 1988.

[Dean *et al.* 1988] Dean, T., Firby, R., Miller, D., *Hierarchical Planning involving deadlines, travel time and resources*; Computational Intelligence, 4, 1988.

[Dean 1989] Dean, T., *Using Temporal Hierarchies to Efficiently Maintain Large Temporal Databases*; Journal of ACM, 36, 4, 1989.

[Evans 1990] Evans, C., *The Macro-Event Calculus: Representing Temporal Granularity*; Proc. PRICAI, Japan 1990.

[Fum *et al.* 1989] Fum, D., Guida, G., Montanari, A., Tasso, C., *Using Levels and Viewpoints in Text Representation*; in Artificial Intelligence and Information-Control Systems of Robots - 89, North-Holland, I. Plander (Ed.), Proc. 5th International Conference, Strbske Pleso, Czechoslovakia, 6-10 November, 1989.

[Galton 1987] Galton, A., *The Logic of Occurrence*; in Temporal Logics and their applications, Galton A., (Ed.), Academic Press, 1987.

[Giunchiglia and Walsh 1989] Giunchiglia, F., Walsh, T., *Abstract Theorem Proving*; Proc. 11th IJCAI, Detroit, USA, 1989.

[Greer and McCalla 1989] Greer, J., McCalla, G., *A Computational Framework for Granularity and its Application to Educational Diagnosis*; Proc. 11th IJCAI, Detroit, USA 1989.

[EQUATOR 1991] *Formal Specification of the GRF and CRL*; CISE, FERRANTI, SYSECA (Eds.), ESPRIT Project no. 2409 EQUATOR, Deliverable D123-1, 1991.

[Hayes and Allen 1987] Hayes, P., Allen, J., *Short Time Periods*; Proc. 10th IJCAI, Milano, Italy 1987.

[Hobbs 1985] Hobbs, J., *Granularity*; Proc. 9th IJCAI, Los Angeles, USA 1985.

[Kautz and Allen 1986] Kautz, H., Allen, J., *Generalized Plan Recognition*; Proc. AAAI, 1986.

[Kowalski and Sergot 1986] Kowalski, R., Sergot, M., *A Logic-based Calculus of Events*; New Generation Computing, 4, 1986.

[Maim 1991] Maim, E., *Reasonig with Different Granularities in CRL*; Proc. IMACS '91, Dublin, Ireland, July 1991.

[Maim 1992a] Maim, E., *Uniform Event Calculus with Constructive Negation*; to appear in Artificial Intelligence, Expert Systems, and Symbolic Computing, Houstis,E.N., and Rice,J.R. (Eds.), Elsevier North-Holland, 1992 (a reprint of this paper will be sent by the author on request).

[Maim 1992b] Maim, E., *Abduction and Constraint Logic Programming*; to be presented at the Logic Programming in AI ESPRIT Compulog-Net Workshop, London, 23-24 March, 1992.

[Montanari *et al.* 1991] Montanari, A., Ratto, E., Corsetti, E., Morzenti, A., *Embedding Time Granularity in Logical Specification of Real-Time Systems*; Proc. 3rd IEEE Euromicro Workshop on Real-Time Systems, Paris, France, June 1991.

[Plaisted 1981] Plaisted, D., *Theorem Proving with Abstraction*; Artificial Intelligence, 16, 1981.

[Sergot 1990] Sergot, M., *(Some Topics in) Logic Programming in AI*; Advanced School on Foundations of Logic Programming, Alghero, Italy, 1990.

[Shanahan 1990] Shanahan, M., *Representing Continuous Change in the Event Calculus*; Proc. 9th ECAI, Stockholm, Sweden 1990.

[Shanahan 1991] Shanahan, M., *Towards a Calculus for Temporal and Qualitative Reasoning*; Proc. AAAI Symposium, Stanford, 1991.

[Shoham 1988] Shoham, Y., *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*; MIT Press, 1988.