KOS

Nuova serie n° 141, (1997)

pp. 34-39

IL TEMPO DELL'INFORMATICA E' UN TEMPO REALE?

di Fabio A. Schreiber

Prof. Fabio A. Schreiber
Professore Ordinario
Dipartimento di Elettronica e Informazione
Politecnico di Milano

Vengono introdotti alcuni aspetti relativi al tempo nel variegato mondo dell'Informatica e si definisce l'ontologia del tempo nel contesto dei calcolatori elettronici e delle loro applicazioni. La stessa architettura dell'Hardware e del Software dei calcolatori, nonchè le loro applicazioni ai Sistemi Informativi, ai sistemi di Intelligenza Artificiale e ai sistemi in Tempo Reale fanno dei concetti temporali uno strumento di lavoro fondamentale. I concetti utilizzati affondano le loro radici nel pensiero di filosofi e fisici che, nel corso dei secoli, hanno sviluppato un ricco insieme di idee e di teorie temporali.

Comprendere la natura del tempo è un problema che da sempre ha affascinato e stimolato i più grandi pensatori dall'antichità ai nostri giorni; Aristotele, Epicuro, Sant'Agostino, Kant e Einstein, oltre a tanti altri, hanno lasciato la loro impronta negli studi sul tempo contribuendovi al di là delle loro diverse concezioni filosofiche, scientifiche e religiose e dei secoli che li separano. Potremmo anzi dire che all'inizio del V° secolo D.C., ma già da prima, alcuni degli argomenti più importanti sulla natura e struttura del tempo erano stati chiaramente identificati:

- linearità o circolarità nella rappresentazione;
- finitezza o infinitezza della storia;
- apertura o chiusura degli intervalli temporali;
- discretezza o continuità della struttura temporale;
- ordinamento assoluto {passato, presente, futuro} o relativo {prima, contemporaneamente, dopo};
- oggettività o soggettività dei giudizi temporali;
- definizione di alcune modalità temporali.

Ulteriori contributi furono apportati, nel corso del XVII° e del XVIII° secolo, da Newton e da Leibnitz, anche su questo argomento attestati su posizioni opposte, e da Kant, che alla rigida contapposizione tra razionalismo ed empirismo affianca i giudizi sintetici *a priori* resi possibili grazie all'intuizione, unendo così l'esperienza del mondo reale con la necessaria universalità del puro ragionamento. I fisici del XIX° e del XX° secolo, in particolare Mach, Einstein e Prigogine, solo per citarne alcuni, hanno apportato grandi contributi alla comprensione del tempo, ma questi - lungi dal risolvere i problemi - hanno portato alla crisi i fondamenti della fisica teorica.

Lasciando ad altri la discussione sulla *natura* fisica e metafisica del tempo, nel seguito si cercherà di delineare quali siano gli strumenti, i modelli ed i problemi aperti nel campo della *generazione* e della *rappresentazione* del tempo nei sistemi informatici.

Il tempo nei sistemi digitali

I calcolatori elettronici, considerati come una categoria di sistemi digitali di grande complessità, sono fortemente sensibili ai problemi di temporizzazione. Infatti i segnali numerici (vale la pena di ricordare che in inglese "digit" significa cifra) sui quali si bassa tutto il loro funzionamento sono discretizzati sia nello spazio dei valori, ossia il numero finito di valori diversi che essi possono assumere, sia nel tempo, in quanto essi vengono misurati in istanti di tempo predeterminati.

A livello di architettura, un elaboratore elettronico può essere visto come un insieme interconnesso di unità funzionali che si scambiano dati e segnali di controllo. Per poter cooperare, queste unità devono poter sincronizzare le loro operazioni in modo che i dati giusti arrivino nel punto giusto al momento giusto. Tale sincronizzazione viene ottenuta grazie alla distribuzione di un particolare segnale - detto segnale di orologio (clock) - in tutti i punti della macchina, sia a livello di circuito integrato (chip), che di scheda (board), che di telaio (backplane).

Generalmente si suppone che il valore di un segnale all'interno di un circuito elettronico sia identico in tutti i punti di uno stesso collegamento conduttore. Tale ipotesi è giustificata dal fatto che il tempo di propagazione del segnale dal punto nel quale viene generato agli altri punti del circuito è generalmente molto inferiore al periodo di validità o alla durata dell'informazione che il segnale stesso convoglia. Le regioni dello spazio per le quali tale ipotesi è verificata vengono chiamate *regioni equipotenziale* e per esse il segnale di clock costituisce quindi un riferimento temporale **assoluto**.

Negli elaboratori delle prime generazioni, l'ampiezza della regione equipotenziale era tale da comprendere tutta la macchina e per chi, come il sottoscritto, ha incominciato a lavorare nell'Informatica una trentina di anni fa il valore del clock era una certezza. La situazione tuttavia è drammaticamente cambiata negli ultimi tempi a causa di tre fatti:

- il *progresso delle tecnologie elettroniche* ha consentito incrementi notevolissimi nella velocità di funzionamento dei componenti;
- *l'addensamento dei componenti* all'interno dei chip, ottenuto grazie alla riduzione delle loro dimensioni e della sezione delle piste conduttrici, ha avuto come effetto l'aumento della capacità e della resistività delle piste stesse, con la conseguenza che la velocità di propagazione dei segnali non è cresciuta di pari passo con quella di funzionamento dei componenti;
- *l'affinamento delle tecnologie di produzione* di componenti allo stato solido ha consentito l'aumento delle dimensioni complessive dei chip.

Il risultato della combinazione di questi fattori è che punti diversi del circuito vedono un tempo differente a causa del ritardo con il quale il segnale di clock si propaga; pertanto oggi neppure il singolo circuito integrato puo più essere considerato una regione equipotenziale. Questa situazione pone il progettista di circuiti digitali in una posizione simile a chi progetta sistemi di elaborazione distribuiti su scala macroscopica, quali le grandi reti di calcolatori; il sistema deve essere suddiviso in piccole "isole equipotenziali" che operano in modo *asincrono* tra loro, comunicando mediante uno *scambio di messaggi* che segue alcune *regole di precedenza*. Il concetto di tempo assoluto lascia quindi il campo a quello di un tempo **relativo**.

La relatività introdotta dalle regole di precedenza non è tuttavia sufficiente qualora si vogliano ottenere proprietà di prestazioni e affidabilità elevate quali, ad esempio, nei sistemi tolleranti ai guasti, nei sistemi di controllo di processi fisici e industriali, e, più in generale, nei sistemi cosidetti "in tempo reale". Sono questi ultimi sistemi che, interagendo con un mondo esterno all'elaboratore elettronico, devono soddisfare vincoli temporali (minimi e massimi) dipendenti dall'applicazione. Se infatti l'applicazione riguarda il controllo di una centrale termica, che lavora con tempi propri dell'ordine di minuti o di ore, una risposta troppo rapida da parte dell'elaboratore di controllo potrebbe non essere utilizzabile dall'impianto controllato ed andare persa; viceversa, l'elaboratore che gestisce le funzioni di pilota automatico di un velivolo deve reagire in tempi molto brevi agli stimoli di variazione dell'assetto del velivolo per evitare conseguenze infauste. Sembra oramai accertato che la causa del fallimento dell'intercettazione del missile Scud, che durante la guerra del Golfo causò la morte di decine di militari nell'accampamento di Dahran, fu una inaccuratezza nel software di temporizzazione del missile Patriot che avrebbe dovuto agganciarlo e distruggerlo.

Esistono quindi condizioni per le quali **l'assolutizzazione del tempo rimane necessaria** e deve essere ottenuta con strumenti estremamente sofisticati.

Orologi digitali e sincronizzazione

Il modo più semplice per realizzare un orologio digitale è quello di utilizzare un segnale binario, al quale corrisponde elettricamente una forma d'onda costituita da due livelli di tensione e dalle transizioni tra di essi. Una sequenza regolare di simboli **101010101 …** dà luogo pertanto ad un'onda rettangolare e si può assumere la transizione tra il livello alto e il livello basso come scatto (tick) dell'orologio; un contatore che rileva il numero di transizioni fornisce la misura del tempo trascorso. La realizzazione pratica di tale meccanismo può essere affidata a semplicissimi circuiti, tra i primi che uno studente di elettronica impara a realizzare.

Tra la realizzazione di un semplice meccanismo di conteggio e quella di un *buon* orologio corre però parecchia tecnologia. La caratteristica fondamentale di un buon orologio è infatti quella di avere una frequenza costante, ossia il tempo che intercorre tra un tick e il successivo deve essere sempre uguale. Anche se gli orologiai svizzeri hanno costruito veri prodigi di tecnologia nella misura del tempo, tuttavia la precisione ottenibile con i normali orologi è ben lontana da quella necessaria alle applicazioni informatiche più raffinate; è sufficiente una piccola variazione di temperatura per far variare la frequenza di oscillazione e quindi produrre errori nella misura, errori che, se non corretti periodicamente, si amplificano al passare del tempo.

I migliori orologi disponibili oggi, usati per mantenere i riferimenti di tempo a livello nazionale, sono basati sulle transizioni tra gli stati orbitali di un elettrone in un atomo di idrogeno e forniscono una stabilità in frequenza di $2*10^{-14}$ al giorno, con uno scostamento massimo di $1*10^{-12}$ all'anno. Gli oscillatori al quarzo usati nei sistemi informatici hanno stabilità in frequenza dell'ordine di 10^{-6} al giorno e il loro utilizzo per la generazione e la distribuzione dei segnali di temporizzazione condiziona pesantemente l'architettura dei moderni sistemi di elaborazione.

Abbiamo accennato precedentemente al fatto che nei sistemi di elaborazione distribuiti le varie parti del sistema si scambiano messaggi in modo asincrono. Più precisamente possiamo definire *distribuito* un sistema nel quale *il tempo di trasmissione dei messaggi non è trascurabile rispetto al tempo che intercorre tra gli eventi in un singolo processo*. Come si può facilmente intuire quindi la nozione di distribuzione, intesa come separazione spaziale tra le diverse unità del sistema di elaborazione, viene applicata, con il progresso della tecnologia, ad architetture molto differenti: dalle grandi reti di calcolatori ad estensione mondiale, quali ad esempio la citatissima Internet, alle reti locali che supportano architetture "client/server", ai sistemi multiprocessore e, come abbiamo visto, anche al singolo microcircuito.

Nelle reti di calcolatori, in particolare, ogni elaboratore collegato alla rete possiede un proprio clock, indipendente da tutti gli altri, la cui misura differisce più o meno marcatamente da quella degli altri. Diventa quindi necessario *sincronizzare* i tempi delle diverse macchine, dove per sincronizzazione si intende l'azione di *far sì che differenti processi in una rete di calcolatori o differenti parti di uno stesso circuito o differenti orologi presenti nel sistema concordino tutti su una stessa lettura temporale*. In Internet, ad esempio, viene utilizzata una apposita procedura - il Network Time Protocol (NTP) - per sincronizzare alcuni siti della rete (Time Synchronization Subnet) nei confronti del Tempo Coordinato Universale (UTC), determinato a livello nazionale.

Lo scopo della sincronizzazione nei sistemi di elaborazione distribuita è di garantire che le operazioni avvengano in una sequenza logicamente corretta e di consentire di stabilire implicazioni causali tra eventi che avvengono in unità computazionali distinte. Infatti la relazione ***è avvenuto prima di***, implicita in ogni computazione sequenziale, è significativa e consistente per tutto il sistema solo se gli eventi fanno riferimento ad un unico orologio, altrimenti si possono stabilire solamente degli ordinamenti parziali relativi ad uno stesso orologio. Questo tipo di considerazioni è essenziale nel gestire la concorrenza e la correttezza, anche a fronte di eventuali malfunzionamenti, di un insieme di transazioni su un sistema di elaborazione distribuito.

Il primo a sollevare tra i ricercatori informatici il problema della **dualità tra tempo assoluto e tempo relativo** è stato Leslie Lamport, forse più noto nella comunità scientifica per essere l'ideatore del sistema LATEX per la scrittura di testi scientifici e matematici. Lamport riconobbe che all'interno di un singolo sistema equipotenziale è possibile costruire un ordinamento basato solamente su *orologi logici*, ossia sull'incremento di contatori non necessariamente legati ad un tempo fisico; egli definì un algoritmo che, mediante un opportuno scambio di messaggi ciascuno dei quali riporta il valore del contatore del mittente, ottiene un ordinamento totale a partire dagli ordinamenti logici parziali di un insieme di nodi indipendenti.

L'algoritmo originale di Lamport non è tuttavia sufficiente per la sincronizzazione di applicazioni in tempo reale e tolleranti ai guasti o per impedire comportamenti anomali del sistema che potrebbero essere ottenuti scavalcando le normali procedure di funzionamento "perforando il cono di luce degli eventi", come elegantemente direbbe un fisico relativista. Più banalmente si può avere un comportamento anomalo se un operatore, immediatamente dopo aver inviato un messaggio di posta elettronica, telefona al collega nel nodo destinatario fornendogli informazioni contrastanti con quelle del messaggio elettronico e la telefonata arriva prima di quest'ultimo, in tal modo violando il principio di causalità.

Il ricorso all'uso di orologi fisici, non risolve tuttavia automaticamente tutti i problemi. Innanzitutto occorre fissare comunque dei limiti per la validità delle letture degli orologi prima di dover ricorrere ad una loro risincronizzazione; tali limiti dipendono dalla deriva della frequenza dei singoli orologi, dallo scarto massimo ammissibile tra le letture istantanee di due orologi, dal più breve tempo di trasmissione tra due processi interagenti. Inoltre occorre tener conto della possibilità di guasto degli stessi orologi o dei processi coinvolti nella loro sincronizzazione. Sono stati pertanto definiti algoritmi che forniscono il minimo numero di orologi necessari (e sufficienti) a garantire comunque la sincronizzazione anche in presenza di un numero *n* di orologi che funzionano in modo imprevedibile (guasti bizantini), o che fissano il numero di messaggi che occorre scambiare tra i diversi partner per arrivare alla sincronizzazione. Dato che l'argomento è particolarmente complesso, di questi algoritmi sono state date prove formali e automatiche di correttezza, che talvolta hanno rivelato alcuni errori.

Riguardando a quanto esposto finora da un punto di vista ontologico, possiamo dire che nella generazione e disseminazione dei riferimenti temporali in un sistema di elaborazione vengono coinvolti i seguenti problemi:

- gli **orologi fisici sono continui** mentre gli **orologi logici sono discreti**; per essi occorre quindi scegliere una *granularità* che soddisfi i vincoli posti dalle applicazioni;
- la **dualita tra tempo assoluto e tempo relativo** viene risolta a favore del primo, ma al prezzo di stabilire complessi protocolli di comunicazione per sincronizzare tra loro gli orologi locali;
- il tempo assoluto viene utilizzato laddove sia necessario stabilire **nessi causali** tra vari processi del sistema e, in particolare, per assicurarne l'affidabilità.

La rappresentazione del tempo nei sistemi informativi

Dopo aver parlato dei modi usati per *materializzare* il tempo nei sistemi di elaborazione, passiamo ora a considerare come esso venga *rappresentato* nei sistemi informativi e, più in generale, nelle applicazioni informatiche. Per far ciò dobbiamo tornare alla metà degli anni quaranta quando i filosofi, dopo secoli di abbandono, ricominciarono a considerare il tempo come oggetto di studio, in particolare nell'ambito delle scienze cognitive e della linguistica.

I primi approcci al problema del ragionamento temporale, fatti da Russel e da Quine, hanno considerato il tempo come una qualsiasi variabile del calcolo dei predicati del primo ordine. Prior fu il primo a considerare la *logica temporale* - meglio sarebbe dire la logica dei tempi verbali (*tense logic*) - come un particolare tipo di logica modale[1]. Le due diverse posizioni hanno suscitato tra i logici accessi dibattiti sulla effettiva necessità di utilizzare logiche non standard per modellare i ragionamenti che coinvolgono il tempo. Gli autori di scuola fisico-matematica (Russell, Quine, Allen, McDermott, Kowalski e altri) propendono per una risposta negativa e vengono pertanto indicati come *detenser* (detemporizzatori), gli studiosi di linguistica (Prior, Von Wright) attribuiscono invece grande importanza alle modalità temporali insite nei tempi verbali tanto che Reichenbach arriva a definire tre istanti diversi per ogni tempo verbale: l'istante nel quale la *frase viene pronunciata*; un tempo di *riferimento*, al quale ci si riferisce all'interno della frase; un *tempo dell'evento*, che è l'oggetto della frase stessa[2]. Senza entrare in dettagli si può comunque dire che i due approcci, pur diversi tecnicamente, non sono tra loro in contrasto.

Gli studiosi di Informatica hanno cominciato ad interessarsi delle logiche temporali sia per la loro capacità di modellare la conoscenza temporale nelle applicazioni di Ingegneria della conoscenza (sistemi esperti, robotica, ecc.), che per la loro natura di formalismo rigoroso per la specifica e la verifica di sistemi concorrenti in tempo reale. Per citare una frase di Galton "… nel primo caso la logica temporale è utilizzata per consentire ai programmi in un calcolatore di ragionare sul mondo esterno; nel secondo caso essa permette al mondo esterno di ragionare sui programmi in un calcolatore …". In entrambi i casi le questioni ontologiche relative alla natura del tempo e delle altre grandezze ad esso collegate sono comuni; le scelte fatte per realizzare i rispettivi modelli variano invece a seconda della particolare applicazione. Passiamo ora in rassegna le varie alternative.

<u>Entità temporali primitive</u>

Le entità temporali da considerare come primitive, i mattoni cioè con i quali costruire i modelli, costituiscono uno dei fondamentali elementi di scelta e danno luogo al classico problema se sia nato prima l'uovo o la gallina. Le opzioni possibili sono:

- gli *istanti* (punti) di tempo;
- gli *intervalli* (segmenti) di tempo;
- gli *eventi* che accadono nel tempo.

Ovviamente ognuno di questi concetti può essere ricondotto agli altri, ma la scelta non è banale per quanto riguarda la costruzione dei modelli applicativi. In particolare gli eventi vengono considerati in modo diverso a seconda che si assuma una posizione temporale *assolutistica* o *relativistica*: nel primo caso il tempo precede ontologicamente gli eventi dato che esso è visto come un contenitore di eventi e viene utilizzato per distinguerli; nel secondo sono gli eventi a precedere ontologicamente il tempo poichè sono proprio essi a identificare gli istanti di tempo! A causa di questa ambiguità e del fatto che gli eventi sono entità spazio-temporali, la loro scelta come entità temporali primitive risulta problematica.

<u>Topologia del tempo</u>

Dato che il tempo ha una natura dinamica, sorge spontaneo di definire delle *traiettorie temporali* in modo da poter applicare una relazione d'ordine (prima-di, dopo-di) tra due qualsiasi entità temporali. Anche qui si aprono due possibilità: la prima considera un *ordinamento totale,* tale cioè che sia sempre possibile per *qualsiasi coppia* di entità dire quale è avvenuta prima; le entità possono quindi essere mappate su di una retta di lunghezza infinita, dando luogo al cosiddetto tempo *lineare,* oppure su di un cerchio, dando così luogo al tempo *periodico*. La seconda ipotesi considera invece solamente *ordinamenti parziali*, tali cioè che ogni entità possa essere immediatamente preceduta (*ramificazione nel passato*) o seguita (*ramificazione nel futuro*) da molte altre entità.

Il modello lineare è quello standard delle scienze naturali, in primo luogo della fisica, e della concezione giudaico-cristiana del divenire del mondo mentre un modello costituito da un passato lineare e un futuro ramificato ha ottenuto una notevole fortuna presso i ricercatori informatici in quanto esso accoppia la descrizione di un passato deterministico (ciò che è accaduto è noto) a quella di un futuro aperto (tutte le alternative possibili da un certo istante in avanti).

<u>Relazioni temporali</u>

La semplice relazione di precedenza, che è sufficiente a definire la relazione tra due elementi di tempo istantanei, non basta più se si considerano come elementi primitivi gli intervalli. In questo caso occorre definire ben 13 relazioni diverse tra due intervalli di tempo *x* e *y*: *prima-di(xy), incontra(xy), sovrapposto(xy), durante(xy), all'-inizio(xy), alla-fine(xy)*, le loro 6 relazioni inverse e la relazione *coincide(xy)*.

<u>Limitatezza</u>

Una questione importante riguarda il fatto che ogni entità temporale abbia un *successore* (*predecessore*) oppure se esista un *istante finale* (*iniziale*) del tempo. Se gli istanti iniziale e finale non esistono, la struttura del tempo viene detta *omogenea* e ogni posizione temporale è equivalente alle altre. Ciò rende possibile comprimere o dilatare i tempi come accade nelle simulazioni al calcolatore dove il tempo reale degli eventi viene compresso, sulla retta temporale, in un tempo di simulazione molto più breve.

Se parliamo di intervalli di tempo, la questione della limitatezza equivale a chiedersi se essi siano aperti o chiusi ad una o ad entrambe le estremità. Il problema è particolarmente rilevante per la relazione *incontra(xy)*; in essa il punto di incontro dei due intervalli costituisce un istante di separazione tra il periodo nel quale una certa variabile logica assume il valore vero e il periodo nel quale assume il valore falso. Ad esempio, nel primo periodo la luce nella stanza è accesa, nel secondo è spenta. In linea di massima, per i sistemi fisici reali bisogna *escludere* i modelli per i quali gli intervalli sono *aperti o chiusi ad entrambe le estremità* poichè questo violerebbe il principio di *non contraddizione* (potrebbe essere vero che la luce sia contemporaneamente accesa e spenta) e quello del *terzo escluso* (la luce deve essere o accesa o spenta).

<u>Struttura del tempo</u>

Il problema precedente è strettamente collegato alla struttura usata nella rappresentazione del tempo. Molti autori ritengono che il tempo sia essenzialmente *continuo*, ossia sia rappresentabile con numeri reali; altri ritengono che un tempo *denso*, ossia in corrispondenza con i numeri razionali, sia più che sufficiente a rappresentare tutte le situazioni della vita reale. Infine le applicazioni all'Ingegneria del Software propendono per una struttura *discreta* del tempo, corrispondente ai numeri interi. Si noti che la scelta di un tempo denso o discreto elimina il problema dell'istante di separazione visto prima.

<u>Metriche temporali</u>

Nelle applicazioni informatiche, sia di Intelligenza Artificiale che di Ingegneria del Software, le sole caratteristiche logiche e topologiche non sono sufficienti a modellare tutte le proprietà dei sistemi reali. Devono pertanto essere introdotti nei sistemi temporali alcuni concetti *quantitativi* che li trasformino in sistemi *cronologici*. Ciò viene fatto introducendo la nozione di *distanza* tra due elementi di tempo; si ottiene così uno *spazio metrico* nel quale è possibile correlare qualsiasi elemento temporale con un numero che rappresenta la sua distanza da un unico *elemento di riferimento*. In genere tale elemento viene identificato con il *tempo presente* e si stabilisce un ordinamento lineare simmetrico verso il passato e verso il futuro.

L'introduzione di una metrica per il tempo comporta alcuni problemi per l'utilizzo di modelli con tempo ramificato; questi sono dovuti soprattutto al fatto che non è possibile assicurare che gli orologi che misurano il tempo nei diversi rami si comportino tutti allo stesso modo e quindi non si può garantire la confrontabilità delle misure (lascio quì libera la fantasia degli appassionati di fantascienza!).

Un sistema metrico richiede la definizione di unità di misura con una *granularità* diversa a seconda delle applicazioni: anno, mese, giorno, minuto, …; inoltre è possibile usare *locuzioni temporalmente stabili*, quali le date (p.e. "il 4 febbraio 1997") oppure *locuzioni temporalmente instabili*, quali le pseudo-date (p.e. "dopodomani"), che richiamano nuovamente l'eterna dualità tra tempo assoluto e tempo relativo.

Infine è l'introduzione della metrica che consente di dare una chiara visione dei fenomeni periodici, svolgendo i cicli in un insieme di intervalli di uguale lunghezza, che si susseguono sulla retta del tempo.

Applicazioni ai sistemi deduttivi e di pianificazione

Se la comprensione del linguaggio naturale è stata l'area applicativa che ha risvegliato l'interesse dei ricercatori rispetto alla rapresentazione del tempo, gli approcci più fruttuosi nella pratica sono emersi considerando la possibilità di costruire sistemi artificiali che fossero in grado di "ragionare" su fatti ed eventi temporali. I dominii applicativi interessati a questa caratteristica spaziano dai sistemi di pianificazione delle attività, ai sistemi di supporto alle decisioni, alla diagnosi dei guasti in macchine e impianti, ai sistemi di automazione degli uffici, alle basi di dati temporali e deduttive.

L'essenza di tutte queste applicazioni sta nel poter stabilire le *proprietà* di alcuni oggetti che sono *valide* in alcuni intervalli di tempo, definiti dagli *eventi* che in essi avvengono e dai *processi* che in essi operano.

Dal punto di vista ontologico, l'unico aspetto sul quale la maggior parte degli studiosi concorda è sull'uso del calcolo dei predicati del primo ordine, mentre viene avversata (anche se con alcune eccezioni) l'introduzione di logiche modali. Su tutto il resto - istanti, intervalli o eventi; struttura discreta o continua; topologia lineare o ramificata - il dibattito si è acceso, con toni da guerra di religione, in favore di questo o quel modello, senza talvolta rendersi conto che deve essere l'applicazione che guida la scelta del modello senza posizioni preconcette.

Un ulterirore elemento di controversia tra i ricercatori, che peraltro si trascina dai tempi dei filosofi greci, sta nella possibilità di stabilire collegamenti *causa-effetto* a partire da relazioni di precedenza temporale; basta dare uno sguardo alla letteratura scientifica nel settore per rendersi conto che il dibattito proseguirà vivace anche nell'ormai vicino prossimo millennio!

Come ultimo esempio di complessità della gestione del tempo nei sistemi informativi, si può citare quello delle basi di dati temporali. In questi sistemi, contrariamente a quanto accade nelle basi di dati classiche - che fotografano la situazione istantanea dei dati cancellando la storia passata attraverso la quale si è pervenuti al valore attuale del dato - si vuole mantenere tutta la storia degli aggiornamenti e, in taluni casi, può essere necessario aggiornare retroattivamente il valore di un dato nel passato sulla base di quanto appreso in un momento successivo. Si devono pertanto definire tre diverse dimensioni temporali: il *tempo della transazione*, quando il dato è stato introdotto nella base di dati; il *tempo di validità*, quando il fatto modellato era valido; il *tempo dell'interrogazione*, quando si va a considerare il contenuto della base di dati. E' riconoscibile in questa triade temporale un modello simile a quello definito da Reichenbach, e già citato, per l'analisi dei tempi verbali.

Applicazioni all'ingegneria del software

L'interesse degli ingegneri del software per i problemi di temporizzazione è scaturito nell'ambito delle tecniche per la verifica formale dei programmi, quando ci si è resi conto che i metodi usati con successo per i programmi sequenziali non andavano bene per i programmi *concorrenti* a causa delle possibili interferenze tra i relativi processi o tra questi ed eventi esterni. Ben presto lo studio è stato allargato a quei sistemi che vengono definiti *reattivi,* quelli cioè che mantengono una interazione continua con l'ambiente e che in alcuni casi, come quello dei sistemi operativi e quello del controllo in tempo reale di processi industriali, non hanno un proprio punto di terminazione.

Le tecniche di verifica tradizionali devono, in questi casi, essere ricondotte alla prova di nuove classi di proprietà:

- *invarianza*: proprietà di un programma che vale con continuità durante la sua esecuzione; essa assicura che *non succederà nulla di male* (per tale motivo viene anche definita *safety*). Essa viene espressa con la clausola modale di necessità **always** (sempre);
- *eventualità*: proprietà che garantisce che in qualche momento verrà fatta qualche cosa che deve essere fatta; essa assicura che *accadrà qualche cosa di buono* (viene pertanto chiamata *liveness*). Essa viene espressa mediante la clausola modale di possibilità **sometimes** (qualche volta);
- *precedenza*: proprietà che stabilisce *l'ordine degli eventi*. Viene espressa mediante l'operatore **until** (fino a chè).

Altri operatori temporali - quali ad esempio **next** (il prossimo), **unless** (a meno che), **since** (fin da) - possono essere ricondotti ai tre precedenti. Una distinzione particolare è stata fatta per gli operatori **sometimes** e **not never** (non mai) poichè il loro significato coincide in una topologia lineare del tempo, ma differisce se il tempo è ramificato. Inoltre, poichè la prova delle proprietà di un programma richiede di ragionare a partire da un certo istante in avanti (tutto ciò che è passato è noto fino a quel punto), una logica temporale limitata al futuro e che comprenda l'operatore **until** è l'unico strumento *necessario* per le applicazioni in questo settore; tutti gli altri operatori risultano utili, ma ridondanti.

Le scelte ontologiche differiscono da quelle fatte per il trattamento della conoscenza nella preferenza data alle *logiche modali* e nell'utilizzo sia del *calcolo proposizionale* (per la verifica di programmi concorrenti) che del *calcolo dei predicati del primo ordine* (nei linguaggi di specifica per sistemi in tempo reale). Per i sistemi in tempo reale è anche necessaria l'introduzione di opportune *metriche* mentre per le verifiche di concorrenza è sufficiente il semplice concetto di ordinamento relativo.

Le altre scelte ontologiche premiano la *discretezza* (tranne che per il controllo di sistemi continui) e gli *istanti o eventi istantanei* come entità fondamentali. Quanto alla topologia si può dire che se la scelta di un tempo ramificato consente di costruire modelli molto potenti, tuttavia per la maggior parte delle applicazioni reali risulta sufficiente e molto più semplice la scelta di un tempo lineare. Per descrivere dettagliatamente fenomeni periodici che si protraggono nel tempo è possibile utilizzare anche una topologia elicoidale, come è stato fatto in qualche linguaggio di specifica per sistemi in tempo reale.

Conclusioni

Ci siamo chiesti nel titolo se "il tempo dell'Informatica sia un tempo reale". A parte il gioco di parole, potremmo rispondere con Sant'Agostino "… so abbastanza bene che cosa sia il tempo, purchè nessuno me lo chieda; ma se me lo chiedono e cerco di spiegarlo rimango confuso". Per fortuna i problemi affrontati dagli informatici non richiedono risposte metafisiche sulla natura del tempo, ma hanno bisogno solamente di un insieme di strumenti pragmatici che consentano loro di realizzare architetture e applicazioni. Da questo punto di vista le necessità dei progettisti di hardware e di software non differiscono molto e possiamo quindi concludere che sì, nell'ambito di ciascun campo di applicazione, il modello del tempo utilizzato è quello di un tempo reale!

Bibliografia

Galton A. - *Temporal Logics and their applications* - Academic Press, 1987

Pizzi C. (a cura di) - *La logica del tempo* - Boringhieri, 1974

Raju C.K. - *Time: Towards a Consistent Theory* - Kluwer, 1994

Rescher N., Urquhart A. - *Temporal Logic* - Springer-Verlag, 1971

Schreiber F.A. - *Is time a real time? An overview of time ontology in Informatics* - in *Real Time Computing*, ASI Vol. F27, Springer-Verlag, 1994

Van Benthem J.F.A.K. - *The Logic of Time* - Reidel, Dordrecht 1983 (2° edizione 1991)

---

[1]LOGICA FORMALE: scopo della logica è di formulare regole per condurre argomentazioni valide la cui conclusione discende necessariamente dalle premesse. Tali regole possono riguardare diversi livelli di complessità; in questo articolo interessano in particolare i seguenti: CALCOLO DELLE PROPOSIZIONI - consente di valutare se una affermazione (proposizione) complessa è VERA o FALSA, in base alla verità o falsità dei suoi componenti. Coincide essenzialmente con l'algebra di Boole, fondamento dei circuiti logici degli elaboratori elettronici. CALCOLO DEI PREDICATI - consente di verificare la veritào la falsità di una asserzione contenente delle variabili, in dipendenza del valore da queste assunto. Un esempio di tali asserzioni sono i sillogismi. I linguaggi basati sul calcolo dei predicati comprendono, come caso particolare, quelli basati sul calcolo delle proposizioni. LOGICHE MODALI - permettono di ragionare su affermazioni che comprendono operatori quali necessario/possibile, obbligatorio/permesso, passato/futuro, ecc.

[2] Reichenbach si è occupato dell'analisi dei tempi verbali riuscendo a spiegare formalmente la differenza tra passato prossimo e passato remoto, tra futuro e futuro anteriore, ecc. (H. Reichenbach - Elements of Symbolic Logic - The Free Press, New York 1966 (ristampa dal 1947))