Logical Specification of Real-Time Granular Systems in an Object Oriented Language

Emanuele Ciapessoni, Edoardo Corsetti, Manlio Migliorati, Elena Ratto

CISE - Tecnologie Innovative -spa- via Reggio Emilia 39, Segrate (Mi) -Italy
edoardo@sia.cise.it

Abstract*

This paper presents the semantics of TRIO*, an object oriented language devoted to specify real-time systems referring to different time granularities. Time granularity allows to describe the behavior and the properties of a system and its environment with respect to different time scales. TRIO* semantics is expressed by translation into a logical framework supporting the notion of time granularity. Such a semantics provides the executability of object oriented specifications.

1. Introduction

The aim of the paper is to present the semantics of TRIO*, an object oriented language, devoted to specify real-time systems, in which several temporal granularities can be referred. The semantics is provided by means of translation into a logical framework supporting the notion of time granularity [Ciapessoni,92], and this paper focuses on the translation of temporal aspects. This allows to get both a formal semantics and the logical executability of the object oriented framework. The logical language supports the specification of granular real-time systems. It is a revision of previous time granularity formalization proposals [Corsetti,91a/91b], based on a metric temporal logic named TRIO [Ghezzi,90]. Temporal granularity is a significant issue in AI, both in a theoretical and applicative perspective. From a theoretical point of view, dealing with time granularity requires to structure temporal models into differently grained components and to formally define their relationships. From an applicative point of view, temporal granularity plays a major role in several domains as temporal data base, planning, scheduling, diagnosis and natural language understanding. The notion of granularity allows to embed different levels of knowledge in a representation language and refers to the level that abstracts from the domain only those aspects relevant to the actual goal. In particular, temporal granularity enhances both the *expressive* and the *computational* power of a knowledge representation formalism.

About the former, time granularity allows to represent the dynamics of different processes according to different time constants as separate as possible [Corsetti,91a], and to model the dynamics of a process with respect to different time scales.

About the latter, time granularity supports different grains of reasoning. In such a way it allows to deal with incompleteness and uncertainty of knowledge [Allen,83]. Further, it allows to switch among different temporal granularities during the execution of a task in order to solve each incoming problem at a temporal granularity as coarse as possible [Dean,88]. Such a switching among temporal granularities requires the definition of a number of *simplification* and *articulation* rules [Hobbs,85], [Greer,89]. It minimizes the computational complexity of the problem solving process. The simplification, induced by the minimization, speeds up the reasoning, but implies a relaxation of the precision of the solution [Levesque,86]. The ratio between the temporal granularities provides a measurement of the approximation of the achieved result.

The main issue addressed in this paper is the semantics of the temporal aspects of the object oriented language expressed into the granular logical language. The object oriented language extends the temporal logic one with modular and abstraction primitives in order to deal with the specification in the large. With respect to other framework that deal with the description of actions and change, our proposal is more general. Indeed, it allows to deduce assertions by means of classical inference rules, without any mechanism of default reasoning [McCarthy,69], [Kowalski,86] and [Montanari,92]. The semantics is provided by objects translation into logical formulae, and a number of rules stating the theory articulation and simplification.

The semantics of the object oriented language allows the verification of specifications. That is, a verification that ensures the consistency and the adequacy of the specification, at each step of the incremental development.

The paper is organized as follows: section 2 presents how to deal with time granularity, section 3 briefly sketches the logical granular language, and section 4 presents the object

The work is funded by the Centro Ricerche Automauca (CRA) of the Electricity Board of Italy (ENEL), and partially by the National Research Council (CNR), within Piano Finalizzato Informatica e Calcolo Parallelo.

oriented extension and the main semantic issues regarding time granularity.

2. Assertions over several time granularities

It is possible to point out a wide class of systems, whose main characteristics are time critical response to external stimuli and dynamic behavior regulated by very different time constants. We call such class of systems real-time *granular systems.*

Let us consider as an example a simplified specification of a controller of a pondage-power plant. Such specification requires the definition of the temporal constraints: each action affecting the system has a specific time constant, which is the time needed for its completion. For instance, *the filling* the *empty-reservoir,* with a given input of water, takes (about) two months, whereas the *closing* an *open-sluice-gate* takes one minute.

The description of a granular system in a language without any abstraction structure and with just one temporal domain (of instants, points), constrains to specify the whole system with respect to the finest time granularity, to avoid loss of information. Thus, if in the previous example the finest temporal domain is the domain of seconds then, *closing* takes 60 seconds and *filling* takes 2*30*24*60*60 = 5 * 10* seconds (assuming 30 days a month).

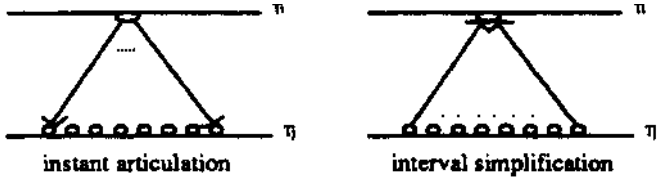
Dealing with different time granularities improves the naturalness of system description, simplifying its model, but involves difficult semantics problems. Referring to the specification of a company as an example, the underlying assertions in: *"Every month, if an employee works, then he. gets his salary",* cannot hold over all instants of each temporal domain finer than months. A unifying model for the above sentences provides a set of temporal domains (including at least months and days) and a refinement mechanism to relate a formula asserted on a domain to finer, or coarser, ones. Such a mechanism must allow that if an *employee works* during a month, then *he works* at most during 22 days of this month, whereas, *he gets his salary* in just one day during the month.

We summarize the steps that provide time granularity in a temporal logic language: to replace a single temporal domain with a temporal universe; to allow the contextualization of a formula over a temporal domain, the assertion of a formula over time instants in correspondence with the current one, and the definition of rules to relate assertions to several (finer or coarser) temporal domains.

- the temporal universe

The *temporal universe* (T) is a set of instants (points), and includes a finite number of temporal domains ($T_1..,T_k$). Temporal domains of the universe are totally ordered with respect to the binary granularity relation ≺, such that if ≺ Tj then T_t is *coarser* than Tj, or Tj is *finer* than T_l

Among time instants and temporal domains three relations are stated. The first relation allows to *contextualize* an instant with respect to a temporal domain, the second one expresses a *displacement* between each couple of instants of a temporal domain and the third one puts into *correspondence* instants belonging to the temporal universe. The contextual relation is specified so that the set of temporal domains constitutes a partition of the temporal universe. Further, the displacement relation is specified by a number of properties so that the notion of metric temporal logic ([Koymans,90]) is supported by any temporal domain. The correspondence relation is subject to a number of properties that constrain time instants and temporal domains to specify a certain kind of temporal universe. For instance given the temporal domains of years, months and days, the correspondence of a calendric-universe can be specified. The correspondence between each couple of temporal domains obeys to the *instant-articulation* and *interval-simplification* rules. Giving two temporal domains such that Tj \ Tj, each time instant of the coarser one is put into correspondence with an interval of time instants of the finer one, and vice versa. Next figure depicts the rules.



For instance, according to such a rule it is possible to state the correspondence between a minute and its 60 corresponding seconds. In order to get the specification of a real universe, a new relation between temporal domains is needed. The relation of disjointedness states that given two temporal domains such that *Ti i T*j, if each interval simplification of Tj over T_t is disjoined from the other ones, then Tj Z Tj holds too. For instance, month articulations over weeks are not disjoined, whereas month articulations over days are disjoined. In fact there can be a week corresponding to two months, but each day corresponds to one, and only one, month. The whole set of properties stated over temporal universe relations are formally defined in [Ciapessoni,92].

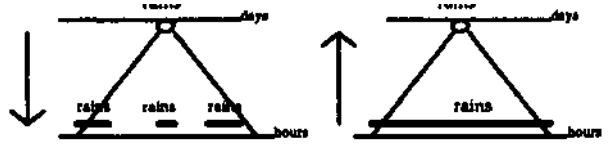
- contextualization, correspondence and projection

In a temporal logic language each formula implicitly refers to a time instant belonging to the underlying temporal domain ([Ghezzi,90], [Rescher 71] and [Pnueli,81]); replacing the temporal domain with a temporal universe is useful to specify which temporal domain the current instant of a formula belongs to, and to assert the truth of a formula over instants in correspondence with the current one. To this aim a couple of intensional operators are needed. For instance, given the assertion *"rains",* it can be contextualized either to the minutes temporal domain or to the days temporal domain (and so on), and it is possible to assert over a day that there is a corresponding hour over which it *"rains".*

Temporal logic languages can be thought as a way to qualify the truth of classical first order formulae with respect to instants. Thus, in a temporal logic language that

deals with time granularity, first order formulae are asserted over instants belonging to several domains. This means that first order formulae belonging to instants between which the correspondence relation is stated should be put into correspondence. Correspondence between first order theories is stated by *projection rules.* These rules can be classified in *downward projection,* i.e., *theory articulation,* and *upward projection,* i.e., *theory simplification.* The former specifies the correspondence between a theory true over an instant and a theory true over the instant articulations on finer domains. The latter specifies the correspondence between a theory that holds over an interval and a theory that holds over any interval simplifications on coarser domains.

For instance, let us consider the case in which *"rains"* holds during a day and we want to know if *"rains"* holds or not during the corresponding hours. Downward projection can be specified, in a *weak* way (sequences), stating that if an assertion holds over an instant, it holds at least over one instant of any articulation. Upward projection can be specified, in a *strong* way (pervasive), stating that if an assertion holds over any instant of an interval then it holds over any interval simplification.



downward (weak) upward (strong)

Other downward projections can be defined: the projection states the assertion holds over just an articulation instant *(punctual),* over any articulation instant *(pervasive)* and over a bounded set of articulation instants *(bounded).* Previous rules state a monotonic projection. On the other hand, alternative upward projection rules can lead to semantic problems. Let us consider the (weak) rule: if an assertion holds over an articulation instant, then it holds over the corresponding simplifications. Thus, if during the current hour *"rains"* holds and during the previous hours *"not-rains"* holds, then, applying twice the *(weak)* upward projection rule we can contradictorily conclude that today *"rains"* and *"not-rains".*

The projection rules sketched above can be compared to [Allen,83] and [Shoham,88].

3. The logical granular language

The typed temporal logical language is sketched throughout a brief survey of the syntax, the semantics and the axiomatization; for a deep and complete description we refer to [Ciapessoni,92]. The syntax includes the first order terms and formulae, and those terms and formulae referring to temporal aspects. Among the latter ones, we take into account just those that will be used in the next section. The language provides a *displacement operator,* a *context operator* and a *correspondence* operator (and dual ones). The displacement operator $\nabla_\alpha\Phi$ ($\Delta_\alpha\Phi =_{def} \neg\nabla_\alpha\neg\Phi$) takes as arguments a formula and a temporal term denoting a distance. $\nabla_\alpha\Phi$ is true if Φ is true α-instants far away the current one. The contextual operator $\nabla^A\Phi$ ($\Delta^A\Phi =_{def} \neg\nabla^A\neg\Phi$) takes as arguments a formula and a temporal term denoting a temporal domain (i.e., a granularity). $\nabla^A\Phi$ restricts the evaluation of Φ to time instants belonging to context A only. The correspondence operator $\Box\Phi$ ($\Diamond\Phi =_{def} \neg\Box\neg\Phi$) takes as argument a formula. $\Box\Phi$ is true if Φ is true in any instant in correspondence with the current one. From the previous temporal operators two other ones can be defined, whose argument terms denote a context and a distance. The *contextual and displacement* operator, $\nabla^A{}_\alpha\Phi \equiv \nabla^A\nabla_\alpha\Phi$ ($\Delta^A{}_\alpha\Phi$) and the *context-displacement-correspondence* operator, $\Box^A{}_\alpha\Phi \equiv \Box\nabla^A{}_\alpha\Phi$ ($\Diamond^A{}_\alpha\Phi$). About those predefined predicates defined in the language, and relevant to the next section, we mention the binary predicates $\wr$, $\angle$, whose arguments are temporal terms denoting contexts, and whose meaning was explained in section 2.

A *granular system specification* is a formula of the kind:

$$\forall\alpha\forall A\Box^A_\alpha\Sigma$$

where Σ is a closed formula; beside the notion of specification we define the one of *granular system history,* that is a temporal evolution of the granular system with respect to a clipping temporal universe, it is expressed by a formula of the kind:

$$\exists\alpha\exists A\Diamond^A_\alpha H$$

where H is a conjunction of formulae like that *Op* η, η is a ground atom, and *Op* is a temporal operator whose terms are bounded or ground.

The semantics of the language is defined according to the notion of *temporal structure.* Such a structure is composed by a *temporal universe* (T), a set of *interpretation domains* and a set of *interpretation functions* ($\Im$). The temporal universe sketched in section 2 is a set of time instants and a set of temporal domains. Among instants and temporal domains the relation of *displacement* (+), *contextualization* (:) and *correspondence* ($\rightarrow$) are stated. The set of interpretation domains is typical of any typed language. Each interpretation function assigns a value to language terms and formulae. First order interpretation rules are the usual ones (we assume rigid designators for variables, and non-rigid designators for constants). Let us formally define the semantical interpretation of each temporal operator (and its dual one):

$$\Im_\iota(\nabla_\alpha\Phi) = true \Leftrightarrow \forall j(+(\iota,\Im_\iota(\alpha),j) \;\&\; j{:}T \Rightarrow \Im_\iota(\Phi) = true)$$
$$(\Im_\iota(\Delta_\alpha\Phi) = true \Leftrightarrow \exists j(+(\iota,\Im_\iota(\alpha),j) \;\&\; j{:}T \;\&\; \Im_\iota(\Phi) = true))$$
$$\Im_\iota(\nabla^A\Phi) = true \Leftrightarrow \iota : \Im_\iota(A) \Rightarrow \Im_\iota(\Phi) = true$$
$$(\Im_\iota(\Delta^A\Phi) = true \Leftrightarrow \iota : \Im_\iota(A) \;\&\; \Im_\iota(\Phi) = true)$$
$$\Im_\iota(\Box\Phi) = true \Leftrightarrow \forall j(\iota\rightarrow j \Rightarrow \Im_\iota(\Phi) = true)$$
$$(\Im_\iota(\Diamond\Phi) = true \Leftrightarrow \exists j(\iota\rightarrow j \;\&\; \Im_\iota(\Phi) = true))$$

where is the current instant of the universe.
The language provides an axiomatic system to define both the temporal universe properties and the correspondence rules for first order formulae between temporal domains. The former set of axioms was defined in [Ciapessoni,92] and its soundness checked in [Corsetti,93]. The latter set includes several axioms stating the chosen correspondences between temporal domains. Among them, let us show the formalization of weak downward projection. Weak downward projection is defined by the following axiom schema:

$$\forall A,B(A \angle B \supset \nabla^A (\Phi \supset \Diamond\Delta^B \Phi))$$

Such a schema states that for each couple of temporal domains related by disjointedness relation, if a formula o holds over an instant of the coarser one, then it holds in at least an instant of the finer one. According to such a formula the theorem stating the upward projection[1] rule can be deduced.

4. The object-oriented extension

4.1. The language: ontology and syntax

The object oriented framework allows to partition the specification by means of *objects,* and to link them by *relations.* An object represents a stereotype likewise a frame [Minsky,75], and it holds the minimal part of a specification. With respect to a methodological point of view an object can identify a *class* or an *instance,* and from an ontological point of view an *entity* or an *event* ([Borgida,85]). According to the methodology an object-class denotes a collection of related individual concepts, while an object-instance denotes an individual concept (e.g., in the block world, the class of blocks and the block#l, respectively). According to the ontology an object-entity denotes an object that persists in time, while an object-event denotes the instantaneous change occurred in entities (e.g., in the block world the entity block and the event of block-change-position). The set of relations among objects is partitioned into *relations among instances and classes, relations among classes,* and *relations among instances.* Relations among instances and classes and relations among classes are the usual relations of belongings and (monotonic) inclusion, respectively [Brachman,85], and both are identified by IS-A. Relations among instances provide the definition of an object-class: methodologically identify the relevant characteristics and logically identify the object *signature.* In a very simplified view the object-class signature may be thought as the specification of the Cartesian product for each relation declared within the object, providing *class identifiers* and *class cardinalities* (i.e., the number of class instances involved into relation extension). Object-instances comprise ground-instance relations declared within the classes they belong to. Entity and event signatures mainly differ for the *at-time* relation that must be declared within each event. Such a relation specifies the occurring instant and the granularity of the event.
The subset of relations declared in an object and changing in time defines *object-state* (time dependent relations).

As an example the specification of the sluice-gate entity and the open event; entity relations are the time independent *area*, and time dependent *open*, *closed* and *moving*; and event relations are *actor* (the controller), *object* (the sluice gate), and *at-time* (the minute):

entity-class S : sluice-gate
time-independent
total[2] area : sluice-gate, square-meters;
time-dependent
total open : sluice-gate;
closed : sluice-gate;
moving : sluice-gate;
end-entity-class

event-class open-sluice-gate
total[2]
actor : open-sluice-gate, C : controller;
object : open-sluice-gate, S : sluice-gate;
at-time : minute;
end-event-class

Relations declared within objects must satisfy integrity-constraints, if any. Integrity-constraints state general and atemporal conditions about object relations, and are expressed by first order formulae.
The complete event-class specification needs the description/prescription of the *conditions* referring to the state of the entities before and after the event occurrence. Conditions state causal and temporal relationships between the event occurrence and the entity-states. Causal relationships codify constraints over the states that validate *(sufficient-conditions)* and are validated *(necessary-conditions)* by the event occurrence. Temporal relationships allow to refer to intervals whose starting, or ending, point is the event occurrence instant. Conditions are expressed by means of logical formulae. Differently from other models of change based on the notion of event, e.g. the Event Calculus [Kowalski,86], the conditions encompass both the deduction of relations from events and the deduction of events from relations.

The sluice gate state is regulated by an entity integrity constraints; the necessary and sufficient conditions associated with the event, state the sluice gate is closed before and open after the event occurrence:

entity-class S : sluice-gate
(...)
integrity-constraints
$\neg$moving(S) $\rightarrow$ (open(S) $\leftrightarrow$ $\neg$closed(S))
moving(S) $\leftrightarrow$ $\neg$(open(S) $\vee$ closed(S))
end-entity-class

event-class open-sluice-gate
(...)
conditions
necessary-sufficient
$\exists t \forall t'(0 \le t' \le t \supset \nabla^{-t^{minute}}$closed(S))
$\exists t \forall t'(0 \le t' \le t \supset \nabla^{t^{minute}}$open(S))
end-event-class

Objects can be decomposed by means of the *part-of* relation. The specification of such a relation is provided by means of the list of classes belonging to the decomposition (i.e., the *decomposition signature*), and the constraints stated among components (i.e., the *decomposition*

[1] The following formula states the upward projection rule:

$$\forall A,B(A \angle B \supset \nabla^A (\Box\nabla^B \Phi \supset \Phi))$$

for each couple of disjontdeness temporal domains if <o> holds over an interval on the finer temporal domain, then it holds over the corresponding simplification on the coarser one.

[2] Such a keyword denotes a relation defined for each class-instance.

specification). Let us pay a special attention to the event decomposition, because of the notion of time granularity is needed. The decomposition of an event consists of a set of events, and a number of constraints stating the causal and temporal relations among component occurring time. We call the event decomposition the *process* associated to the compound event, that is expressed by means of a logical granular formula. A process is the description, with respect to a simplification, of an event occurring time.

The open event decomposition provides a process whose events are *start* and *finish* opening, the latter happens 20 seconds after the former occurrence:

event-class open-sluice-gate
(...)
process
component {O : start-opening, F : finish-opening)
specification
$\exists t\,\{0 \le t \le 10 \wedge \nabla_t^{\text{second}}(\text{start-opening}(O) \wedge \nabla_{20}\,\text{finish-opening}(F))$
end-event-class

Referring to the notions of system specification and history, given in the previous section, we can say that a *system specification* can be expressed by a set of classes, and a *system history* can be expressed by a set of instances.

4.2. The object oriented semantics

The semantics of the object oriented language is provided by means of translation into the logical language, and such a translation provides the object oriented executability, too. Translation put into correspondence each object oriented structure with a formula so that: each object-oriented-specification is translated into a specification-formula, and each object-oriented-history is translated into a history-formula. Object oriented executability is ensured by the logical executability, thus the consistency, or validity, proof of a formula can be applied to a translation, and the semantical result can be extended to the corresponding object. In this section we only describe the object class translation, indeed instance translations is just a conjunction of ground atoms. The description of the translation will be detailed with respect to those aspects involving temporal representation, sketching the other ones. The whole semantics definition of the object oriented language can be found in [Ciapessoni,93].

The following object oriented translation provides an object identifier O, which is a unary predicate that in case of an entity does not change its value in time; *"grain"* is the temporal domain of the event object.

(1) object identifier and signature

let be Σ_i the formula translating the object signature referring to class identifiers, object identifier and object signature are composed according to the instantiation rule ([Hayes,79]):

$$\forall \iota\,(\Phi(\iota) \supset \Sigma_i(\iota))$$

such a rule is specialized in case the object is an event as follows:

(a) the event happens over the temporal domain "*grain*":

$$\forall \iota\ \nabla^{\text{grain}}(\Phi(\iota) \supset \Sigma_i(\iota))$$

(b) events happens almost once over a temporal domain:

$$\forall \iota, g\ \nabla^{g}(\Phi(\iota) \supset \forall t\,(t \neq 0 \supset \nabla_t \neg\Phi(\iota)))$$

(c) events cannot happen over finer temporal domains (*event downward projection*):

$$\forall \iota, g\,(\text{grain} \{ g \supset \nabla^{g} \neg\Phi(\iota)))$$

(d) events can happen over simplifications including the event occurrence instant over "*grain*" (*event upward projection*):

$$\forall \iota, g\,(g \{ \text{grain} \supset \Delta^{g}\,(\Phi(\iota) \supset \Diamond\Delta^{\text{grain}}\,\Phi(\iota)))$$

(2) IS-A relation

the relation of object inclusion is monotonically stated according to the logical implication; let be Φ_1 and Φ_2 two objects such that Φ_1 *is-a* Φ_2, formally, in case Φ_1 and Φ_2 identify entities or events, respectively:

$$\forall \iota\,(\Phi_1(\iota) \supset \Phi_2(\iota))$$
$$\forall \iota\ \nabla^{\text{grain}}(\Phi_1(\iota) \supset \Phi_2(\iota))$$

(3) integrity constraints

let be Ψ the translation of an object integrity-constraints, such a translation is composed with the object signature Σ_n, referring to class cardinalities. In such a way integrity constraints verify object signature consistency [Kowalski, 90]. The compositions of entity-integrity constraints (contestualized over each instant) and event-integrity constraints (contestualized over grain) are, respectively:

$$\forall \iota\,, g\ \nabla^{g}(\Phi(\iota) \supset (\Sigma_n(\iota) \supset \Psi(\iota)))$$
$$\forall \iota\ \nabla^{\text{grain}}(\Phi(\iota) \supset (\Sigma_n(\iota) \supset \Psi(\iota)))$$

(4) conditions (events)

conditions are contestualized over the temporal domain the event occurs, and their formulae are combined with the event identifier in order to cope the causal relation they identify;

(a) consider first necessary conditions, be χ_n the translation of necessary conditions and χ_{ns} the translation of necessary and sufficient condition, then formally:

$$\forall \iota\ \nabla^{\text{grain}}(\Phi(\iota) \supset (\chi_n \wedge \chi_{ns}))$$

(b) consider sufficient conditions, be χ_s the translation of sufficient conditions, then formally:

$$\forall \iota\ \nabla^{\text{grain}}((\chi_s \vee \chi_{ns}) \supset \Phi(\iota))$$

(5) object decomposition

consider Σ_d the translation of decomposition signature, and Ψ_d the translation of decomposition specification:

(a) in case the object identifies an entity the combination of the formulae must be asserted over each instant of the temporal universe:

$$\forall \iota, g\,(\Phi(\iota) \supset \nabla^{g}(\Sigma_d(\iota) \supset \Psi_d(\iota)))$$

(b) in case the object identifies an event the combination of the formulae differs a little from the previous one: (i) we consider the process occurrence equivalent to the event occurrence, that is, if the event decomposition happens

enforcing decomposition constraints then the occurrence of the compound event can be deduced (event decomposition upward projection rules), and vice versa (downward projection rule); (ii) the translation of the decomposition holds over first instant of any articulation of the event occurrence instant, formally:

$$\forall \iota \nabla^{grain}(\Phi(\iota) \equiv \exists g(grain \angle g \wedge \Diamond \Delta^{g}((\Sigma_d(\iota) \supset \Psi_d(\iota)) \wedge \nabla_{-1} \Box \nabla^{grain}{}_{1} \Phi(\iota)))$$

(iii) each component event happens with respect to an instant belonging to an articulation of the event occurrence instant, be $\{e_1,...,e_k\}$, the set of component events, then for each e_i holds:

$$\forall \iota \nabla^{grain}(\Phi(\iota) \supset \exists g(grain \{ g \wedge \Diamond \Delta^{g} e_i))$$

(6) temporal correspondence of entity states

entity states can change values in time: such a change should be always regulated by an event occurrence to get a complete specification; thus, the extension of a relation held in the state is fixed with respect to the temporal domain the event happens, and it is necessary to put such extension into correspondence with the other (finer, coarser) temporal domains of the universe. To state such a correspondence we adopt the *downward* (weak) and *upward* (strong) projection rules (defined in section 2), such that in the former if a literal (i.e., the atomic formula, asserted or negated, translating the relation), holds over an instant, then it holds over an instant of any articulation; the latter one states that if a literal holds throughout an interval, then it holds over any interval simplifications. Let be $R = \{rel_1,..., rel_k\} \cup \{rel_{n1},...,rel_{nk}\}$ the set of all entity relations asserted and negated, respectively, held in the entities states, then for each $r_i \in R$ holds:

downward projection

$$\forall g_1, g_2 (g_1 \{ g_2 \supset \nabla^{g1} (r_i \supset \Diamond \Delta^{g2} r_i))$$

upward projection

$$\forall g_1, g_2 (g_1 \{ g_2 \supset \nabla^{g1} (\Box \nabla^{g2} r_i \supset r_i))$$

Conclusions

The paper has defined the semantics of the temporal aspects of an object oriented language that supports the notion of time granularity. Such a semantics is expressed by means of object translation into a logical formula. Thus, the object oriented language gets a formal semantics and gains logical executability. Further, in the translation we proposed a number of projection rules, between temporal domains, grounded on the object oriented ontology. Such projection rules could be embedded within the logical language to improve its expressiveness.

Acknowledgments

We would like to thank Angelo Montanari for the contributions he gave to the argument during his working period at CISE. Further, we are grateful to Paolo Mancarella and Dino Pedreschi for helpful comments on an early draft of this paper.

References

[Allen,83] Maintaining Knowledge about Temporal Intervals, J.Allen, Communications of the ACM, 26,11,1983.

fBorgida,85] Knowledge Representation as the Basis for Requirements Specifications, Borgida A., Greenspan S., Mylopoulos J., IEEE Computer, vol 18(4), April 1985.

[Brachman,85] Readings in Knowledge Representation, R.Brachman, H.Levesquc, Morgan Kaufmann, San Matteo CA, 1984.

[Ciapessoni,92] Embedding Time Granularity in a Logical Specification Language for Synchronous Real-Time Systems, E.Ciapessoni, E.Corsetti, A.Monunan, P.San Pietro, Science of Computer Programming Magazine, Elsevier: North-Holland, 1992, (in printing).

[Ciapessoni,93] TRW* semantics by translation into granular TRIO, E.Ciapessoni, E.Corsetti, P.Mancarella, M.Migliorati, D.Pedreschi, CISE Technical Report, 7657, March 1993, (in Italian).

[Corsetti,91a] Dealing with Different Time Granularities in Formal Specification of Real-Time systems, Corsetti E., Montanari A., Ratto E., The journal of Real-Time Systems, Vol III, Issue 2,1991.

(Corsetti,9lb] Dealing with Different Time Scales in Formal Specifications, Corsetti E., Crivelli E., Mandrioli D., Montanari A., Morzenti A., Ratto E., San Pietro P., Proc. 6th International Workshop on Software Specification and Design, Como, Italy, October 1991.

|Corsetti,93) Logical Specification of Real-Time Granular Systems in the Synchronous Case, E.Corsetti E. Ciapessoni, CISE - Technical Report, n.7217, 1993.

[Dean,88J Reasoning about Partially Ordered Events, T.Dean, M.Boddy, Artificial Intelligence, 36,1988.

[Ghezzi 90] TRIO a Logic Language for Executable specifications of Real-Time Systems, Ghezzi C, Mandrioli D., Morzenti A., The Journal of Systems and Software, June-July 1990.

IGreer,89] A Computational Framework for Granularity and its Application to Educational Diagnosis, J.Greer, G.McCalla, in Proc. 11th IJCAI, Detroit, USA, 1989.

[Hayes 79] The Logic of Frames, Hayes P., in Frame Conception and Text Understanding, Metzing D., (Ed), Walter de Grutyer and Co., Berlin, 1979.

[Hobbs,85] Granularity, J.Hobbs, in Proc. 9th IJCAI, Los Angeles (USA), 1985.

|Kowalski,86) A Logical Based Calculus of Events, R.Kowalski, M.Sergol, New Generation Computing, 4,1986.

[Kowalski,90] Problems and Promises of Computational Logic, R.Kowalski, in Computational Logic, Brussels, November, 1990.

[Koymans 90] Specifying Real-Time Properties with Metric Temporal Logic, R.Koymans, The Journal of Realtime Systems, vol. II, 1990.

[Levesque,86] Making Belivers out of Computers, H.Levesque, Artifical Intelligence, 30, 1986.

[Montanari ,92] Dealing with Time Granularity in the Event Calculus, A.Montanari, E.Maimm, E.Ciapessoni, E.Ratto, Proc. Int. Conf. on 5th Generation Computer Systems, Tokio, Japan, 1992.

[McCarthy,69] Some Philosophical Problem from the Standpoint of Artificial Intelligence, J.McCarthy, P.Hayes, Machine Intelligence 4, 1969.

[Minsky,75] A Framework for Representing Knowledge, M.Minsky, in The Psychology of Computer Vision, P.Winston ed., McGraw Hill, 1975.

[Pnueli,8l] The Temporal Semanticss ofCuncurrent Programs, Pucli A., Theoretcal Computer Science, 13, North-Holland, 1981.

[Rescher,71] Temporal Logic, Rescher N., Urquhart A., Springer-Verlag, Wien-New-York, 1971.

[Shoham,88] Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence, Shoham Y., MIT Press, 1988.