IS TIME A REAL TIME?

AN OVERVIEW OF TIME ONTOLOGY IN INFORMATICS

Fabio A. Schreiber

Published in :
W.A. Halang, A.D. Stoyenko (Eds.) *Real Time Computing* Springer Verlag NATO ASI, Vol F 127,
Berlin 1994, pp. 283--307

Is Time a Real Time?
An Overview of Time Ontology in Informatics [⋆]

Fabio A. Schreiber

Dipartimento di Elettronica e dell'Informazione – Politecnico di Milano, P. zza L. da Vinci 32, I20133 Milano, Italy. *e-mail*: schreibe@ipmel2.elet.polimi.it

1 Introduction: the Nature of Time

Time drives our lives in a pervasive and convulsive way. It was not always so; people, today as in the past, have very different feelings about time. Lunch–time seems to be a synchronous, clock driven, event for the employees of a large city, but it was absolutely asynchronous for the stone–age men, who were driven only by hunger stimuli.

For a long time and in many different environments, men thought of time as something periodical, strictly connected to natural events; therefore in many ancient cultures, the Greek and Roman classical world included, time was mainly conceived with a *cyclic* structure. However, the first appearance of the modern conception of time dates back to the oldest "best–seller" ever written: the Bible! *Bereshit* – in the beginning, the first word in the book of Genesis (Fig. 1) – marks the start of a flow of events which go on through the history in a *linear* way towards the messianic era [Hes1]. Even if the idea of an open structure of time seems to have been present also in other ancient civilizations, such as the Zoroastrian Iranians [Wit1], it was through the Jewish/Christian thought that the western world got its view of time. Anyhow, periodic phenomena still retain their importance in the *measurement* of time, whatever it means, as we shall later see.

Things do not become easier even when we limit ourselves to give only a scientific connotation to time. On this ground, every high-school student could tell you that "time is the independent variable used to describe the laws of motion of mechanical systems", but we all know that modern physics is deeply rooted in the critique of the space and time concepts to such a point as theories and models, both at atomic and cosmological levels, are challenged today on the ground of the definition of time itself [Pau1].

In Computer Science and Engineering, time has always been of major concern, owing to the need of synchronization among the different functional units in the

---

[⋆] This work was partially supported by CNR-CSISEI and MURST

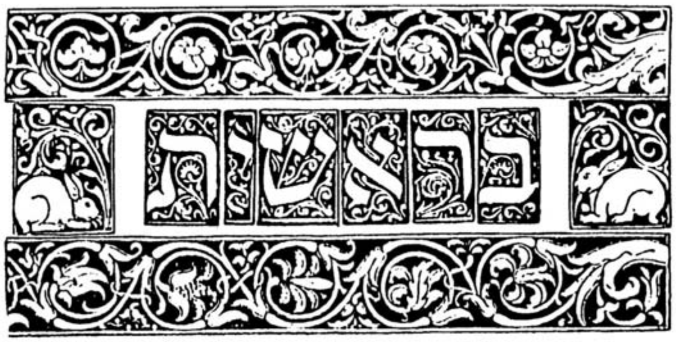


Fig. 1. *Bereshit*, (Soncino, Italy, 1488)

computer; the clock has always been the heart of any digital system. However the first problems which required a deep understanding of temporal issues came in with the birth of distributed computing systems, where the definition of a *global time* has to cope with finite propagation delays [Lam1]. We shall see that today, thanks to the widening of the dimensions of silicon devices achieved with Wafer Scale Integration, and with the very high switching rates attainable with ballistic transistor technologies, the notion of a single definite time is questionable even for a single circuit!

However, time is a concern not only for computer systems architects, but, during the eighties, it became increasingly relevant in many application fields, such as real-time control systems, information systems, automatic reasoning systems, planning systems, etc.. Accordingly, new kinds of time have been introduced such as calendar (wall-clock) time, execution time, simulation time, etc..

Aim of this paper is to introduce several aspects of time in the heterogeneous world of Informatics and define ontologies for time in different domains of computers and their applications, and not to discuss the very nature of time by itself, a philosophical problem which is to remain open forever. However, in sections 1.1 and 1.2 we are going to give some philosophical and physical background, because, from the richness of often contrasting ideas, developed in the framework of these disciplines, many useful concepts have been derived; we shall leave the reader to confront himself against them, following his personal belief. Section 2 deals with architectural aspects, and section 3 with information systems applications.

1.1 Philosophical Theories

Since the oldest times and in the commonsense feeling space and time have been perceived as something different from the other objects and processes; they have been considered either as some kind of structured containers of events in the natural world (absolutistic view) or just as some kind of abstraction to represent the relations among objects and processes (relational view). These views correspond to the different conceptions about the reality of the physical world and about the classical concept of “substance”.

However, while the modern conception gives space and time the same dignity, philosophers in the classical times – noticeably those of the Eleatic school – gave space a greater attention, while considering time just as a “disturbance” in their conceptual systems. Time, in particular, was given mainly a practical role, being connected to the changes evidenced by astronomical phenomena and by the problem of motion. Therefore its metrical properties were considered, while no clear distinction was made between events happening in time and time itself. This conception explains also the cyclical structure attributed to time since the most archaic thinkers, which survived up to the modern era. Typical of this conception is the Heraclitus doctrine of “transmutation” – later recalled by the Stoic school – in which the world was endlessly involved in a birth and death cycle, the substance remaining essentially unchanged. The links of time to the celestial phenomena is further pushed by Plato, for whom time is being produced by the revolution of the celestial sphere and was created with it.

The first to perceive the problem of differentiating time from movement was Aristotle [Ari1]; he thought that time was the numerable aspect of motion, in the ordering of “before” and “after”(IV, 11) [2], time and motion defining each other, while distinguishing between them since time cannot cease or change its speed like ordinary motions do. He also perceived that the uniform circular motion can be used to measure time (IV, 12)(IV, 14) and he raised the problem of the nature of “present” as a separating instant between “past” and “future” (IV, 10). However he arrives at seemingly contradictory conclusions since in (VIII, 1) he claims an infinite, open, and continuous structure of time, while in (VIII, 8) he states that infinity and continuity are only possible on the circle, thus – in this fact following Plato – linking time to the perfect circular and uniform motion of heavens.

It was the Atomistic school, mainly with Democritus and Hepicurus, the first who abandoned the cyclic view of time while retaining its infinity: time was not created, hence there is no beginning for time; however for them the future is still closed since evolution is but a transformation of states.

The relation of time to modal logic was another topic which concerned ancient thinkers; temporal quantifiers such as *sometimes, never, always, etc.* were directly connected to the notions of *possibility* and of *necessity* by the Megarians and the Stoics. Their views diverged on the inclusion of the present time in the definition of necessity and of possibility: Stoics – namely Diodorus Chronus – considered as **possible** that which is realized at *some present-or-future time*

---

[2] The numbering in parenthesis refer to book and chapter of [Ari1]

and as **necessary** that which is realized at *every future time*; Megarians, on the other hand, did not admit the *now-relativization* and defined as possible that which is realized at *some time* and necessary that which is realized at *all times* [RUr1].

We must make a jump of nearly a millennium before time is considered again as a philosophical issue; with St. Augustine the linearity of time is definitely assessed on the basis of theological arguments [Aug1] (even if the circular view survived for a long time). Another important issue is risen by Augustine, although the problem had already been pointed out by Aristotle [Ari1] (IV, 14): time is presented as a *subjective feeling*; only present is real, past is identified with memory and future with expectation, both memory and expectation being present facts *connected with the human mind* [Aug2].

In Middle Age also temporal modalities had a great success thanks to the Arabic logician Avicenna, to St Thomas Aquinas, and to the British school (William of Ockham, Albert of Saxony, and John Buridan) [RUr1]. We shall come back to this subject in section 3.

Therefore we can say that by the beginning of the 5th century A. D., and even well before, some major issues in the nature and in the structure of time had been identified:

- **linearity vs. circularity;**
- **finiteness vs. infinity;**
- **openness vs. closure;**
- **discreteness vs. continuity;**
- **absolute {past, present, future} vs. relative {before, concurrent-with, after} ordering;**
- **objectivity vs. subjectivity;**
- **definition of temporal modalities.**

During the 17th century a separation process begins between the natural sciences and philosophy, mainly due to methodological issues, but still we find the same topics (and often the same authors) going on in parallel on the two sides. In this period other major discussions on the nature of time were carried on by Newton and Leibnitz dealing with the foundations of mechanics, we are going to mention in section 1.2, and, after them, a few other contributions were made by philosophers of the rationalistic and of the empiricist schools, respectively supporting an *absolute* against a *relative* view of time, until the work of E. Kant.

Kant adds to the traditional components of knowledge – the empiric sensations and the rational intellect – a new component, he calls *intuition*, which is present, in its primary form, with two modalities: space, the form of external sensibility, and time, the form of the internal sensibility. Space and time allow the formulation of *a priori* synthetic judgments which put together the experience from the real world with the necessary universality of the pure reasoning [Kan1].

After Kant, physics and philosophy tend to concentrate on different goals, and space and time become definitely **scientific objects** themselves [Pau1].

Time comes again as a philosophical topic in the second half of the twentieth century, in the frame of logic and linguistics, as we shall discuss in section 3.

1.2 Physical Theories

The birth of modern Mechanics, thanks to Copernicus and Galilei, and its great development due to Kepler, Newton and others, puts time in a privileged, central position as *the independent variable* in the observation and description of the motion and the formulation of the physical laws.

Newton looks at time as a container of events, which homogeneously flows independently of anything else. It is an **absolute mathematical entity**, which can also be called *duration*. Relative time is the sensible measure of the duration we perceive by means of motion.

Leibnitz, on the other hand, can be considered the father of relativism. Space and time only represent **relative order relations** and they do not possess any objective substantiality.

The debate between the two opposite conceptions marks the following centuries until the fundamental work of A. Einstein [Ein1]. The logical foundation of the restricted theory of relativity is the discovery that many statements, whose truth or falsity was thought to be demonstrable, were only conventional definitions [Rei2]. The major consequence of this consideration is the *relativity of the simultaneity*; we can say that two events are simultaneous only if they happen under our *direct* perception. Otherwise we must observe the local clocks of each of them and stipulate – by convention – a procedure which tells us if the two clocks mark the same time. However, to do that, we must exchange information and, as it was experimentally proven, this can be done only at a finite speed, the speed of light.

Without going into the details of the theory of relativity, for which we refer to [Ein1] or to the tutorial exposition [Ein2], we mention here two results which are relevant from the ontological point of view: first, time looses its privileged position as the independent variable for describing natural phenomena, while only a *four coordinate space-time continuum* is used to express the physical laws; second, each event (a point in space-time) is the vertex of a twofold cone which contains the past and the future of the event itself. Points which belong to trajectories (*world lines*) lying within the cone are related to the vertex event by a precedence relation, which express **causality**. Points outside the cone are simply "elsewhere" and cannot be causally related to the event; they are independent of the event or *concurrent* with it. Therefore any interaction between two events can only occur within the intersection of their *light cones* as shown in Fig. 2.

The reality of *becoming*, a problem which also bothered some philosophers in ancient times, was risen during the 19th century by the discovery of irreversible processes in thermodynamics. The temporal precedence relation is asymmetric, but is it also unidirectional? In other words, is there a **time arrow**? While a great difference is empirically and psychologically observable between past

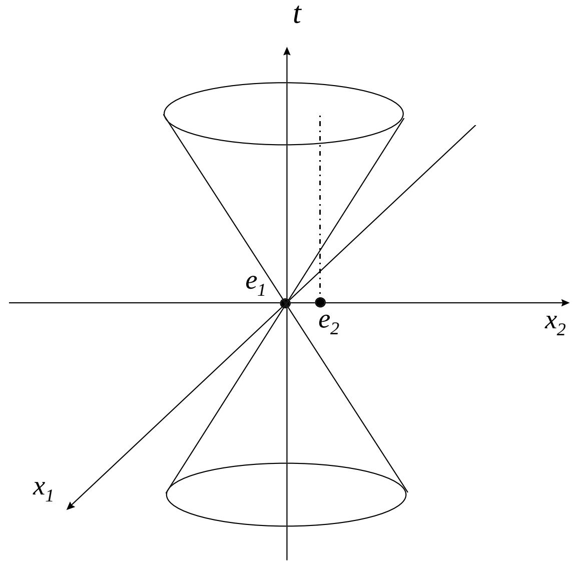


Fig. 2. a three-dimensional picture of the *light cone* of event e_1 and a *world line* of event e_2

and future (e.g. we can remember the past, but not the future and we can determine the future, but not the past), physicists still harshly debate whether there is a scientifically provable theory in support of an anisotropy of time or not [Rei2, Haw1].

The concepts of space and time discussed up to this point are the result of our perception of the world of the "middle dimensions"; when we enter in the world of microphysics, ruled by the laws of quantum mechanics, or in the deep spaces of relativistic cosmology we are not sure of the meaning itself of time. At the cosmological level we should define a *cosmic time*, to which all local clocks should synchronize, and this has been made through a set of very complex geometrical assumptions on the space-time [Pau1]. As to the quantum theory, the non-deterministic nature of the micro-objects forces to extrapolate to the micro world, in an analogical way, the time concept used in the macro world [Hei1]; even if the empirical results are very good, this make the theory inconsistent from an ontological point of view. The situation then is the following: the Universe is made by progressive aggregations of objects of increasing size, but no general definition can be given of time which encompasses all the dimensions of it, from elementary particles to galaxies and quasars. Only sophisticated mathematical theories, mostly inconsistent with each other, support the different views of today theoretical physicists.

So the question is legitimate: **is time a real time?**

2 Time in Computer Systems

Computers, as a class of very complex digital systems, have always been deeply concerned with timing problems. Digital signals in fact are discrete both in space (the space of their values) and in time (the sampling instants). Therefore, a circuit generating well defined periodical signals – *the clock* – has always been fundamental in many types of digital circuits and components.

Moreover, communication between different functional components inside the computer, for exchanging both control and data signals, need some information to synchronize their operations. This information is provided by the distribution of the clock signal around the chip, the board or the backplane of the computer.

A common abstraction which is made dealing with electronic circuits, consists in considering that the value of a signal is identical at all points along a given conductive path. The largest region for which this condition holds is called *equipotential region* [Sei1]. As long as the propagation delay of the signals is much smaller than their frequency, the equipotential region is as large as to include the whole machine; in this case we can consider the clock as an **absolute reference signal** and build synchronous devices relying on it. However the progress in electronic technology in the recent years has been such as to increase device speeds up to 1 ns delay per logical gate, while reducing conductor sections and crowding devices on the chip. The result is that, owing to the increase in resistance and capacitance of the paths, propagation delays did not improve at the same rate. Therefore even a single chip cannot be considered anymore an equipotential region, not to say of Wafer Scale Integration components [Fri1].

The presence of a *clock skew*, that is a difference of phase of the global clock signal at different circuit locations, poses the designer of digital circuits problems similar to those faced by the distributed computing systems designer; the circuit must be divided into several equipotential regions, which communicate with each other following some **precedence constraints**, thus abandoning the notion of an absolute time and assuming a **relativistic** attitude.

On the other hand, in distributed computing systems the partial ordering of events, induced by the precedence relation, is not enough if we want to obtain fault-tolerance, performance, or hard real-time properties, and synchronization with physical time becomes necessary [Lam1, Mok1]. In [Neu1] it is reported how a clock drift in a Patriot missile software prevented it from adequately tracking the incoming Scud that hit the Dhahran barracks during the Gulf war.

In the following, we shortly examine the problems of generating clock signals in logic networks, and in section 2.2 those related to an ordered and reliable exchange of information in distributed systems.

2.1 The Clock

The simplest way to model a clock is to use a boolean signal (a square wave) whose transitions between the two voltage levels mark the clock ticks. This abstraction, however, does not take into account some important phenomena occurring in real circuits: *metastability, finite rise time,* and *jitter*.

While metastability is an important issue, related to the nature of cross coupled circuits, that should be avoided by designers since it can delay the next tick forever, here we are more concerned with the other two problems since they influence the clock's precision. A finite rise time influences the definition of the exact instant at which the transition occurs, phase jitter consists in small variations of the intervals between transitions. Therefore a general model of a clock signal is [Mes1]

$$x(t) = p(((f + \Delta f)t + \Phi(t)) \text{ modulo } 1) \tag{2.1}$$

where:

- $p(t)$ is a 50% duty cycle square wave;
- f is the *nominal frequency*;
- Δf is a possible *frequency offset*;
- $\Phi(t)$ is the *instantaneous phase variation*.

$\Phi(t)$ can represent a deterministic signal, a random process (e.g. white noise), etc. If we assume that $\Phi(t)$ be a deterministic, continuous, and differentiable function, then – for a clock to be "good" – the following conditions must hold:

$$\begin{cases} \Phi(t) \leq \Phi_{\max} \\ \overline{\frac{d\Phi(t)}{dt}} = 0 \end{cases} \tag{2.2}$$

i. e. the phase must be bounded and its derivative must have a time average equal to zero.

The *instantaneous frequency* is defined as

$$f(t) = f + \Delta f + \frac{d\Phi(t)}{dt} \tag{2.3}$$

If the *average frequency* $(f + \Delta f)$ is a constant (i.e. the *instantaneous frequency deviation* $d\Phi(t)/dt$ averages to zero) the clock is **isochronous**, while if $\Delta f = \Delta f(t)$ the signal is **anisochronous** and its phase is not bounded. The jitter induced by a time-varying phase, with components greater than 10 Hz, is a very disturbing effect in digital communications and it must be neutralized by the use of *elastic store* registers.

The best clocks available today are based on transitions in the orbital states of an electron; the Hydrogen Maser has a *frequency stability* of $2 * 10^{-14}$ per day, with a maximum offset of $1 * 10^{-12}$ per year. These devices are used by national administrations to keep a standard reference time; the quartz oscillators used in computer systems show stabilities in the order of 10^{-6} per day.

It is out of the scope of this paper to go into the details of the structure and implementation of physical clocks, for which we address the reader to [Mil1, DSn1]; however we must mention the fact that the use of clocks in logic networks puts some constraints on the shape and duration of the pulses generated by the local clock and the clock period must be adjusted so that there is a *certainty period* during which the output signal is guaranteed to be correct [Mes2]. Then the simple boolean (one phase) clock and the combinational logic must obey the following rules (Fig. 3a) [Sei1]:

- the delay of the combinational logic must be
  - greater than the clock width;
  - less than the clock period;
- the clock width must be
  - greater than the time for charging the present state combinational inputs;
  - less than the minimum combinational delay.

If such a scheme could work with discrete components circuits, the satisfaction of all the constraints, in spite of the possible environmental changes, becomes unfeasible today for large systems, unless the clock is included on the integrated system chip.

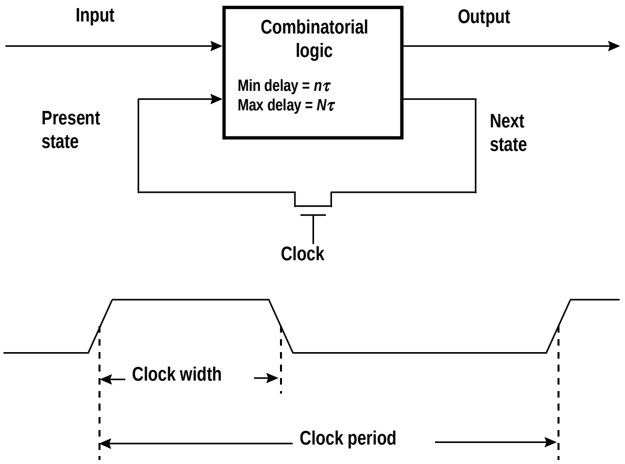
Therefore multiple phase clocks have been defined, in order to simplify the constraints and to provide a region of reliable operation in any working condition (Fig. 3b) [Sei1]. It should be noted that, when a clock skew is present, it is not always possible to choose the clock period large enough to fall into the certainty region and an additional delay must be added, so reducing the system throughput [Mes2].

2.2 Synchronization in Distributed Systems

A distributed computing system is one in which computational units are spatially separated and communicate between each other by exchanging messages. More generally, we can say that a system is distributed if *the message transmission delay is not negligible compared to the time between events in a single process* [Lam1]. The previous considerations, make the "spatially separated" attribute to shrink and to be applied to very different architectures as long as technology advances: wide area networks, local networks, multiprocessors, wafer scale circuits, etc. Moreover, in distributed systems we often find as many independent clocks as the number of computational units; this is the common case of a set of loosely connected processors. Therefore, in large computer networks the *Internet Network Time Protocol (NTP)* is adopted as a standard to synchronize the clocks of a number of *Time Servers* and to automatically organize and maintain a *Time Synchronization Subnet* to keep their accuracy with respect to the *Coordinated Universal Time (UTC)* provided by the national standards. The synchronization subsystem in Internet is *plesiochronous*, since there are several master oscillators which are closely synchronized in frequency, but not phase-locked to a single frequency standard [Mil1].

We can say in general that **synchronization is the action of making different processes in a computer network or different parts of a circuit or different clocks to agree on a same time reading**.

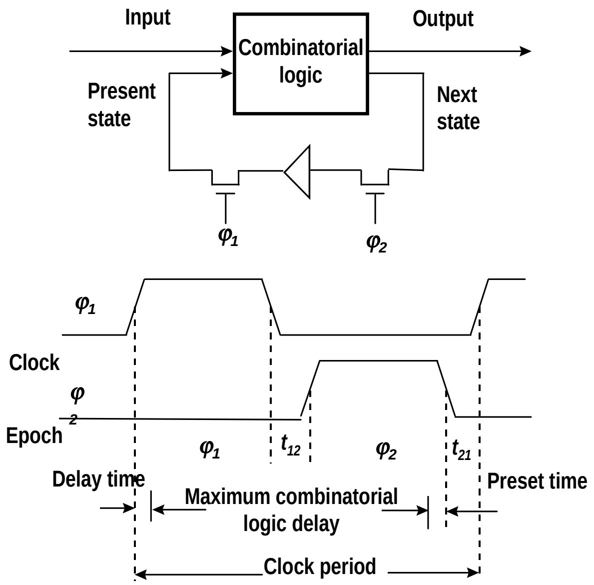
In the context of multiprocessors and of distributed systems, where processes are executed in a real concurrent environment, synchronization insures that operations occur in the logically correct order, and it allows the establishment of causal implications between events in different computational units. In fact,



$(R_{on}C_{in})_{max} <$ clock width $< n\tau$

$N\tau <$ clock period $<$ refresh period

a one-phase clock



b two-phase clock

Fig. 3. clock schemata

the *happened before* relation, which is implicit in any sequential computation, is meaningful and systemwide consistent only if events are referred to a single clock, otherwise it can only establish a *partial order* for the set of events happening under the same local clock, as we saw in section 1.2. As an example, such considerations are essential in many concurrency control and recovery algorithms for distributed transaction management.

Leslie Lamport was the first to recognize this fact, rising – within the computer scientists community – the **duality between a relative and an absolute time** [Lam1]. Partial ordering of events, in fact, can be obtained getting rid of any physical "real" time, by just relying on local *logical clocks* which are but a *non decreasing monotonic function mapping events on the set of integers*; any such a logical clock can be implemented by a counter without any link to actual time.

In an isolated distributed system, i.e. one which does not interact with an external environment, it is even possible to get a total ordering built on a set of partially ordered logical clocks; a distributed algorithm, based on the exchange of "time stamped" messages among the different processes, which achieves a somehow arbitrary total ordering was given in [Lam1]. Rules to obtain a "causal ordering" from partially ordered logical clocks appear in [Fid1].

However, in many cases, such as in real-time systems, in fault-tolerant systems, and to avoid anomalous behaviours (e.g. by actions which bypass the system's normal operation and could violate the cause-effect relation by "making holes" in the "light cone" of the events), the interactions between the system and the environment must be explicitly taken into account. This is possible recurring to physical clocks [Lam1, LMS1, Sha1]; therefore we must deal again with the precision and the drift of physical clocks and with the propagation delay occurring in message transmission [3]. In his paper [Lam1], Lamport gives a limit for avoiding anomalous behaviours when using physical clocks: be $\kappa \ll 1$ the *frequency drift* of a physical clock, ϵ the *greatest (small) difference between two clocks readings* at the same time, μ the *shortest transmission time* for interprocess communication, then the following inequality should hold

$$\frac{\epsilon}{(1-\kappa)} \leq \mu \tag{2.4}$$

To keep (2.4) valid over long time intervals, clocks must be periodically synchronized; an algorithm and a bound for the resynchronization period is also given in [Lam1]. Not surprisingly, the conditions set by this algorithm on the physical clocks behaviour closely match those in (2.2).

Anomalous behaviour however can arise in a distributed system not only due to shortcuts through the environment, but also because of faults in the clocks or in the processes involved in their synchronization. Some algorithms which maintain clock synchronization even in the presence of faults have been proposed in

---

[3] And we are lucky that, for the time being, space vehicles housing Earth-connected computing systems fly at speeds which are well below the speed of light, so we can use Newtonian space-time without requiring relativistic corrections!

the literature setting bounds on the worst-case skew, on the total number of clocks necessary (and sufficient) to obtain synchronization in presence of m malicious (Byzantine) faults, on the number of messages to be exchanged among the partners to obtain synchronization, etc. [WLy1, LMS1, KOc1]. Moreover, since the topic is an intriguing one, formal and mechanical proofs of their correctness (or of flaws in them) have been given [DHS1, Sha1]; a survey of software and hardware fault-tolerant synchronization in distributed systems is provided in [RSB1].

Other very interesting points about synchronization in *natural* systems can be found in [GCo1, Win1].

Looking back at this section from an ontological point of view, we notice that, in the generation and dissemination of a time-reference signal, the following questions are addressed:

- **physical clocks are continuous** while **logical clocks are discrete**, therefore an optimal granularity must be chosen, based on application constraints;
- **absolute time vs. relative time** is resolved in favor of the first, but at the price of establishing complex communication protocols for synchronizing local clocks;
- absolute time is used for establishing **causal** relations, which are essential in many applications to obtain reliability.

3 Time Representation in Information Systems

After dealing with the *nature* of time in section 1 and with its *materialization* in section 2, we are going to present here the different *representations* of time which have been used in Information Systems.

Philosophers began to consider time again in the middle forties, in the frame of cognitive sciences and linguistics. The first approaches [Rus1, Qui1] regarded time just as a variable in first order predicate calculus. *Tense Logic* was defined by Prior [Pri1] as a particular kind of modal logic, for reasoning about the modalities of the discourse; a systematic exposition of *Temporal Logic* was given by Rescher and Urquhart in [RUr1] and by Van Benthem in [Van2].

In the logicians' community there is a strong debate on the need of creating a non standard *Temporal Logic.* Scholars having mathematical and physical background and interests claim that times can be designated by terms in a first order theory, which is more than adequate for time modeling. Besides Russel and Quine, these authors – referred often to as *detensers* – comprise Allen, McDermott, Kowalski and others. People interested in linguistic aspect of logic, on the other hand, feel that time is tightly woven into languages, under the form of different tenses of the verb, and they relate modal to temporal notions, as we saw in section 1.1; Prior and Von Wright belong to this *tensers* school. Just to show how things become complicate, we only mention that, in his theory of tense, Reichenbach defines three different times for each tense: an *utterance* time, at which the sentence is expressed, a *reference* time, which we refer to in the sentence, and an *event* time, which is the object of the sentence; with such a scheme he can explain structures like the future perfect tense [Rei1]. We do not insist on this point because we think it has a technical more than a substantial nature; after all, the two approaches are not in contrast with each other since the first order approach can be used as an interpretation for the modal one [Gal2].

In the meanwhile, Temporal Logic arose the interest of computer scientists both for its capability of expressing the needs of temporal description in knowledge engineering applications and for providing a rigorous formalism for specification and verification of concurrent and Real-Time software. As Galton points out, in the first field "... temporal logic is used to *enable computer programs to reason about the world* ...", while in the second it is used "... *to enable the world to reason about computer programs* ..." [Gal2]. In the following, we shall review the fundamental ontological issues common to both fields, then the choices made for the different applications will be presented.

• **Primitive Time Entities.**

One of the most fundamental questions in the whole ontological issue, which raises a sort of "chicken-and-egg" problem, is the choice of the primitive time entity. Three are the options found in the literature:

- **points of time** (instants);
- **segments of time** (intervals);
- **occurrences in time** (events).

Obviously these concepts can be derived from each other [Van1], however the choice is not trivial as far as the application world modeling is concerned. Instants can be viewed as the limit of intervals shrinking to zero duration, while intervals are defined by a beginning and an ending point. Events are considered differently whether an **absolute** or a **relative** view is taken. In the first case, **time ontologically precedes events** since it is a container of events and it is used to distinguish among them; in the second case **events ontologically precede time** since events are used to identify time instants! Moreover events are spatio-temporal entities, as we saw in section 1.2, therefore their role as basic temporal entities is questionable.

• **Time Topology.**

Since Time has a dynamic nature, it comes spontaneous to define trajectories in time so that any two time entities can be related by an **order relation**; the question at issue is if the order is a *total* or a *partial* one. In the first case, each time entity has at most one predecessor and one successor; this property can be mapped on the infinite line - thus producing a true **linear** time - or on a closed (circular) line - thus producing a **periodic** time [4]. If only a partial order is allowed, one speaks of a **branching** time, in the sense that each time entity can have many predecessors (*branching in the past*) and many successors (*branching in the future*).

---

[4] To be rigorous, periodic time can be defined only after a *metric* has been defined; see later.

While the linear time model is the standard in natural sciences (e.g. physics), the **linear past – branching future** model has gained popularity among computer scientists because it allows the representation of a deterministic past (what happened did happen) together with an open future (all that could happen). The semantics regarding the topology of time is usually reflected by the temporal modalities provided for describing events [Mca1, BPM1, EHa1, Mcd1, MPn2].

Branching time raised a dispute on whether it should be interpreted as *branching of time* or *branching in time*, i.e. if branching is a structural property of time, or it is the *course of events* which branches in a linearly structured time. The second view has been generally accepted because of its simplicity and conceptual clarity (how could we establish precedence relations between divergent courses of time?). On the other hand, the distinction only makes sense in an absolutistic conception, while the relativistic point of view clears out the duality since events constitute time [RUr1].

- **Temporal Relationships.**

One of the first consequences of an instant based against an interval based description emerges when we consider the relationship between any two entities. With a point description, a simple **precedence** relation is all that we need; with intervals we need a set of 13 different relations (Fig. 4): **before(x y), meets(x y), overlaps(x y), during(x y), starts(x y), finishes(x y),** their inverses and **equal(x y)** [All1].

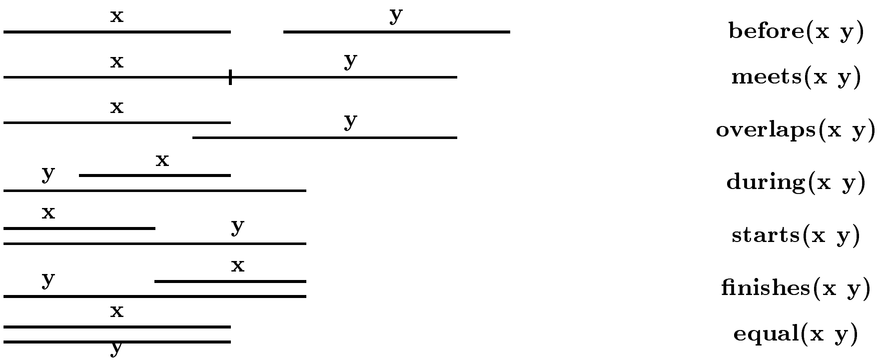


Fig. 4. relationships between two intervals

- **Boundedness.**

The relationship between time entities exhibits some problems as far as the boundary conditions are concerned. We can ask whether every time has a *successor (predecessor)* or if there is a last (first) moment in time. Notice that boundedness does not imply that time is *finite* or *infinite*, which is a metrical property.

If no beginning and no ending times exist, we can define a **homogeneous** time structure, where the local environment of any temporal position is temporally equivalent to any other. This makes simulation possible by compressing a long *real time* into a shorter *simulation time* on the continuous time line [5].

Speaking in terms of intervals, the question is whether they are **open (closed)** at one or at both ends. This property is especially relevant for the **meets(x y)** relation, where a *dividing instant* exists when some variable which is **true** in x becomes **false** in y (or viceversa), i.e. a state change intervenes between the two intervals. If we remain in a classical logic context, symmetric models of the kind **closed-closed** or **open-open** must be ruled out in the description of physical systems, because it is impossible that p and $\neg p$ be both false (Excluded Third principle) or both true (Non Contradiction principle) at the same instant; in other terms, we should choose between contradiction and indeterminacy!

- **Time Structure.**

The problem above is closely connected to the kind of structure we use to represent time. Many authors feel time is **continuous** in the sense that time points can be mapped on the set $\mathcal{R}$ of real numbers and time intervals are totally covered by a set of subintervals. Others think that a **dense** structure, i.e. one that maps time entities on the set $\mathcal{Q}$ of rationals, is enough to represent all real life situations. Finally a strong claim is made in many software engineering applications for a **discrete** time structure, mapping points or intervals on the integers set $\mathcal{Z}$. Obviously, choosing $\mathcal{Q}$ or $\mathcal{Z}$ makes the problem of the dividing instant to vanish, while $\mathcal{R}$ and $\mathcal{Q}$ rule out finiteness.

A deep discussion on time entities and their algebraic and logic structures and implications can be found in [Van1, Van2].

- **Temporal Metric.**

Both in Artificial Intelligence and in Software Engineering applications, purely logical and topological features are not enough to model all the properties of real systems. Quantitative concepts must be introduced into temporal systems, transforming them into **chronological** systems, by defining a *distance* function between any two time entities, which satisfies at least the following conditions:

a) the **null** distance between two entities is defined

$$d(t,t') = 0 \text{ iff } t = t';$$

b) distances among any three entities obey the triangular inequality

$$d(t,t') + d(t',t'') \geq d(t'',t').$$

Therefore, the set of values for the temporal variables constitute a **metric space** in which it is possible to correlate an arbitrary time entity with a unique number, which represents the distance of the entity itself from a *reference element.* Usually this is assumed to be the identity element of the additive group formed by the set

---

[5] This is also possible with circular time, but not with branching time [RUr1]

of the time values together with the (relative) addition operation; in this case the identity element is the *zero* and distances add up in the usual arithmetic way [RUr1, Koy1]. Such groups have an intrinsic linear ordering relation which is symmetric with respect to past and future, the identity element behaving as the *present time*.

In true branching time structures the notion of a metric time becomes tricky. In fact one could measure distances between the projections of the elements on an underlying (linear) time axis, but simultaneity could bring to a violation of condition *a*); moreover *we are not assured* that clocks in different branches behave in the same way, thus granting the comparability of the measures [RUr1].

The introduction of a metric entails the definition of *metric units*, which define the **granularity** of the time structure, e.g. *year, day, minute, etc.*. Moreover, we can express a temporal proposition in terms of a chronologically *stable* time specification, i.e. a **date** such as "April 26th 1992", or in terms of a chronologically *unstable* one, i.e. a **pseudo-date** such as "the day after tomorrow".

Finally, it is the introduction of metric time which allows us to define periodic (circular) time and its representation as a sequence of intervals of equal length on the time line. Such a representation is useful to clearly distinguish between past and future of repeating events, whereas on the circle they coalesce. More complex spatial topologies, such as *helix*, which can account both for periodicity and evolution in time, are seldom mentioned in the literature.

A compact and structured summary of the axioms and rules of temporal logic most relevant to the ontology of time can be found at the end of [Sch1]; implications of special relativity and thermodynamics on tense logic are discussed in [Bur1].

3.1 Applications in Reasoning and in Planning Systems

If natural language understanding was the first application area which moved researchers in Artificial Intelligence to deal with time representation, it was the possibility of building systems which could reason about temporal facts which stimulated the most fruitful practical approaches. The application domains to which time reasoning has been applied are: planning [Mcd1, DMD1], Decision Support Systems [DPW1], Fault Diagnosis [MPe1], office systems [MPB1], deductive and temporal databases [KSe1, SAh1].

The work of J. F. Allen must be acknowledged for its attempt to set the foundations for time reasoning systems [All1, All2, All3]; he is concerned with *properties* of objects which *HOLD* in *every subinterval* of an interval, with *events* which define the interval in which they *OCCUR*, and with *processes OCCURRING* during *some subinterval* of an interval.

From the ontological point of view, the only thing on which authors agree is the use of *first order calculus against modal logic* (although time modalities are treated in [MPB1] and some mixed approaches are presented in [Gal1]). With respect to all the other aspects the choices have been different and strongly debated [Gal3, Tur1]; while Allen favors intervals as basic time entities against point based descriptions, Galton, Mc Dermott and others use instants, and Kowalski uses events. Discrete time structures are often preferred, unless continuous phenomena are to be described, but this does not seem a critical issue, while authors contend again on the matter of linear vs. branching time. Table 1 shows a comparison of the ontological features of some interesting proposals; a clear survey and comparison of several time reasoning systems appears in [MPe1].

Table 1. reasoning systems comparison

Author	Allen[All1]	McDerm.[Mcd1]	TSOS[MPB1]	EC[KSe1]	TMM[DMD1]
Time Primitive	interval	point	point	event	point
Time Reference	relative	relative	abs/rel	relative	relative
Time Topology	linear	R-branch.	linear	linear	R-branch.
Time Bounds	[–)	∞	[–) ; ∞	(–)	[→ ∞
Time Structure	continuous	continuous	discrete	discrete	discrete
Time Metric	no	no	yes	no	yes

[–) means left-closed, right-open interval

Another controversial point in reasoning systems is in the possibility of establishing a *causal relation* between events related by a pure precedence relation. As we saw, causation is the source of a millenary debate among philosophers and scientists, who support or deny a cause-effect relation; a proof that things are far from being settled is given by the work of Shoham [Sho1] and by the critical reviews of it [NDe1].

As a last point, I remember Temporal Databases as a complex form of time management in information systems. Traditional Database systems are static representations of the last updates performed on the stored information. However sometimes it could be necessary to keep the history of the updates to the database and also to update retroactively the information in the Database, on the base of data which have been known only recently. These operations require the addition of a temporal dimension to the DBMS and the definition of a *transaction time*, i. e. the time the information was stored in the Database, and of a *valid time*, i. e. the time when the relationship in the enterprise being modeled was valid [SAh1, TSC1]. These two times, together with the *query time*, i. e. the time the query is done, closely resemble the Reichenbach triad [Rei1], mentioned before, query time being analogous to utterance time, valid time to event time, and transaction time to reference time. The last of a series of comprehensive bibliographies on all the different aspects of Temporal Databases appeared on ACM SIGMOD Record [Soo1].

3.2 Applications in Software Engineering

The interest of software engineers for timing problems first arose in the context of formal techniques for program verification, when methods successfully used with sequential programs failed to work with *concurrent* programs. In fact, the input-output relations computed by a program composed of two parallel processes cannot be inferred by combining the input-output relations computed by the two processes running independently, since they can be altered by possible interference between the processes themselves [MPn1]. Standard propositional or first-order calculus has been used to model the static part, while the dynamics is modeled by modal logic.

Very quickly the study was generalized to *reactive* systems, i. e. those programs whose role is to maintain an ongoing interaction with their environment, which include - besides concurrent programs - Real-time programs, and Operating Systems [MPn2]. Many reactive programs do not even terminate (e. g. Operating Systems and Real-time process control systems), therefore the traditional verification techniques must be directed toward new classes of properties; Manna and Pnueli define them as follows [MPn1]:

- **Invariance (Safety)**: program properties that hold continuously throughout the execution; they assure that *nothing bad will happen.* They are expressed through the **always** (necessity) modal clause $\Box p$.
- **Eventuality (Liveness)**: program properties that guarantee that some event will finally be accomplished; they assure that *something good will happen.* They are expressed through the **sometimes** (possibility) modal clause $\Diamond p$.
- **Precedence (Fairness)**: program properties stating that a certain event always precedes another; they are expressible using the **until** operator Up.

In these classes all the interesting properties of reactive systems are included, e. g. partial correctness and deadlock freedom (invariance), total correctness and responsiveness (eventuality), absence of unsolicited response and fair responsiveness (precedence).

We shall only mention that, together with the standard boolean connectives, the first-order universal $\forall$ and existential $\exists$ quantifiers, the modal clauses (necessity having a universal nature and possibility having an existential one), and the **until** operator introduced above, many other temporal operators have been introduced and utilized in the literature; noticeably: **next** $\bigcirc p$, **unless** $\mathcal{W}p$, **since** $\mathcal{S}p$. The $\Box p$ (always p) and $\Diamond p$ (sometimes p) operators, including the present instant as part of the future, are the reflexive counterparts of the Priorean strict operators Gp (it will always be the case that p) and Fp (it will be the case that p).

Another distinction was made between the two operators **sometimes** $\leadsto p$ and **not never** $\Diamond p$ whose meanings are equivalent in linear time, but differ in branching time [Lam2, EHa1].

From an ontological point of view, it is interesting to notice that *a temporal logic with the* **until** *operator and restricted to the future fragment* is sufficient to express all the first-order properties of reactive systems [Gal2, MPn2], other operators for future and past being useful, but not necessary. This can be explained with the fact that proving programs properties requires reasoning from a program step onwards and not viceversa.

The need of formal specification languages comes out as a logical consequence of the possibility of formally proving program properties, since formal proofs need a formal statement of functional as well as of performance requirements. Several proposals have been made, mainly in the frame of Real-time process control applications, where timing constraints are an explicit requirement in assuring system integrity.

Table 2 shows a comparison among some well known proposals of verification and specification systems. We can notice that all of them, except RTL, use modal logic (against reasoning systems, which prefer standard logic), systems applied to verification of concurrent programs preferring propositional calculus, while Real-time specification languages prefer 1^{st} order calculus. The two application domain differ also as to the adoption of a metric, which is necessary for real-time, whereas relative ordering is enough for concurrency. Other ontological issues common to most proposal are:

- the *discreteness* of the time model (except Koymans, who is interested in continuous process control) ;
- the choice for *points* or instantaneous events as basic time entities (except ITL).

Once again, time topology requires a special consideration, since in software engineering, as well as in reasoning systems applications, it is a major source of debate among the different proposals. Contrasting with the simplicity of linear time logics, very complex branching time structures have been defined. In branching time models two kind of formulas can be defined: *state formulas*, which are valid at each time point, and *path formulas*, which hold over an entire path spanning many branches. In [BPM1] quantification over possible futures and quantification over individual times are tightly combined into six non decomposable modalities, while in [EHa1] the path quantifiers $\forall$ (for all paths) and $\exists$ (for some path) can be applied to expressions containing state quantifiers F, G, U, etc. to convert path formulas to state formulas and viceversa. Expressive power of the resulting model is the goal of branching time advocates [EHa1] and really they succeed in providing it, however for most practical applications linear time models are sufficient.

Besides linear and branching time, a third interesting topology is related to periodic phenomena; we saw above that circular time has been a very popular model in the past and that periodic time can be represented on the line by the series of the repeating events. A topology that integrates both aspects of periodicity and evolution in time is the *helix* in a three-dimensional space, where the base plane projection describes periodicity and the third coordinate axis describes evolution. A practical example of such a coordinate scheme is given by the "@" function in RTL, where the value of @(e,i) is the time of occurrence of the i^{th} instance of the event e, as shown in figure 5.

Table 2. formal verification and specification systems proposals

Author	Lamport [Lam2]	Ben Ari [BPM1]	Manna [MPn1]	Emerson [EHa1]
Scope	conc. ver.	conc. ver.	conc. ver.	conc. ver.
Approach	modal propos.	modal propos.	modal propos.	modal propos.
Primitive	point	point	point	point
Reference	–	–	–	–
Topology	linear	branch/circ	linear	branching
Direction	future	future	future	future
Bounds	[→ ∞	∞	[→ ∞	[→ ∞
Structure	discrete	discrete	discrete	discrete
Metric	no	no	no	no

[→ means left-closed, right-open interval

Author	Koymans [Koy1]	RTL [JMo1]	ITL [Mos1]	TRIO [FMM1]
Scope	RT spec./ver.	RT spec./ver./synth.	conc. spec./ver.	RT spec./ver.
Approach	modal 1^{st} ord.	non modal 1^{st} ord.	modal 1^{st} ord.	modal 1^{st} ord.
Primitive	point	event (point)	interval	point
Reference	relative	abs/rel	relative	relative
Topology	linear	helix	linear	linear
Direction	past/future	–	future	past/future
Bounds	∞	[→ ∞	[→ ∞	∞
Structure	continuous	discrete	discrete	cont./discr.
Metric	yes	yes	no	yes

[→ means left-closed, right-open interval

4 Conclusions

So, can we answer the question: "Is time a real time?"? Quoting St. Augustine [Aug2], we could say: "... I know well enough what it is, provided that nobody asks me about; but if I am asked what it is and I try to explain, I am baffled". Fortunately, the problems computer engineers are faced with do not ask for metaphysical answers on the very nature of time, but they need a set of pragmatic guidelines which could assist designers and programmers in the realization of architectures and applications.

We saw that the fundamental issues are the same both for hardware and for software, therefore a distinction on this ground does not seem useful, while a classification of temporal models based on the *realistic* requirements of the different applications seems much more fruitful. In this sense we can conclude that **yes; everybody can have his "real" time!**

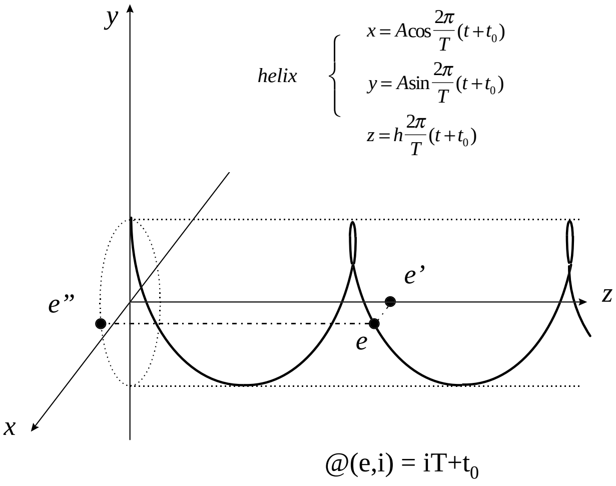


Fig. 5. topology of the @ function in RTL

As a last thing, I want to thankfully mention Angelo Morzenti, Massimo Pauri and Barbara Pernici for the time spent in helpful discussions on time.

References

[All1] Allen J.F. – Maintaining Knowledge about Temporal Intervals – Communications of ACM, vol. 26, n. 11, 1983

[All2] Allen J.F. – Towards a General Theory of Action and Time – Artificial Intelligence, vol. 23, pp. 123-154, 1984

[All3] Allen J.F. – Time and Time Again: The Many Ways to Represent Time – Int. Journal of Intelligent Systems, vol. 6, pp. 341-355, 1991

[Ari1] Aristotle – Physica – Laterza, Bari 1991

[Aug1] Augustine St. – The City of God, book XII, ch. XIII

[Aug2] Augustine St. – Confessions, book XI, ch. XXX

[BPM1] Ben-Ari M., Pnueli A., Manna Z. – The Temporal Logic of Branching Time – Proc. 8th ACM Symp. on Principles of Progr. Lang., pp. 164-176, 1981

[Bur1] Burgess J.P. – Basic Tense Logic – in D. Gabbay, F. Guenter (Eds.) "Handbook of Philosophical Logic", vol.II, pp. 89-133, Reidel Dordrecht 1984

[DPW1] De S., Pan S., Whinston A. – Temporal Semantics and Natural Language Processing in a Decision Support System – Information System, vol. 12, n. 1, 1987

[DMD1] Dean T.L., McDermott D. – Temporal Data Base Management – Artificial Intelligence, vol. 32, pp. 1-55, 1987

[DHS1] Dolev D., Halpern J.Y., Strong H.R. – On the Possibility and Impossibility of Achieving Clock Synchronization – Journal of Computer and System Sciences, vol. 32, pp. 230-250, 1986

[DSn1] Dyreson C.E., Snodgrass R.T. – Time-stamp Semantics and Representation – TempIS Technical Report n. 33, Dept. of Computer Science, Univ. of Arizona, Tucson 1992

[Ein1] Einstein A. – The Meaning of Relativity – 5th ed., Princeton Univ Press, Princeton 1955

[Ein2] Einstein A. – Relatività: esposizione divulgativa – (Transl. from "Ueber die spezielle und allgemeine relativitaetstheorie: gemeinverstaendlich, 16th ed.), Boringhieri, Torino 1960

[EHa1] Emerson E.A., Halpern J.Y. – "Sometimes" and "not never" Revisited: on Branching vs. Linear Time – Journal of ACM, vol. 33, n. 1, pp. 151-176, 1986

[FMM1] Felder M., Mandrioli D., Morzenti A. – Proving Properties of Real-time Systems Through Logical Specifications and Petri Net Models – Dip. Elettronica Politecnico di Milano, rep. n. 91-072, 1991

[Fid1] Fidge T. – Logical Time in Distributed Computing Systems – IEEE Computer, vol. 24, n. 8, 1991

[Fri1] Fried J. – An Analysis of Power and Clock Distribution for WSI Systems – in G. Saucier, J. Trilhe (Eds) "Wafer Scale Integration", North-Holland, Amsterdam 1986

[Gal1] Galton A.P. – Temporal Logics and their Applications – Academic Press, London 1987

[Gal2] Galton A.P. – Temporal Logic and Computer Science: an Overview – in [Gal1]

[Gal3] Galton A.P. – A critical Examination of Allen's Theory of Action and Time – Artificial Intelligence, vol 42, pp. 159-188, 1990

[GMM1] Ghezzi C., Mandrioli D., Morzenti A. – TRIO a Logic Language for Executable Specification of Real-time Systems – Journal of Systems and Software, vol. 12, n. 2, 1990

[GCo1] Goodwin B.C., Cohen M.H. – A Phase-shift Model for the Spatial and Temporal Organization of Developing Systems – Journal of Theoretical Biology, vol. 25, pp. 49-107, 1969

[Haw1] Hawking S.W. – A Brief History of Time – Bantam Books, New York 1988

[Hei1] Heisenberg W. – I pricipi fisici della teoria dei quanti – (Transl. from "Die physikalischen Prinzipien der Quantumtheorie"), Boringhieri, Torino 1963

[Hes1] Heshel A.J. – The Shabbath: Its Meaning for Modern Man – Farrar, Straus & Giroux Inc., New York 1951

[JMo1] Jahanian F., Mok A. – Safety Analysis of Timing Properties in Real-Time Systems – IEEE Transactions on Software Eng., vol SE-12, n. 9, 1986

[Kan1] Kant E. – Critica della ragion pura – (Transl. from "Kritik der Reinen Vernunft"), Laterza, Bari 1949

[Koy1] Koymans R. – Specifying Real-Time Properties with Metric Temporal Logic – Real-Time Systems, vol. 2, pp. 255-299, 1990

[KOc1] Kopetz H., Ochsenreiter W. – Clock Synchronization in Distributed Real-Time Systems – IEEE Trans. on Computers, vol. C-36, n. 8, 1987

[KSe1] Kowalski R.A., Sergot M.J. – A Logic Based Calculus of Events – New Generation Computing, vol. 4, pp.67-95, 1986

[Lam1] Lamport L. – Time, Clocks and the Ordering of Events in a Distributed System – Communications of ACM, vol. 21, n. 7, 1978

[Lam2] Lamport L. – "Sometimes" is sometimes "not never": on the Temporal Logic of Programs – Proc. 7th. ACM Symp. on Principles of Progr. Languages, pp. 174-185, Las Vegas 1980

[LMS1] Lamport L., Melliar-Smith P.M. – Synchronizing Clocks in the Presence of Faults – Journal of ACM, vol. 32, n. 1, 1985

[MPe1] Maiocchi R., Pernici B. – Temporal Data Management Systems: A Comparative View – IEEE Transactions on Knowledge and Data Engineering, vol. 3, n. 4, 1991

[MPB1] Maiocchi R., Pernici B., Barbic F. – Automatic Deduction of Temporal Information – to appear in ACM Transactions on Database Systems

[MPn1] Manna Z., Pnueli A. – Verification of Concurrent Programs: the Temporal Framework – in R.S. Boyer, J. Strother Moore "The Correctness Problem in Computer Science", Academic Press, London 1981

[MPn2] Manna Z., Pnueli A. – The Temporal Logic of Reactive and Concurrent Systems: Vol. 1, Specification – Springer-Verlag, New York 1992

[Mca1] McArthur R. – Tense Logic – Reidel, Dordrecht 1976

[Mcd1] McDermott D. – A Temporal Logic for Reasoning about Processes and Plans – Cognitive Science, vol. 6, pp. 101-155, 1982

[Mes1] Messerschmitt D.G. – Synchronization – in T.H. Meng "Synchronization Design in Digital Systems", Chap. 2, Kluwer Academic Publishers, Boston 1991

[Mes2] Messerschmitt D.G. – Isochronous Interconnect – in T.H. Meng "Synchronization Design in Digital Systems", Chap. 6, Kluwer Academic Publishers, Boston 1991

[Mil1] Mills D.L. – On the Chronometry and Metrology of Computer Network Timescales and their Application to the Network Time Protocol – ACM Computer Communication Review, vol. 21, n. 5, 1991

[Mok1] Mok A.K. – Towards Mechanization of Real-Time Systems Design – in A.M. van Tilborg, G.M. Koob "Foundations of Real-Time Computing:Formal Specifications and Methods", Kluwer Academic Publishers, Boston 1991

[Mos1] Moszkowski B. – Executing Temporal Logic Programs – LNCS 164, Springer-Verlag, Berlin 1984

[NDe1] Naur P., Delgrande J. – Reviews of [Sho1] – ACM Computing Reviews, vol. 30, n. 1, 1989

[Neu1] Neumann P.G. – Survivable Systems – Communications of ACM, vol 35, n.5, 1992

[Pau1] Pauri M. – The Universe as a Scientific Object – in E. Agazzi, A. Cordero (Eds.) "Philosophy and the Origin and Evolution of the Universe", Kluwer Academic Publishers, Boston 1991

[Pri1] Prior A.N. – Diodorean Modalities – The Philosophical Quarterly, vol. 5, pp. 202-213, 1955

[Qui1] Quine W.V. – Elementary Logic – Harper & Row, New York 1965

[RSB1] Ramanathan P., Shin K.G., Butler R.W. – Fault-Tolerant Clock Synchronization in Distributed Systems – IEEE Computer, vol. 23, n. 10, 1990

[Rei1] Reichenbach H. – Elements of Symbolic Logic – Macmillan, New York 1947

[Rei2] Reichenbach H. – The Philosophical Meaning of the Relativity Theory – in P.A. Schilpp (Ed.) "Albert Einstein: Philosopher–Scientist" The Library of Living Philosophers, Evanston 1949

[RUr1] Rescher N., Urquhart A. – Temporal Logic – Springer-Verlag, Berlin 1971

[Rus1] Russel B. – Principles of Mathematics – George Allen and Unwin, London 1903

[Sch1] Schreiber F.A. – A Breviary to Time Concepts for Information Systems – Rivista di Informatica, vol. XX, n. 1, 1990

[Sei1] Seitz C.L. – System Timing – in C. Mead, L. Conway "Introduction to VLSI Systems", Chap. 7, Addison-Wesley, Reading (Mass.) 1980

[Sha1] Shankar N. – Mechanical Verification of a Generalized Protocol for Byzantine Fault Tolerant Clock Synchronization – in Vytopil J. (Ed.) – Formal Techniques in Real Time and Fault Tolerant Systems – LNCS n. 571, Springer-Verlag, Berlin 1992

[Sho1] Shoham Y. – Reasoning about Change, Time and Causation from the Standpoint of Artificial Intelligence – MIT Press, Cambridge 1988

[SAh1] Snodgrass R., Ahn I. – Temporal Databases – IEEE Computer, vol. 19, n. 9, 1986

[Soo1] Soo M.D. – Bibliography on Temporal Databases – ACM SIGMOD Record, vol. 20 n. 1, 1991

[TSC1] Tansell A., Snodgrass R., Clifford J., Gadia S., Segev A. (Eds.) – Temporal Databases: Theory, Design and Implementation – Benjamin-Cummings, Reading 1993

[Tur1] Turner R. – Logics for Artificial Intelligence – Ellis-Horwood, Chichester 1984

[Van1] Van Benthem J.F. – Time, Logic, and Computation – in J.W. De Bakker, W.P. de Roever, G. Rozenberg (Eds.) "Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, LNCS 354, Spriger Verlag, Berlin 1989

[Van2] Van Benthem J.F. – The Logic of Time – 2nd edition, Reidel, Dordrecht 1991

[WLy1] Welch J.L., Lynch N. – A New Fault-Tolerant Algorithm for Clock Synchronization – Proc. Third ACM SIGACT-SIGOPS Symp. on Principles of Distributed Computing, 1984

[Win1] Winfree A.T. – The Geometry of Biological Time – Springer-Verlag, Berlin 1990

[Wit1] Withrow G.J. – Reflections on the History of the Concept of Time – in J.T. Fraser et Al. (Eds.) "The Study of Time", Springer-Verlag, Berlin 1972

An exhaustive list of bibliographical references on time could take an entire volume by itself. Therefore we urge the interested reader to follow, like a pointers chain, the references included in the items above.

This article was processed using the LaTeX macro package with LMAMULT style