An intensional approach to qualitative and quantitative periodicity-dependent temporal constraints

Luca Anselma°, Stefania Montani*, Paolo Terenziani*
°DI, Università di Torino, Corso Svizzera 185, Torino, Italy
anselma@di.unito.it
*DI, Univ. Piemonte Orientale "A. Avogadro", Via Bellini 25/g, Alessandria, Italy
{stefania, terenz}@mfn.unipmn.it

Keywords: Temporal Reasoning. User-defined periodicities. Periodicity-dependent temporal constraints. Temporal constraint propagation.

Abstract

In this paper, we propose a framework for representing and reasoning about qualitative and quantitative temporal constraints about periodic events. In particular our contribution is two-folded: (i) we provide a formalism to deal with both qualitative and quantitative "periodicity-dependent" constraints between repeated events, considering also user-defined periodicities; (ii) we propose an intensional approach to temporal reasoning, which is based on the operations of intersection and composition. Such a comprehensive approach – to the best of our knowledge – represents an innovative contribution, which integrates and extends results from both the Artificial Intelligence and the Temporal Databases literature.

1 Introduction

Temporal representation and temporal reasoning are fundamental issues within the Artificial Intelligence (henceforth, AI) communities, since they play a central role in many "intelligent" activities. Starting from the early 80's, many Artificial Intelligence researchers took interest into the development of specialised approaches to deal with different kinds of temporal constraints [Vila, 94]. In particular, a relevant amount of work has been devoted to the treatment of (constraints between) periodic events (consider, e.g., [Ladkin, 86; Ligozat, 91; Morris et al., 96; Anselma, 04]). Some of such approaches tackled the problem of dealing with “periodicity-based” temporal constraints, in which constraints between periodic events depend on the specific periodicity in which events occur. For instance, in [Loganantharaj & Giambrone, 95], the authors dealt with "periodicity-dependent" durations (e.g., *During the early morning (5am-7am), going to work by car takes from 20 to 30 minutes; During morning rush hours (7am-9am), going to work by car takes from 30 to 40 minutes*), while in [Terenziani, 97] "periodicity-based" qualitative (e.g., “*before*” [Allen, 83]) constraints were considered.

At the same time, mostly in the area of Temporal Databases, many authors have realized that, since the use of a periodicity or calendar depends on the cultural, legal, and even business orientation of the users (see e.g., [Soo & Snodgrass, 93]), dealing with *user-defined* periodicities is an essential task. As a consequence, several approaches (consider, e.g., [Leban, 86; Niezette & Stevenne, 92; Bettini & De Sibi, 99; Ning et al., 02, Terenziani, 03] and the survey in [Tuzhilin & Clifford, 95]) have been devised to model user-defined periodicities (and the related notions of *granularities* and *calendars*).

The goal of this paper is to propose an original and comprehensive framework dealing with both *quantitative* and *qualitative* “periodicity-based” temporal constraints between periodic events, taking also into account *user-defined* periodicities.

Once a set of temporal constraints has been defined, temporal reasoning is needed, in order to support the intended semantics they represent. As a matter of fact, without temporal reasoning, a user can represent any set of constraints - even an inconsistent one - with no reaction by the system. Given a Knowledge Base (henceforth, KB) of periodicity-based temporal constraints, we thus aim at providing a constraint-propagation-based form of temporal reasoning to amalgamate them and to check their consistency. As most approaches in AI, our temporal reasoning approach is based on the operations of intersection and composition. Unfortunately, usual approaches in the literature (such as, e.g., Allen [Allen, 83] and Morris et al. [Morris et al., 96]), although synthetically defined and semantically clear, do not apply to our case, since we have to manage two different components (periodicity and temporal constraint) and the interactions between them. One possibility would be that of "exploding" each periodicity over a time span of interest (or, in the most general case, on the "least common multiple" of the period of each periodicity in the KB), explicitly generating all the repetitions (instances) of the events over such a time span, imposing all the input temporal constraints on the generated instances, and then using a "standard" constraint-propagation technique to perform temporal reasoning on them. Such an ***extensional*** approach may indeed work. However, it has two main drawbacks:

(i) in most cases, too many instances of events need to be taken into account (since, e.g., the least common multiple of "week" and "month" is 28-year long);

(ii) in any case, the output of constraint propagation would not be user-friendly and perspicuous (e.g., no train schedule lists explicitly the departure-arrival times of trains in all the days of a year; see e.g. the discussion in [Terenziani, 03]).

Therefore, we have worked at the definition of an ***intensional*** calculus, which, although not complete, performs part of the inferences providing a *correct* perspicuous (i.e., intensional) output to users.

The paper is organized as follows. In section 2 we formalize our representation language for periodicity-based temporal constraints. In section 3, we describe the intensional calculus we have defined for reasoning on them. Section 4 is devoted to comparisons and conclusions.

2 Representing periodicity-based constraints

As stated in the introduction, our formalism aims at representing (i) *qualitative* and (ii) *quantitative* periodicity-based temporal constraints, considering also (iii) *user-defined* periodicities.

As a motivating example, consider the following real-world situation.

Example 1.

1. Every Monday, Sam takes 10 to 30 minutes to read his email; no later than 1 hour after reading the email, he has a meeting, which lasts from 1 to 2 hours;
2. during the meeting, he makes a presentation;
3. every Monday afternoon, from 1 hour to 4 hours after the meeting, Sam reports the meet-

ing results to his boss (and this activity takes from 1 hour to 1 hour and a half);

4. on working days, (and later than 30 minutes after having seen Sam, on Monday), Sam's boss meets the head of the department for 1-2 hours;
5. on working days (and later than 2 hours after the meeting, on Monday), Sam writes a report about his work (this activity takes from 30 minutes to 1 hour);
6. on working days, no later than 2 hours after having written the report, Sam talks to the head of the department for 1-2 hours;
7. on Friday and Saturday, no later than 1 hour after (possibly) having had a meeting (which takes up to 1 hour), Mary writes a report about her work, and this activity requires from 30 minutes to 1 hour of time;
8. on working days, no later than 5 hours after having written the report, Mary talks to the head of the department for 1-2 hours.

Since the main goal of our work is that of devising a comprehensive approach covering issues (i)-(iii) above (and the interactions between them), in the current version we have chosen to rely as much as possible on contributions described in literature, both as concern the formalism to define periodicities and the formalism coping with temporal constraints. In both cases, we have selected widely known and used AI approaches: the *collection* formalism [Leban et al., 86], recalled in section 2.1, and *STP* (Simple Temporal Problem [Dechter et al., 91]), recalled in section 2.2.

2.1 The collection Formalism

The collection formalism allows one to represent user-defined periodicities in a symbolic, user-friendly and incremental way. Leban et al. take *calendars* (e.g., seconds, hours, days, years) as basic elements, and provide two classes of operators: *dicing* and *slicing*. The *dicing* operators provide a means to further divide each interval in a periodicity according to another periodicity. For example,

Days:during:Weeks

breaks up weeks into days. *Slicing* operators provide a way of selecting intervals from collections. For instance, *Monday = [1] \ Days:during:Weeks*

selects the first day in each week (e.g. Monday in the European calendar), while

Monday&Tuesday = *[1,2] \ Days:during:Weeks*

selects the pair of Mondays and Tuesdays and

WorkingDays = [1,2,3,4,5] \ Days:during:Weeks

selects all working days (see the periodicities in Example 1). Collection expressions can be arbitrarily built by using a combination of these operators. For example

MondayAfternoon = [14,15,16,17] \ Hours:during:Monday

selects the afternoon hours of Monday.

The semantics of the expressions is informally represented in [Leban et al., 86] as nested sets of convex time intervals. However, as in [Bettini & DeSibi, 99], we do not consider the possibility of nesting sets of intervals, in fact, we choose to rely on a simpler notion of temporal extent of a periodicity.

Definition: Extension function. We define the extension (Ext) of a periodicity P the set of time intervals which constitute the instances of P on the timeline, expressed using the basic granularities. The function Ext takes in input a periodicity (e.g., Monday) and gives in output a set of time intervals (e.g., supposing that the basic granularity is days and that day 0 is a Monday, Ext(Monday)={…[-7,-7],[0,0],[7,7], …}). □

2.2 STP

The STP framework, on the other hand, is based on the notion of *bound on difference* constraints. A bound on difference is a linear inequality of the form $d_1 \leq X\text{-}Y \leq d_2$ (also strict inequalities are allowed). The variables in a bound on difference may correspond to time points (and extremes of time intervals). Thus, a bound on difference $d_1 \leq X\text{-}Y \leq d_2$ has the following temporal interpretation: the temporal distance between the time points (endpoints of time intervals in our case) *X* and *Y* is included into the *admissibility range* $[d_1, d_2]$. A temporal knowledge base (KB) is a *conjunction* of bounds of differences constraints (called *STP constraints* in [Dechter et al., 91] and henceforth). In [Meiri, 91; Brusoni et al., 97] it is shown how different types of qualitative and quantitative temporal constraints can be easily mapped to STP constraints. In particular, all the qualitative constraints of the *Continuous Interval Algebra* (i.e., the subset of relations of Allen's Algebra which can be mapped onto conjunctions of constraints between points, excluding inequality [Vila, 94]) can be mapped onto STP constraints, as well as precise and imprecise dates, durations and delays. Given an STP, the strictest admissibility range between each pair of time points (i.e., the *minimal network*) can be obtained using a standard all-pairs shortest paths algorithm such as, e.g., Floyd-Warshall's one, which operates in a time cubic in the number of points, is correct and complete for STP constraints, and can be trivially adapted to check consistency [Dechter et al., 91].

For instance, the temporal constraint between the end of the email activity and the end of the meeting Sam activity in constraint 1 of Example 1 would be represented as follows: *0 < Start(meet_S)-End(email) < 1h*.

2.3 Our representation formalism

In our approach, a (binary) periodicity-based constraint (PBC) is modelled by a triple *<Ev, Per, Con>* where *Ev* denotes a pair of events, *Per* a user-defined periodicity, and *Con* the temporal constraints[1].

The (intuitive) semantics of periodicity-based constraints *<<ev_1,ev_2>, P, C>* in our approach is the following.

ev_1 and *ev_2* are two periodic events. For *each* occurrence (*instance*) p_i of the periodicity *P*,

(i) there is exactly an instance $e1_i$ of ev_1 taking place in p_i and only in p_i[2]*;*

(ii) there is exactly an instance $e2_i$ of ev_2 taking place in p_i and only in $p_{i;}$

(iii) *C* ($e1_i$, $e2_i$) holds.

We allow users to associate names with expressions in Leban et al.'s formalism, and only user-defined names can be used in the ***Per*** component of the constraints.

[1] Durations are a degenerate case, in which the Ev component only consists of one event, and the Con component of the distance between its endpoints.

[2] Note that we admit the existence of events of the same type in different periodicities (e.g. writing a report, see constraints 5 and 7 in Example 1); in this situation, we use indexes (S for Sam and M for Mary in the example) in order to distinguish between them).

In our approach, the ***Con*** component of each periodicity-based constraint can thus be expressed by a quadruple of admissibility ranges, representing the minimal and maximal distance between the starting/ending points of ev_1 and ev_2, i.e. the distances between the points <Start(ev_1) and Start(ev_2), Start(ev_1) and End(ev_2), End(ev_1) and Start(ev_2), End(ev_1) and End(ev_2)>.

Finally, the events (***Ev*** component) can be simply specified using identifiers.

Note that we use explicit indexes (e.g., S for Sam and M for Mary in Example 1) to differentiate between events of the same type, taking place at different times.

Considering the notation introduced above, the constraints in Example 1 can be expressed as follows and graphically depicted in Figure 1:

Example 1'.

1. <<email, meet_S>, Monday, <(10m, 1h30m), (1h10m,3h30m), (0,1h), (1h,3h)>>;
2. <<presentation, meet_S>>, Monday, <(-inf, 0), (-inf,+inf), (-inf,+inf), (0,+inf)>>;
3. <<meet_S, boss>, MondayAfternoons, <(2h,6h), (3h,7h30m), (1h,4h), (2h,5h30m)>>;
4. <<boss, head>, WorkingDays, <(1h,2h), (2h,4h), (0,30m), (1h,2h30m)>>;
5. <<meet_S, rpt_S>, WorkingDays, <(1h,4h), (1h30m,5h), (0,2h), (30m,3h)>>;
6. <<rpt_S, head>, WorkingDays, <(30m,3h), (1h30m,5h), (0,2h), (1h,4h)>>;
7. <<meet_M,rpt_M>, Monday&Tuesday, <(0,2h), (30m,3h), (0,1h), (1h,2h)>>;
8. <<rpt_M, head>, WorkingDays, <(30m,6h), (1h30m,8h), (0,5h), (1h,7h)>>.

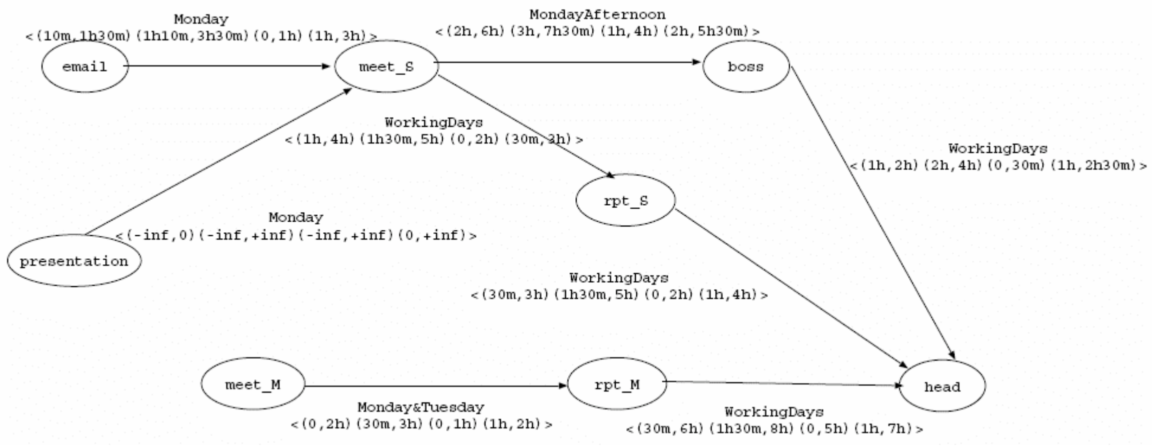
Monday
<(10m,1h30m)(1h10m,3h30m)(0,1h)(1h,3h)>
MondayAfternoon
<(2h,6h)(3h,7h30m)(1h,4h)(2h,5h30m)>
email
meet_S
boss
WorkingDays
<(1h,4h)(1h30m,5h)(0,2h)(30m,3h)>
WorkingDays
<(1h,2h)(2h,4h)(0,30m)(1h,2h30m)>
rpt_S
Monday
<(-inf,0)(-inf,+inf)(-inf,+inf)(0,+inf)>
presentation
WorkingDays
<(30m,3h)(1h30m,5h)(0,2h)(1h,4h)>
meet_M
rpt_M
head
Monday&Tuesday
<(0,2h)(30m,3h)(0,1h)(1h,2h)>
WorkingDays
<(30m,6h)(1h30m,8h)(0,5h)(1h,7h)>

Figure 1: the periodicity-based temporal constraints among the activities in Example 1. S and M stand for Sam and Mary respectively, and are used as indexes to differentiate the same type of action, when it takes place at different times.

3 Intensional calculus

Given the drawbacks of an extensional approach to temporal reasoning presented in the introduction, we concentrated our efforts on the definition of an intensional calculus, able to perform part of the inferences and to generate a *correct* perspicuous (i.e., intensional) output to users. The details of our contribution and the related properties are illustrated in the following sections.

3.1 Intensional calculus: main issues

Our basic idea is that of defining a modular intensional calculus, in which intersection (∩) and composition (@) are separately computed on the periodicities and on the constraints, i.e.,

$<<ev_1, ev_2>, P_1, C_1> \cap <<ev_1, ev_2>, P_2, C_2> = <<ev_1, ev_2>, P_1 \cap^P P_2, C_1 \cap^C C_2>$

$<<ev_1, ev_2>, P_1, C_1> @ <<ev_2, ev_3>, P_2, C_2> = <<ev_1, ev_3>, P_1 @^P P_2, C_1 @^C C_2>$

The definition of the operators of intersection and composition between STP constraints (i.e., the *Con* component of a periodicity-based constraint) are the standard ones in Floyd-Warshall's algorithm, i.e.,

$C_1 \cap^C C_2 = C_1 \cap C_2$, where $\cap$ is the standard intersection between time intervals (admissibility ranges);

$C_1 @^C C_2 = K$, where $K = (a+c, b+d)$ for $C_1=(a,b)$ and $C_2=(c,d)$.

On the other hand, several problems have to be faced in order to provide an intensional definition of intersection and composition between user-defined periodicities (i.e., the *Per* component of a periodicity-based constraint). Specifically, operating at the intensional level means taking into account only one "prototypical" repetition, i.e., the repetition in a *"typical" common period* (i.e., $P_1 \cap^P P_2$ or $P_1 @^P P_2$), *represented in an intensional way*. This means that:

(1) a one-to-one correspondence between the instances of P_1 and P_2 must be pointed out (see comments below);

(2) the formalism for representing periodicities must be extended to be able to intensionally express the common period;

(3) an algorithm must be devised to compute the common period on the basis of two periodicities P_1 and P_2. Notice that the computation must operate at the intensional level, and provide an intensional output.

3.2 One-to-one correspondences between periodicities

Given the points (i) and (ii) in the definition of the semantics of our periodicity-based constraints (PBC) (see section 2.3), two constraints $<<ev_1, ev_2>, P_1, C_1>$ and $<<ev_1, ev_2>, P_2, C_2>$ referring to the *same* repeated events ev_1 and ev_2 are consistent only in case the periodicities P_1 and P_2 are such that:

(1) there is a 1 to 1 correspondence between instances of P_1 and instances of P_2, and

(2) the corresponding instances intersect in time.

For instance, the periodicities Tuesday&Wednesday and Wednesday&Thursday (with the intuitive meaning) satisfy the above constraints (so that one can consistently assert that ev_1 occurs exactly once in each Tuesday&Wednesday and exactly once in each Wednesday&Thursday, implying that ev_1 occurs exactly once in the intersection). On the other hand, Tuesday&Wednesday and Days do not satisfy constraint (1) (since Days are more "frequent" than Tuesday&Wednesday) and Tuesday&Wednesday and Thursday&Friday do not satisfy (2).

Unfortunately, since periodicities are intrinsically cyclic, more than one correspondence (specifically, two) satisfying the above constraints (1) and (2) may hold between pairs of periodicities. For instance, the periodicities P_1=Tuesday&Wednesday&Thursday&Friday and P_2=Friday&Saturday&Sunday&Monday&Tuesday "circularly" intersect twice so that two one-to-one correspondences are possible (one pairing each instance of P_1 with the instance of P_2 starting the preceding Friday, and one pairing it with the following instance).

To disambiguate, we first introduce the notion of Reference Time point (RT), as a point working as an absolute reference for all the periodicity-based constraints in a KB. Any disambiguating rule could work. In the following, we just propose one possibility, as a working example.

Anchoring Rule. *Given two periodicities P_1 and P_2, take as the "anchoring pair" the pair consisting of the first instance of P_1 and P_2 which start after RT; the following pairs of the correspondence are obtained by pairing the second, third, etc. instances of P_1 and P_2 (and analogously for the preceding pairs).* □

Thus, one-to-one correspondences between periodicities can be unambiguously defined as follows:

Definition: We call "**anchoring relation**" between two periodicities P_1 and P_2 (denoted by $\boldsymbol{A_{RT}(Ext(P_1),Ext(P_2))}$), the one-to-one correspondence (if any) obtained by pairing the intersecting instances of P_1 and P_2, and applying the anchoring rule in case of ambiguity. We state that there is a **one-to-one correspondence** $\mathbf{Cor}^{A_{RT}}(\mathbf{P_1},\mathbf{P_2})$ if and only if there is an anchoring relation $\boldsymbol{A_{RT}}$ between $Ext(P_1)$ and $Ext(P_2)$. □

Property 1. The relation A_{RT} is reflexive, symmetric and transitive. □

3.3 Extending the symbolic formalism for user-defined periodicities

In order to define an algebraic approach (i.e., an approach in which the results of intersection and composition are still constraints in our formalism), it is useful to further extend the formalism to represent user-defined periodicities with two operators: pairwise intersection ($\underline{\cap}^{P}$) and restricted pairwise union ($\underline{\cup}^{P}$) between periodicities. Since they are used to represent in an intensional way “typical” common periods, the operators must take in input periodicities whose instances are in a one-to-one correspondence, and operate on each pair.

Definition (pairwise intersection). Given two periodicities P_1 and P_2 such that $\mathbf{Cor^{A}RT(P_1,P_2)}$ holds, the pairwise intersection between P_1 and P_2, i.e. $P_1 \underline{\cap}^{P} P_2$, is defined using the Ext function as follows (where $p1_h \cap p2_k$ denotes standard intersection between two time intervals –conceived as sets of time points):

$Ext(P_1 \underline{\cap}^{P} P_2)=\{(p1_h \cap p2_k) \backslash <p1_h,p2_k> \in A_{RT}(Ext(P_1),Ext(P_2))\}$. □

Property 2. If $P=(P_1 \cap^{P} P_2)$, then $Cor^{A}RT(P,P_1)$ and $Cor^{A}RT(P,P_2)$ hold. □

Definition (pairwise restricted union). Given two periodicities P_1 and P_2 such that $\mathbf{Cor^{A}RT(P_1,P_2)}$ holds, and such that $\forall <p1_h,p2_k> \in A_{RT}(Ext(P_1),Ext(P_2))$ $p1_h$ and $p2_k$ intersect or meet in time, then the pairwise restricted union between P_1 and P_2, i.e. $P_1 \underline{\cup}^{P} P_2$, is defined as follows (where $p1_h \cup p2_k$ denotes standard union between two time intervals –conceived as sets of time points):

$Ext(P_1 \underline{\cup}^{P} P_2)=\{(p1_h \cup p2_k) \backslash <p1_h,p2_k> \in A_{RT}(Ext(P_1),Ext(P_2))\}$. □

Notice that, given the second condition in the definition, $\underline{\cup}^{P}$ only generates convex time intervals and the restricted union of non-intersecting or non-meeting intervals is empty[3].

Property 3. If $P=(P_1 \underline{\cup}^{P} P_2)$, then $Cor^{A}RT(P,P_1)$ and $Cor^{A}RT(P,P_2)$ hold. □

In the following, we distinguish between *base periodicities* and *composite periodicities*. With *base periodicities* we will intend Leban’s *calendars* and those periodicities obtained by the ap-

[3] Notice also that we use union in order to intensionally define composition (see section 3.5), and composition gives an inconsistency whenever the periodicities to be composed do not intersect in time.

plication of Leban et al.'s language [Leban et al., 86] only. With the term *composite periodicity* we intend periodicities defined using at least one of the operators (i.e., $\underline{\cap}^P$ and $\underline{\cup}^P$). In the calculus we define hereafter, we work as much as possible with base periodicities, but there are cases where the result of intersection or composition of base periodicities is a composite periodicity (see section 3.4). In section 3.5 we provide rules for simplifying composite periodicities.

3.4 Relations between periodicities

In order to provide "perspicuous" results, the intensional operations of intersection and composition cannot just give in output a string concatenation of input periodicities and operators. At least two types of simplifications can be (and need to be) performed:

(i) **redundancy elimination**; e.g., the output of the intersection of WorkingDays (i.e., days from Monday to Friday) and Monday should be just Monday and not WorkingDays $\underline{\cap}^P$ Monday;

(ii) **empty periodicity detection**; for instance, the output of the intersection of Monday and Wednesday should be empty and not Monday $\underline{\cap}^P$ Wednesday.

In the next section we present an intensional definition that performs these simplifications. Since periodicities are user defined, it is not possible to define a priori all the intersections and compositions between all pairs of periodicities. Therefore, we point out a set of five possible relations between two user-defined periodicities which are exhaustive and mutually exclusive, and we define intersection and composition on these bases.

The relations are $\subseteq^P$, $\supset^P$, i^P, ni^P, $\#^P$ are defined as below (where $p1_h \subseteq p2_k$ and $p1_h \cap p2_k$ respectively denote standard inclusion and intersection between two time intervals):

(1) $\mathbf{P_1 \subseteq^P P_2} \Leftrightarrow Cor^{A_{RT}}(P_1, P_2) \wedge \forall \langle p1_h, p2_k \rangle \in A_{RT}(Ext(P_1), Ext(P_2))\ p1_h \subseteq p2_k$

i.e. the inclusion relation between P_1 and P_2 holds if the one-to-one correspondence holds, and for every corresponding pair of instances in the two periodicities, the first element is included in the second one; as an example, Monday $\subseteq^P$ WorkingDays, and Monday $\subseteq^P$ Monday;

(2) $\mathbf{P_1} \supset^{\mathbf{P}} \mathbf{P_2} \Leftrightarrow \text{NOT } P_1 \subseteq^{P} P_2 \wedge \mathrm{Cor}^{A}\mathrm{RT}(P_1, P_2) \wedge \forall\langle p1_h, p2_k\rangle \in A_{RT}(Ext(P_1), Ext(P_2))$ $p1_h \supseteq p2$

i.e. the strict inclusion relation between P_2 and P_1 holds if the previous relation does not hold and the one-to-one correspondence holds, and for every corresponding pair of instances in the two periodicities, the first element includes the second one; as an example, WorkingDays $\supset^{\mathbf{P}}$ Monday;

(3) $\mathbf{P_1}\ \mathbf{i}^{\mathbf{P}}\ \mathbf{P_2} \Leftrightarrow \text{NOT } P_1 \subseteq^{P} P_2 \wedge \text{NOT } P_1 \supset^{P} P_2 \wedge \mathrm{Cor}^{A}\mathrm{RT}(P_1, P_2) \wedge \forall\langle p1_h,p2_k\rangle \in A_{RT}(Ext(P_1),Ext(P_2))\ p1_h \cap p2_k \neq \emptyset$

i.e. the intersection relation between P_1 and P_2 holds if none of the previous relations holds, if the one-to-one correspondence holds, and for every corresponding pair of instances in the two periodicities, the intersection between the first element and the second one is not empty; as an example, Sunday&Monday $\mathbf{i}^{\mathbf{P}}$ WorkingDays;

(4) $\mathbf{P_1}\ \mathbf{ni}^{\mathbf{P}}\ \mathbf{P_2} \Leftrightarrow \mathrm{Cor}^{A}\mathrm{RT}(P_1, P_2) \wedge \exists\langle p1_h,p2_k\rangle \in A_{RT}(Ext(P_1),Ext(P_2))\ p1_h \cap p2_k = \emptyset$

i.e. the non-intersection relation between P_1 and P_2 holds if the one-to-one correspondence holds, and for every corresponding pair of instances in the two periodicities, the intersection between the first element and the second one is empty; as an example, Monday $\mathbf{ni}^{\mathbf{P}}$ Tuesday;

(5) $\mathbf{P_1}\ \#^{\mathbf{P}}\ \mathbf{P_2} \Leftrightarrow \text{NOT } \mathrm{Cor}^{A}\mathrm{RT}(P_1, P_2)$

i.e. the non-correspondence relation between P_1 and P_2 holds if the one-to-one correspondence does not hold; as an example, Monday $\#^{\mathbf{P}}$ Days.

Given two user-defined *base* periodicities, it is possible to devise different sets of rules in order to compute the relation holding between them on the basis of their definition. For example, from any Leban's definition of the form $P_1 = n / P_2$:during: P_3, we have $P_1 \subseteq^{P} P_3$, $P_1 \#^{P} P_2$, $P_2 \#^{P} P_3$. Unfortunately, this set of rules is not complete; whenever the relation cannot be inferred via the rules, it can be asked to the user (or computed by generating and checking extensions).

3.5 Intensional definition of intersection and composition about basic periodicities

Periodicity intersection and composition are defined as shown in Table 1. Each row in Table 1 must be interpreted as a conditional *simplification* rule of the form: “If Relation Then P_1 OP^P P_2 = formula”, where OP^P is $\cap^P$ or $@^P$.

Periodicity Intersection. Intersection applies to pairs of constraints of the form <<ev_1, ev_2>, P_1, C_1> and <<ev_1, ev_2>, P_2, C_2>. For example, the first row states that if $P_1 \subseteq^P P_2$, then $P_1 \cap^P P_2 = P_1$. For instance, Monday $\cap^P$ WorkingDays = Monday. The last row states that, if there is no one-to-one correspondence between P_1 and P_2, or the corresponding instances do not intersect, then the intersection is empty, so that the result of <<ev_1, ev_2>, P_1, C_1> ∩ <<ev_1, ev_2>, P_2, C_2> is an inconsistency.

Periodicity Composition. Composition applies to pairs of constraints of the form <<ev_1, ev_2>, P_1, C_1> and <<ev_2, ev_3>, P_2, C_2>. Concerning the periodicities P_1 and P_2, notice that the semantics of our constraints imposes that the “common” event *ev_2* must occur both exactly once in each P_1 (and only in P_1) and exactly once in each P_2 (and only in P_2). As Table 1 reports, this is possible in case the relation between P_1 and P_2 is one of $P_1 \supseteq^P P_2$, $P_1 \subseteq^P P_2$ and P_1 i^P P_2, while an inconsistency must be reported in the cases P_1 ni^P P_2 and P_1 $\#^P$ P_2. For instance, Monday $@^P$ Tuesday = ∅ while Monday $@^P$ WorkingDays = WorkingDays.

Property 4. Correctness and simplification completeness. Our definitions of intersection and composition (see Table 1) are correct; moreover, they are complete as regards emptiness detection and redundancy elimination. □

Relation between P_1 and P_2	$P_1 \cap^P P_2$	$P_1 @^P P_2$
$P_1 \subseteq^P P_2$	P_1	P_2
$P_1 \supseteq^P P_2$	P_2	P_1
P_1 i^P P_2	$P_1 \underline{\cap}^P P_2$	$P_1 \underline{\cup}^P P_2$
P_1 ni^P P_2	∅	∅
P_1 $\#^P$ P_2	∅	∅

Table 1. Intersection and composition between periodicities.

Proof (hint): correctness can be proved by showing that all simplifications are correct, i.e., (i) for each rule **IF P_1 R P_2 THEN Expr1 → Expr2** in Table 1, then Ext(Expr1) = Ext(Expr2), and (ii) we output an empty value just in case intersection/composition is not possible, given the semantics of the periodicity-based constraints.

Simplification completeness is proven by showing that, given any two base periodicities P_1 and P_2 and an operation OP^P ($@^P$ or $\cap^P$) between them, (i) whenever Ext(P_1 OP^P P_2) = Ext(P_2), there is a rule such that the intensional expression "P_1 OP^P P_2" is transformed into "P_2" (or vice versa, if Ext(P_1 OP^P P_2)=Ext(P_1)); (ii) whenever the intersection or the composition operations are not possible, given the semantics of the periodicity-based constraints, there is a rule that reports an empty value. □

3.6 Intensional definition of intersection and composition about composite periodicities

In this section, we deal with composite periodicities, and we apply our intensional calculus to them.

In the following, we hypothesize that composite periodicities are put in normal form, as below:

$$(P_{11} \underline{\cup}^P \ldots \underline{\cup}^P P_{1s}) \underline{\cap}^P \ldots \underline{\cap}^P (P_{g1} \underline{\cup}^P \ldots \underline{\cup}^P P_{gf})$$

where each P_{ij} is a basic periodicity. We also hypothesize that all input composite periodicities (if any) have already been simplified (this can be done in a pre-compilation step).

The following property holds.

Property 5. Pairwise restricted union is distributive with regard to pairwise intersection. □

Given these premises and given the definitions in Table 1, temporal reasoning on composite periodicities requires to calculate:

(i) the intersection $\underline{\cap}^P$ of normal forms;

(ii) the restricted union $\underline{\cup}^P$ of normal forms (to compute ∩ and @ respectively: see section 3.2).

Basically, operations (i) and (ii) consist of a set of simplification rules to detect emptiness and to eliminate redundancies. Since periodicities are user-defined, operating directly on composite periodicities is a hardly feasible approach. In particular, it is difficult to find rules to determine which one of the five relations holds between two composite periodicities (analogous to the rule in section 3.5). Therefore, the only practical way of operating is that of devising a compositional and modular calculus, in which operations between composite periodicities are decomposed by considering pairwise the basic periodicities composing them.

Note 1. Unfortunately, since certain simplifications can only be captured considering composite periodicities as a whole, such a compositional approach cannot be simplification-complete (see Property 6).

Intersection of normal forms.

(α) $((P_{11}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{1s})\ \underline{\cap}^P\ldots\underline{\cap}^P\ (P_{g1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{gf}))\ \underline{\cap}^P$

$((P'_{11}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{1t})\ \underline{\cap}^P\ldots\underline{\cap}^P\ (P'_{h1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{hk}))$.

Emptiness detection. If there are two basic periodicities P_{il} and P'_{jk} such that $P_{il}\ \#^P\ P'_{jk}$, then the intersection of the two normal forms is empty. For example, in *(Monday $\underline{\cup}^P$ Tuesday) $\underline{\cap}^P$ Days*, since *Monday $\#^P$ Days*, the result is empty.

If there exist two unions of basic periodicities $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})$ and $(P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ such that $\forall P_{il}$ in $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})$ and $\forall P'_{jm}$ in $(P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ $P_{il}\ \mathrm{ni}^P\ P'_{jm}$, the intersection of the two normal forms is empty. For example, *(Monday $\underline{\cup}^P$ Tuesday) $\underline{\cap}^P$ (Friday $\underline{\cup}^P$ Saturday)* is empty, because *Monday* ni^P *Friday*, *Monday* ni^P *Saturday*, *Tuesday* ni^P *Friday* and *Tuesday* ni^P *Saturday*.

If no emptiness is detected, the resulting periodicity can be obtained by applying the redundancy elimination rules to the formula obtained by concatenating by $\underline{\cap}^P$ the two input periodicities (i.e., to a formula like (α)).

Redundancy elimination. If there are two unions of basic periodicities $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})$ and $(P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ such that $\exists P'_{jm}$ in $(P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ such that $\forall P_{il}$ in $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})$ $P_{il} \subseteq^P P'_{jm}$, then the intersection of the two unions is $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})$ (and vice versa, by exchanging the role of the two union sets). For example, in *(SundayMorning $\underline{\cup}^P$ SundayAfternoon) $\underline{\cap}^P$ (Monday $\underline{\cup}^P$ NonWorkingDays)*, since *SundayMorning* $\subseteq^P$ *NonWorkingDays* and *SundayAfternoon* $\subseteq^P$ *Non-*

WorkingDays, the intersection is *(SundayMorning* $\underline{\cup}^P$ *SundayAfternoon)*.
Moreover, if $\exists P'_{jm}$ in $(P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ such that $\forall P_{il}$ in $(P_{i1}\ \underline{\cup}^P\ldots\ \underline{\cup}^P\ P_{is})$ $P_{il}\ \mathrm{ni}^P\ P'_{jm}$, P'_{jm} can be removed (and vice versa). For example, *(SundayMorning* $\underline{\cup}^P$ *SundayAfternoon)* $\underline{\cap}^P$ *(Monday* $\underline{\cup}^P$ *NonWorkingDays)* is equivalent to (*SundayMorning* $\underline{\cup}^P$ *SundayAfternoon)* $\underline{\cap}^P$ *NonWorkingDays* because *SundayMorning* ni^P *Monday* and *SundayAfternoon* ni^P *Monday*.

Union of normal forms.

(β) $((P_{11}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{1r})\ \underline{\cap}^P\ldots\underline{\cap}^P\ (P_{g1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{gf}))\quad \underline{\cup}^P$
$((P'_{11}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{1t})\ \underline{\cap}^P\ldots\underline{\cap}^P\ (P'_{h1}\ \underline{\cup}^P\ldots\ \underline{\cup}^P\ P'_{hn}))$.

By applying property 5, the composite periodicity above can be rewritten as follows:
$((P_{11}\ \underline{\cup}^P\ \ldots\ \underline{\cup}^P\ P_{1r})\ \underline{\cup}^P\ (P'_{11}\ \underline{\cup}^P\ \ldots\ \underline{\cup}^P\ P'_{1t}))\ \underline{\cap}^P\ldots\ \underline{\cap}^P\ ((P_{g1}\ \underline{\cup}^P\ldots\ \underline{\cup}^P\ P_{gf})\ )\ \underline{\cup}^P\ (P'_{h1}\ \underline{\cup}^P\ldots\ \underline{\cup}^P\ P'_{hn}))$
which reduces to the intersection of unions in the form

$$(P_{i1}\ \underline{\cup}^P\ \ldots\ \underline{\cup}^P\ P_{is})\ \underline{\cup}^P\ (P'_{j1}\ \underline{\cup}^P\ \ldots\ \underline{\cup}^P\ P'_{jk}).$$

Emptiness detection. If there exist two basic periodicities P_{il} and P'_{jm} such that $P_{il}\ \#^P\ P'_{jm}$, then the union of the two normal forms is empty. For example, in *(Monday* $\underline{\cup}^P$ *Tuesday)* $\underline{\cup}^P$ *Days*, since *Monday* $\#^P$ *Days*, the result is empty.
Moreover, let us consider the formula obtained after the application of the distributive property. If there is a union of unions $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})\ \underline{\cup}^P\ (P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ which is empty, the overall intersection is empty (from our definition, restricted union may introduce emptiness). Working on the corresponding basic periodicities, this translates to the following rule: if $\forall P_{il}$ in $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})$ and $\forall P'_{jm}$ in $(P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ $P_{il}\ \mathrm{ni}^P\ P'_{jm}$, then $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})\ \underline{\cup}^P\ (P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ (and the whole union of normal forms) is empty. For example, *(Monday* $\underline{\cup}^P$ *Tuesday)* $\underline{\cup}^P$ *(Friday* $\underline{\cup}^P$ *Saturday)* is empty, because *Monday* ni^P *Friday*, *Monday* ni^P *Saturday*, *Tuesday* ni^P *Friday* and *Tuesday* ni^P *Saturday*.
If no emptiness is detected, the resulting periodicity can be obtained by applying the redundancy elimination rules (to a formula like (β)).
Redundancy elimination. Every union in the form $(P_{i1}\ \underline{\cup}^P\ldots\ \underline{\cup}^P\ P_{is})\ \underline{\cup}^P\ (P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jk})$ can be simplified as follows: if $\exists P_{il}$ in $(P_{i1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P_{is})$ and $\exists P'_{jm}$ $(P'_{j1}\ \underline{\cup}^P\ldots\underline{\cup}^P\ P'_{jm})$ s.t. $P_{il}\subseteq^P P'_{jm}$, then

P_{il} can be removed from (P_{i1} $\underline{\cup}^P \ldots \underline{\cup}^P$ P_{is}) (and vice versa). For example, in *(SundayMorning* $\underline{\cup}^P$ *Tuesday)* $\underline{\cup}^P$ *(Sunday* $\underline{\cup}^P$ *Monday)*, since *SundayMorning* $\subseteq^P$ *Sunday*, can be removed from the formula, which is equivalent to (*Tuesday)* $\underline{\cup}^P$ *(Sunday* $\underline{\cup}^P$ *Monday)*.

Moreover, the rules concerning intersection of normal forms can also be applied at this point.

Property 6. Correctness and simplification incompleteness. The simplification rules of empty periodicity detection and redundancy elimination, applicable to composite periodicities, are correct; however, they are not simplification complete. □

Proof(hint). The correctness proof is similar to the one for basic periodicities; concerning incompleteness, see Note 1. For example, (January $\underline{\cup}^P$ February) $\underline{\cap}^P$ (4thWeekOfTheYear $\underline{\cup}^P$ 5thWeekOfTheYear) could be simplified as 4thWeekOfTheYear $\underline{\cup}^P$ 5thWeekOfTheYear, because the fourth and the fifth weeks of the year are included in the union of January and February. However, this simplification cannot be detected by our redundancy elimination rules because it is applicable only considering the periodicities (January $\underline{\cup}^P$ February) and (4thWeekOfTheYear $\underline{\cup}^P$ 5thWeekOfTheYear) as a whole.

3.7 Constraint propagation algorithm

Since our overall intensional calculus cannot be simplification complete (see Property 6), we chose to adopt an algorithm closely inspired to the Floyd-Warshall's all-pairs shortest paths algorithm, since at least it is complete as regards the *Con* part of our constraints.

Constraint propagation algorithm:

```
for each $ev_k$ in periodic events
    for each $ev_i$, $ev_j$ in periodic events
        let $PBC_{i,j}$ the periodicity-based constraint between $ev_i$ and $ev_j$, $PBC_{i,k}$ the periodicity-based constraint between $ev_i$ and $ev_k$ and $PBC_{k,j}$ the periodicity-based constraint between $ev_k$ and $ev_j$
          $PBC_{i,j} \leftarrow PBC_{i,j} \cap (PBC_{i,k}$ @ $PBC_{k,j})$
```

The algorithm iteratively applies the operations of intersection and composition a cubic number of times. The operations are applied modularly on *Con* components of the constraints (as defined in section 3.1) and on *Per* components (as defined in sections 3.5 and 3.6). Thus, given properties 4 and 6, the following property trivially holds:

Property 7. Our intensional temporal reasoning approach is correct, but it is not simplification complete. □

3.8 Example

The application of the constraint propagation algorithm on Example 1' discovers no inconsistency and gives as a result the periodicities in Figure 2 regarding the *Per* component of the constraints and the STP constraints in Figure 3 regarding the *Con* component of the constraints.
The algorithm has discovered, for example, the following constraints:

- <<email, presentation>, Monday, <(10m, 210m), (10m, 210m), (0,180m), (0,180m)>>, i.e. on Monday Sam starts showing a presentation from 10 minutes to 3 hours and a half (i.e., (10m, 210m) in the constraint) after starting reading his email and he shows the presentation after reading his email but no more than 3 hours later (i.e., (0, 180m)).
- <<meet_M, head>, WorkingDays, <(60m,480m), (120m,600m), (60m,420m), (120m,540m)>>, i.e., on working days, Mary starts talking to the head of her department 1 hour to 8 hours (i.e., (60m, 480m)) after the start of the meeting and she ends talking 2 hours to 9 hours (i.e., (120m,540m)) after the end of the meetings; moreover, she ends talking to the head of her department from 2 hours to 10 hours (i.e., (120m,600m)) after the start of the meeting and starts talking 1 hour to 7 hours (i.e., (60m,420m)) after the end of the meeting.

	meet_S	**presentation**	**boss**	**rpt_S**	**head**	**meet_M**	**rpt_M**
email	Monday	Monday	Monday	WorkingDays	WorkingDays	WorkingDays	WorkingDays
	meet_S	Monday	MondayAfternoon	WorkingDays	WorkingDays	WorkingDays	WorkingDays

presentation	Monday	WorkingDays	WorkingDays	WorkingDays	WorkingDays
	boss	WorkingDays	WorkingDays	WorkingDays	WorkingDays
		rpt_S	WorkingDays	WorkingDays	WorkingDays
			head	WorkingDays	WorkingDays
				meet_M	Monday& Tuesday

Figure 2: The result of the propagation of *Per* component of periodicity-based constraints in Example 1'.

		email		**meet_S**		**presentation**		**boss**		**rpt_S**		**head**		**meet_M**		**rpt_M**	
		S	**E**	**S**	**E**	**S**	**E**	**S**	**E**	**S**	**E**	**S**	**E**	**S**	**E**	**S**	**E**
email	**S**	**0**	90	90	210	210	210	450	510	330	390	510	630	450	450	480	510
	E	0	**0**	60	180	180	180	420	480	300	360	480	600	420	420	450	480
meet_S	**S**	-10	0	**0**	180	180	180	360	420	240	300	420	540	360	360	390	420
	E	-70	-60	0	**0**	0	0	240	300	120	180	300	420	240	240	270	300
presentation	**S**	-10	0	0	180	**0**	180	360	420	240	300	420	540	360	360	390	420
	E	-10	0	0	180	0	**0**	360	420	240	300	420	540	360	360	390	420
boss	**S**	-130	-120	-60	-60	-60	-60	**0**	120	60	120	120	240	60	60	90	120
	E	-190	-180	-180	-120	-120	-120	-30	**0**	0	30	30	150	-30	-30	0	30
rpt_S	**S**	-70	-60	-60	0	0	0	120	180	**0**	180	180	300	120	120	150	180
	E	-100	-90	-90	-30	-30	-30	60	120	0	**0**	120	240	60	60	90	120
head	**S**	-190	-180	-180	-120	-120	-120	-60	0	-30	0	**0**	150	-60	-60	-30	0
	E	-250	-240	-240	-180	-180	-180	-120	-60	-90	-60	-30	**0**	-120	-120	-90	-60
meet_M	**S**	290	300	300	360	360	360	420	480	450	480	480	600	**0**	120	120	180
	E	230	240	240	300	300	300	360	420	390	420	420	540	0	**0**	60	120
rpt_M	**S**	170	180	180	240	240	240	300	360	330	360	360	480	0	0	**0**	120
	E	110	120	120	180	180	180	240	300	270	300	300	420	-60	-60	0	**0**

Figure 3: The result of the propagation of *Con* component of periodicity-based constraints in Example 1'. In the table, T[i,j]=b and T[j,i]=-a conjunctively represent the bounds on difference a < j – i < b, where i and j are the starting/ending points of two events and a and b are measured in minutes.

5. Conclusions and comparisons

In this paper, we have described a comprehensive approach dealing with both (i) *qualitative* and (ii) *quantitative* periodicity-based temporal constraints, and considering also (iii) *user-defined* periodicities.

Dealing with these issues and with the interplay between them required to devise a novel approach that integrates and extends the STP framework [Dechter et al., 91] and the Leban's formalism [Leban et al., 86]. In particular, we have: (1) proposed a formalism to represent periodicity-based temporal constraints; (2) extended Leban's formalism to represent composite periodicities; (3) singled out five relations between periodicities and used them to (4) defined the operations of intersection and composition among periodicity-based temporal constraints considering both basic and composite periodicities; (5) described a constraint propagation algorithm based on the specifications at item (4), which is correct and provides perspicuous (intensional) output.

Several AI approaches have dealt with temporal constraints between periodic events. However, to the best of our knowledge, none of them proposed a comprehensive approach covering all the issues (i)-(iii) above. For example, in [Ligozat, 91; Morris et al., 96] only periodicity-independent (e.g., holding "*always*") qualitative constraints between repeated events have been considered. In [Bettini et al., 02], temporal constraints expressed at multiple user-defined granularities between non-repeated events have been tackled. Only periodicity-based quantitative constraints have been considered in [Loganantharaj & Giambrone, 96] and only periodicity-based qualitative constraints in [Terenziani, 97]. Notice also that none of the above-cited approaches has provided an intensional calculus about user-defined periodicities, which, on the other hand, has been devised by some algebraic approaches to Temporal Databases (consider, e.g., [Niezette & Stevenne, 92; Ning et al., 02; Terenziani, 03]).

References

[Allen, 83] J.F. Allen. "Maintaining Knowledge about Temporal Intervals", Comm. ACM, 26(11):832-843, 1983.

[Anselma, 04] L. Anselma, "Recursive Representation of Periodicity and Temporal Reasoning", Proc. TIME 2004, IEEE Society Press, pp. 52-59, 2004.

[Bettini & DeSibi, 99] C. Bettini, R. De Sibi, Symbolic Representation of User-defined Time Granularities, Proc. TIME'99, IEEE Computer Society, pp. 17-28, 1999.

[Bettini et al., 02] C. Bettini, S. Jajodia, and X. Wang, Solving multi-granularity constraints networks, *Artificial Intelligence*, 140(1-2):107-152, 2002.

[Brusoni et al., 97] V. Brusoni, L. Console, B. Pernici, and P. Terenziani. “Later: Managing Temporal Information Efficiently”. *IEEE Expert*, 12(4):56-64, July/August 1997.

[Dechter et al., 91] R. Dechter, I. Meiri, J. Pearl, "Temporal Constraint Networks", *Artificial Intelligence*, 49:61-95, 1991.

[Ladkin, 86] P. Ladkin, "Time Representation: A Taxonomy of Interval Relations", in Proc. AAAI’86*,* Philadelphia, PA, pp. 360-366, 1986.

[Leban et al., 86] B. Leban, D.D. McDonald, D.R. Forster, A representation for collections of temporal intervals, in Proc. AAAI’86, pp. 367-371, 1986.

[Ligozat, 91] G. Ligozat, "On Generalized Interval Calculi", in Proc. AAAI’91*,* pp. 234-240, 1991.

[Loganantharaj & Giambrone, 1995] R. Loganantharaj and S. Giambrone, "Representation of, and Reasoning with, Near-Periodic Recurrent Events", In *9th IJCAI Workshop on Spatial and Temporal Reasoning*, 1995.

[Meiri, 91] I. Meiri, "Combining Qualitative and Quantitative Constraints in Temporal Reasoning", in Proc. AAAI'91, pp. 260-267, 1991.

[Morris et al., 96] R.A. Morris, W.D. Shoaff, and L. Khatib, "Domain Independent Temporal Reasoning with Recurring Events", *Computational Intelligence*, 12(3):450-477, 1996.

[Niezette & Stevenne, 92] M. Niezette, and J.-M. Stevenne, "An Efficient Symbolic Representation of Periodic Time", In Proceedings *1st International Conference Information and Knowledge Management*, Baltimore, Maryland, 1992.

[Ning et al., 02] P. Ning, X.S. Wang, and S. Jajodia, An algebraic representation of calendars, Annals of Mathematics and Artificial Intelligence, 36(1-2):5-38, 2002.

[Soo & Snodgrass, 93] M. Soo, R. Snodgrass, Multiple Calendar Support for Conventional Database Management Systems, Proc. ITDB’93, 1993.

[Terenziani, 97] P. Terenziani, “Integrating calendar-dates and qualitative temporal constraints in the treatment of periodic events”, *IEEE TKDE* 9(5):763-783*,* 1997.

[Terenziani, 03] P. Terenziani, “Symbolic User-defined Periodicity in Temporal Relational Databases”, *IEEE TKDE*, 15(2):489-509, March/April 2003.

[Tuzhilin & Clifford, 95] A. Tuzhilin and J. Clifford, "On Periodicity in Temporal Databases", *Information Systems*, 20(8):619-639, 1995.

[Vila, 94] L. Vila. "A Survey on Temporal Reasoning in Artificial Intelligence", AI Communications 7(1):4-28, 1994.