Un modello di sviluppo di sistemi informativi

Carlo.Meghini@isti.cnr.it
Paola.Venerosi@isti.cnr.it
Agosto 2003

Fase I, parte II: La relazione causale e il calcolo degli eventi

Rationale

La nota intende indagare sull'applicabilita' del formalismo del calcolo degli eventi al modello di sistema informativo ipotizzato in una precedente nota [1].
La procedura adottata e' diretta (i) ad acquisire la conoscenza delle capacita' espressive del formalismo e (ii) verificare la loro rispondenza alle esigenze poste dalla tipologia dei sistemi informativi precedentemente indicata. La verifica viene condotta per esemplificazione.
Il focus della nota punta inoltre a sondare, in linea teorica, la fattibilita' di una integrazione tra le assunzioni concettuali fatte nella nota precedente ed il formalismo del calcolo degli eventi. Il ragionamento muove dall'ipotesi che si possa, partendo dalle tesi portate alla luce dall'analisi filosofico-linguistica di Davidson[2] e dalle argomentazioni prodotte a loro sostegno, o a confutazione di tesi contrapposte, giungere ad una piu' profonda comprensione del formalismo di Shanahan[3], finalizzato, invece, al calcolo. La questione viene impostata a partire dalla flessibilita' dell'ontologia, la sua capacita' di specializzarsi e dalla possibilita' di estendere il formalismo, cosa per altro gia' sperimentata nell'ambito dei linguaggi naturali.

Il calcolo degli eventi

Il "calcolo degli eventi" e' stato sviluppato da Kowalski and Sergot nel 1986 [4] come formalismo della programmazione logica per applicazioni di basi di dati che richiedono inferenze su eventi e azioni, in alternativa al "calcolo situazionista" di McCarthy and Hayes, 1969 [5].
La solida base teorica sviluppata intorno al problema di come rappresentare le azioni ed i loro effetti, ha fatto del calcolo degli eventi uno strumento appropriato a modellare una varieta' di fenomeni come le azioni con effetti indiretti, azioni con effetti non deterministici, azioni composte, azioni concorrenti, cambiamenti continui. L'ontologia messa a punto permette di distinguere diversi tipi di eventi e si e' rivelata utile per risolvere alcuni problemi di semantica dei linguaggi naturali (trattamento formale di espressioni di *nominalization*) [6].

Il calcolo degli eventi di Shanahan

Il calcolo degli eventi con *circumscription* di Shanahan del 1997 [7] si basa su un meccanismo logico che permette di inferire quando qualcosa e' vero, se e' noto cosa accade e quando, e quello che gli accadimenti determinano. "Cosa accade e quando" rappresenta *la narrativa* degli eventi; "quello che gli accadimenti determinano" rappresenta *gli effetti* delle azioni, descritti dalle leggi causali operanti. Per es., se il mangiare mi rende felice (effetto) ed io mangio alle 12 (narrativa), il calcolo degli eventi conclude che, in assenza di eventi turbativi, io sono felice alle 12.05.
L'ontologia del calcolo degli eventi indica i tipi di oggetti sui quali e' ammessa la quantificazione e da' luogo a un linguaggio del primo ordine che rappresenta le azioni ed i loro effetti. L'ontologia di base del calcolo degli eventi proposta da Shanahan comprende: *azioni, eventi (o piuttosto tipi di azione ed eventi), fluenti e time points*. Il fluente e' qualsiasi cosa il cui valore nel tempo e' soggetto a cambiamento. Puo' essere una quantita' come "la temperatura della stanza", oppure una proposizione come "sta piovendo".
I predicati che legano gli elementi dell'ontologia sono scelti per esprimere cosa accade e quando, per descrivere la situazione iniziale, gli effetti delle azioni, e per dire quali fluenti sono presenti e quando.

Il calcolo degli eventi di Shanahan e' in realta' una famiglia di tre calcoli, di complessita' crescente: il calcolo base, il calcolo completo, e il calcolo esteso. Ciascuno di questi estende il precedente, nell'ordine appena dato.

Il calcolo base degli eventi

I simboli di predicato che di seguito indichiamo, formano il formalismo cosiddetto *di base*.
I primi quattro simboli sono preposti al cambiamento normale, quando cioe' la proprieta' dipendente dal tempo e' interrotta o avviata da un evento. Per es. quando il fluente (acqua che scorre) inizia (termina) con l'aprire (chiudere) un rubinetto. Queste primitive si adattano bene ai casi in cui i fluenti hanno un termine.

*$Initially_P(f)$*indica che il fluente *f* e' presente dall'inizio della storia considerata
Happens(a, t) denota che l'azione *a* avviene al tempo *t*.
Initiates(a, f, t) dice che il fluente *f* inizia ad esistere dopo l'azione *a*
Terminates(a, f, t) dice che *f* non persiste dopo l'occorrenza di *a* al tempo *t*.

I seguenti tre simboli di predicato rappresentano in ordine la verita', la relazione temporale di precedenza tra istanti, ed una relazione definita in termini di processo;

HoldsAt(f, t) dice che il fluente *f* e' presente al tempo *t*
t1 < t2 dice che il punto *t1* precede il punto *t2*
Clipped(t1, f, t2) dice che il fluente *f* e' terminato tra il tempo *t1* e il tempo *t2*

Di seguito, tre assiomi (vincoli) che legano i predicati sin qui introdotti e la cui congiunzione e' denominata SC da Shanahan , :
(SC1) HoldsAt(f, t) ← Initiallyp(f) ∧¬ Clipped(0, f, t)
dice che il fluente f esiste al tempo t se e' esistito anche al tempo 0 e non e' terminato tra il tempo 0 ed il tempo t
(SC2) HoldsAt(f, t2) ←
Happens(a, t1) ∧ Initiates(a, f, t1) ∧ t1< t2 ∧ ¬ Clipped(t1, f, t2)
Dice che il fluente f e' presente al tempo t2, se c'e' stato un accadimento a al tempo t1 anteriore a t2 che lo ha fatto iniziare, e se il fluente non e' terminato tra i tempi t1 e t2.
L'assioma seguente specifica che il fluente f e' terminato tra i tempi t1 e t2 se e solo se c'e' stato un'accadimento al tempo t che ha fatto terminare il fluente, e t si trova tra t1 e t2.
(SC3) Clipped(t1, f, t2) ↔
∃ a, t [Happens(a,t) ∧ t1< t < t2 ∧ Terminates(a, f, t)]
Come si vede l'eseguibilita' del calcolo e' logicamente relata al formalismo temporale.

The Frame Problem

Il meccanismo logico di base non e' tuttavia sufficiente a risolvere il cosiddetto frame problem, che pone la necessita' di rappresentare con la stessa logica non solo gli effetti di una azione, ma anche i suoi non-effetti. Il problema preso in seria considerazione dal 1969, e' esemplificato nello Yale shooting scenario di Hanks & McDermott, 1987 [8] del quale Shanahan propone una versione:

Siano Load(Caricare), Sneeze(Starnutire), Shoot(Sparare) le azioni e Loaded(stato di arma carica), Alive(stato di Vita), Dead(stato di Morte) i fluenti.
L'effetto dell'azione Caricare e' di produrre lo Stato di arma carica; l'azione di Sparo implica che, qualora si abbia lo stato di arma carica, tale stato termini, abbia inizio lo Stato di Morte e quindi termini lo Stato di Vita, a condizione che lo Starnuto non abbia avuto effetti.

Lo ***Yale shooting scenario*** prevede una precisa sequenza delle azioni: *Load, Sneezee, Shoot*.
Se rappresentiamo quanto detto sopra nel modo seguente:

Initiates(Load, Loaded, t)
Initiates(Shoot, Dead, t) ← HoldsAt(Loaded, t)
Terminates(Shoot, Alive, t) ← HoldsAt(Loaded, t)
per codificare le leggi causali

Initiallyp(Alive)
Happens(Load, T1)

Happens(Sneeze, T2)
Happens(Shoot, T3)
T1 < T2 < T3 < T4
per rappresentare la sequenza temporale delle azioni

ed indichiamo con Σ la congiunzione dei predicati Initiates e Terminates; Δ la congiunzione di Initiallyp, Happens e le formule temporali; SC la congiunzione degli assiomi che riguardano i vari predicati, non otteniamo pero' quel che era nelle nostre intenzioni:

$\Sigma \wedge \Delta \ \wedge SC \models$ HoldsAt(Dead, T4).

Non otteniamo il risultato desiderato perche' ci sono dei modelli di $\Sigma \wedge \Delta \wedge SC$ in cui lo Starnutire fa terminare il fluente Loaded, e quindi in questi modelli, lo Sparare al tempo T3 non provoca la morte al tempo successivo. La causa va ricercata nella mancata descrizione dei non effetti delle azioni; in particolare, nella mancata dichiarazione che lo Starnutire non termina il fluente Loaded. Inoltre e' necessario descrivere anche la non occorrenza di azioni, per bandire modelli in cui sia vero piu' di cio' che deve. Infine, c'e' il problema di fare in modo che simboli diversi (per esempio Loaded e Dead) denotino cose diverse.

Shanahan indica alcune soluzioni per ovviare a questi problemi:
L'inclusione di un insieme di *uniqueness-of-names axioms* per i fluenti e per le azioni, usando la notazione di Baker del 1991 [9]. Nello scenario proposto:

UNA[Load, Sneeze, Shoot] per le azioni e UNA[Loaded, Alive, Dead] per i fluenti.

In questo modo si chiarisce che Sneeze e' cosa diversa da Load, e Loaded da Alive, nel senso che hanno implicazioni diverse.

I non-effetti delle azioni e la non-occorrenza degli eventi si possono descrivere implicitamente per mezzo di quella che l'autore chiama *completion* dei predicati *Initiates, Terminates, Happens.*

Nel caso di Initiates(a, f, t) si rende esplicito che esso vale solo nel caso in cui avvenga una specifica serie di azioni. Per es. Initiates(a, f, t) $\leftrightarrow$[a = Load $\wedge$ f = Loaded] $\vee$ [a = Shoot $\wedge$ f = Dead $\wedge$ HoldsAt(Loaded, t)]. Lo stesso vale per Terminates(a, f, t).
Happens(a,t) indica l'avverarsi di una azione e deve esprimere, secondo una formula temporale ordinata, la successione delle azioni. Happens(a, t) $\leftrightarrow$ [a = Load $\wedge$ t = T1]$\vee$ [a = Sneeze $\wedge$ t = T2]$\vee$ [a = Shoot $\wedge$ t = T3]; T1< T2, T2 < T3, T3 < T4.

La formula della circumscription
Per permettere la costruzione automatica della *completion* dei predicati, e cosi' ottenere *la common sense law of inertia*, cioe' "l'assunzione che un fluente persiste fino a quando non c'e' una ragione per credere il contrario", Shanahan propone di adottare la *circumscription* di McCarthy, 1980 [10], e Shanahan, 1997 [7], basata sull'idea di minimizzare l'estensione di certi predicati. La *circumscription* di una formula Φ produce una teoria nella quale uno specificato insieme di simboli di predicato ha la piu' piccola estensione ammessa in conformita' con.Φ.
CIRC[Φ; ρ] indica la *circumscription* di Φ che minimizza il predicato ρ.

Quindi se Σ e' la congiunzione dei predicati Initiates e Terminates; Δ la congiunzione di Initiallyp, Happens e le formule temporali; Ω la congiunzione degli assiomi per la univocita' dei nomi per fluenti ed azioni; SC la congiunzione degli assiomi che riguardano i vari predicati, la formulazione nel suo complesso con l'uso della *circumscription* (di Φ che minimizza tuple di predicati ρ) sara':

CIRC[Σ ;Initiates, Terminates] $\wedge$CIRC[Δ ; Happens] $\wedge$ SC $\wedge$ Ω

da cui logicamente segue HoldsAt(Dead, T4), come volevamo ottenere.

La ‘minimizzazione’ dei predicati *Initiates* e *Terminates* serve ad esplicitare l’assunzione che non vi saranno effetti inaspettati; con la ‘minimizzazione’ di *Happens* che non ci saranno occorrenze di eventi non previste.

Il calcolo degli eventi nella versione completa

Il formalismo pieno estende la versione base come segue:

- include tre assiomi che descrivono quando un fluente non e’ presente. I nuovi predicati *InitiallyN* e *Declipped* vengono introdotti come controparti negative di *InitiallyP* e *Clipped*
- incorpora una versione di *Happens* a tre argomenti, che permette di rappresentare sia la durata di un’azione che le azioni composte.
- incorpora il nuovo predicato *Releases*, che disabilita la *common sense law of inertia*.

Il formalismo pieno e’ adatto a quei domini che coinvolgono azioni con effetti indiretti e non deterministici.

Initially$_N$ (f) Il fluente *f* non e’ presente al tempo iniziale *t0*
Declipped(t1, f, t2) Il fluente *f* e’ iniziato tra il tempo *t1* e *t2*
Releases(a, f, t) Il fluente *f* non e’ soggetto ad inerzia dopo l’azione *a* al tempo *t*
Happens(*a, t1,t2*) L’azione *a* inizia al tempo *t1* e finisce al tempo *t2*

Un gruppo di assiomi, la cui congiunzione e’ chiamata da Shanahan EC, definisce l’uso dei predicati :

(EC1) HoldsAt(f, t) $\leftarrow$ Initiallyp(f) $\wedge \neg$ Clipped(0,f,t)

Se il fluente f e’ presente dal tempo iniziale t0 e non e’ terminato tra i tempi 0 e t, f e’ presente al tempo t

(EC2) HoldsAt(f,t3) $\leftarrow$
Happens(a,t1,t2) $\wedge$ Initiates(a,f,t1) $\wedge$
T2 < t3 $\wedge \neg$ Clipped(t1,f,t3)

Se al tempo t1 accade un’azione che termina al tempo t2 e che fa iniziare il fluente f, e se f non termina tra t1 e t3, dove t3 viene dopo di t2, allora il fluente f e’ attivo al tempo t3

(EC3) Clipped(t1,f,t4) $\leftrightarrow$
$\exists$ a,t2,t3 [Happens(a,t2,t3) $\wedge$ t1 < t3 $\wedge$ t2 <t4 $\wedge$
[Terminates(a,f,t2) $\vee$ Releases(a,f,t2)]]

Il fluente f termina tra t1 e t4 se e solo se esiste un’azione che inizia al tempo t2 e termina al tempo t3, e t1 viene prima di t3 e t2 viene prima di t4 e o il fluente termina dopo l’occorrenza dell’azione al tempo t2 oppure il fluente non e’ soggetto alla legge di inerzia dopo l’azione occorsa al tempo t2

(EC4) $\neg$ HoldsAt(f,t) $\leftarrow$ InitiallyN(f) $\wedge \neg$ Declipped(0,f,t)

Il fluente f non e’ presente al tempo t se non esisteva al tempo iniziale t0 e non e’ iniziato tra t0 e t

(EC5) $\neg$ HoldsAt(f,t3) $\leftarrow$
Happens(a,t1,t2) $\wedge$ Terminates(a,f,t1) $\wedge$
T2 <t3 $\wedge \neg$ Declipped(t1,f,t3)

Il fluente f non e’ presente al tempo t3 se e’ occorsa un’azione, durata da t1 a t2, che ha terminato f,e tra t1 e t3 f non e’ stato iniziato, con t3 successivo a t2

(EC6) Declipped(t1,f,t4) $\leftrightarrow$
$\exists$ a,t2,t3 [Happens(a,t2,t3) $\wedge$ t1 < t3 $\wedge$ t2 <t4 $\wedge$
[Initiates(a,f,t2) $\vee$ Releases(a,f,t2)]]

Un fluente inizia tra t1 e t4 se e solo se c’e’ un’azione tra t2 e t3 che ha iniziato il fluente oppure che ha sottratto il fluente alla legge d’inerzia , con t1 < t2 < t3 < t4.

(EC7) Happens(a,t1,t2) $\rightarrow$ t1 $\leq$ t2

Un’azione che occorra tra il tempo t1 e t2 implica che t1 sia non successivo a t2

Il simbolo di predicato Happens(a,t) e’ ora definito come materialmente equivalente alla sua versione con tre argomenti Happens(a,t,t).

Se ogni asserzione fatta sul simbolo Releases risulta sempre falsa, allora dagli assiomi EC1-EC7 si deducono gli assiomi SC1-SC3, ovvero: se tutti i fluenti sono sempre soggetti alla legge di inerzia, gli assiomi del calcolo completo implicano quelli del calcolo di base.Il frame problem viene superato con le stesse modalita’del calcolo degli eventi di base: il problema viene rappresentato dalla formula:

$CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta ;Happens] \wedge EC \wedge \Omega$

dove Σ e' la congiunzione delle formule Initiates, Terminates e Releases; Δ e' la congiunzione delle formule Initiallyp, InitialliN, Happens e dell'ordinamento temporale; Ω la congiunzione degli assiomi relativi all'unicita' dei nomi per azioni ed eventi.

L'uso del calcolo completo

Per spiegare come funziona il calcolo completo Shanahan ricorre al giuoco della roulette russa, che ha come incognita il fatto di non sapere se l'azione di *spin* (rotazione del caricatore della pistola) porra' il caricatore stesso in condizione di sparare. Questa situazione di indeterminatezza viene rappresentata dal formalismo tramite il predicato *Releases* che consente di ragionare su quando i fluenti possono o meno attivarsi.

Fissate le leggi causali:

Initiates(Load, Loaded, t)
Terminates(Shoot, Alive, t) $\leftarrow$ HoldsAt(Loaded, t)
Releases(Spin,Loaded, t)

Specificata la narrativa:

Initiallyp(Alive)
Happens(Load, T1)
Happens(Spin, T2)
Happens(Shoot, T3)
$T1< T2, T2 < T3, T3 < T4$

Applicati gli assiomi per l'unicita' dei nomi

UNA[Load,Spin, Shoot]
UNA[Loaded, Alive]

e la nota formula della *circumscription*,
otteniamo:

$CIRC[\Sigma; Initiates, Terminates, Releases] \wedge$
$CIRC[\Delta ; Happens] \wedge EC \wedge \Omega \models HoldsAt(Alive, T4)$

oppure

$CIRC[\Sigma ; Initiates, Terminates, Releases] \wedge$
$CIRC[\Delta ; Happens] EC \wedge \Omega \models \neg HoldsAt(Alive, T4)$

C'e uno stato di vita al tempo t0, e al tempo t1 e' accaduta un'azione di Load. Per le leggi causali, cio' fa si' che al tempo t1 sia iniziato uno stato di arma carica, rappresentato dalla presenza del fluente Loaded.
Al tempo t2, quando per inerzia lo stato di arma carica continuava ad essere presente, si e' verificata un'azione di *spin* che ha disabilitato la legge di inerzia del fluente Loaded (Releases(Spin, Loaded, t)), per cui al tempo t3, eseguita l'azione di sparo, si potevano avere due situazioni: l'inizio di uno stato di morte al tempo t4, qualora la pallottola fosse stata ben impostata (in direzione della canna); altrimenti, lo sparo avrebbe fatto cilecca e sarebbe continuato lo stato di vita.

L'indeterminatezza e' modellata asserendo il predicato Releases sul fluente Loaded,. E' da notare infatti che il predicato Releases non fa teminare il fluente, ma lo lascia completamente libero di oscillare tra i due stati.

The Ramification Problem

Si possono avere anche ***azioni con effetti indiretti*** che sono alla base del cosiddetto *ramification problem*. Per effetti indiretti si intendono quegli effetti non descritti esplicitamente dalle leggi causali, ma da questi derivanti attraverso altri vincoli, denominati *vincoli di stato*. I vincoli di stato esprimono relazioni logiche che possono instaurarsi tra fluenti in ogni momento.
Nel calcolo degli eventi i vincoli di stato sono espressi usando il predicato *HoldsAt* con un argomento di tempo universalmente quantificato.
Un esempio:
lo stato dell'essere felice, qualsiasi sia il tempo t considerato, si avvera se e solo se non e' attivo al tempo t lo stato di essere affamato o quello di aver freddo

HoldsAt(Happy(x), t) $\leftrightarrow$

$\neg$ HoldsAt(Hungry(x), t) $\neg \wedge$ HoldsAt(Cold(x), t)

questo significa, per esempio, che se al tempo t0 Fred e' affamato ma non ha freddo e sopravviene un'azione di nutrimento al tempo t1, allora, se al tempo t1 Fred continua a non avere freddo, possono dichiarasi soddisfatte le condizioni perche' Fred si trovi in uno stato di felicita' al tempo t2.

Dato che usiamo il formalismo completo (sono cioe' attivi i meccanismi che regolano la legge dell'inerzia), si deve fare attenzione, per non cadere in contraddizione, che il fluente iniziato o terminato da un assioma che riguarda gli effetti non sia a sua volta iniziato o terminato indirettamente da un vincolo di stato a meno che quel fluente non sia stato preventivamente rilasciato (Released).Ad esempio,

HoldsAt(Alive, t)$\leftarrow$ HoldsAt(Walking, t)

e' un vincolo di stato che rappresenta l'interdipendenza tra i due fluenti Alive e Walking, ma il suo uso potrebbe creare una indesiderata terminazione implicita di un fleunte. Infatti, una formula come Terminates(Shoot, Alive, t), combinata con il vincolo sopra, produrrebbe come conseguenza una terminazione impropria del fluente Walking, nel caso questo fluente fosse iniziato per effetto di una formula come InitiallyP. Si contravverebbe in tal modo ad una regola metodologica sull'uso dei vincoli di stato che impedisce che si possa far terminare indirettamente con una formula un fluente quando questo sia iniziato direttamente per effetto di un'altra formula.

L'uso dei vincoli degli effetti (*effect constraint*) serve proprio a rappresentare in modo adatto relazioni come quelle che intercorrono tra il fluente Alive ed il fluente Walking. I vincoli di effetto sono i simboli Initiates e Terminates con una variabile di azione universalmente quantificata. Il vincolo richiesto per il caso su esposto e':

Terminates(a, Walking, t) $\leftarrow$ Terminates(a, Alive, t)

Ci sono inoltre casi di propagazione istantanea di effetti indiretti interagenti che possono essere governati dai vincoli causali (*causal constraints*). In questi casi si introducono nuovi predicati definiti da assiomi e si denotano gli eventi necessari a descrivere l'interdipendenza tra i fluenti durante il processo.

Azioni con effetti non deterministici

Il calcolo degli eventi viene usato anche per rappresentare azioni che non hanno effetti deterministici. Tra i vari metodi Shanahan, si sofferma su quello dei *determining fluents*. Il determining fluent e' un fluente virtuale, che non ha inizio ne' fine e non e' soggetto alla legge di inerzia, ma serve a formalizzare l'azione. Nel giuoco 'testa o croce', ampiamente noto, il determining fluent (ItsHeads) serve a dirci se il gioco del lancio della moneta e' in atto oppure no, talche':

se col lancio (Toss) il fluente che si e' avviato ha lo stato 'testa'(Heads), questo implica che sia in atto il determining fluent (ItsHeads) e che il lancio effettuato abbia avuto l'esito 'testa'. Dato che il determining fluent e' attivo per definizione quando ci sono lanci, e questi ripetono sempre la stessa azione, se ipotizziamo che i lanci avvengano ai tempi 10, 20, 30, l'esito del lancio potra' avere valori diversi, ma sempre lo stesso valore tra un lancio e l'altro. Il fluente 'testa' (Heads), a sua volta, potra' assumere i seguenti stati HoldsAt(Heads, 15) $\wedge \neg$ HoldsAt(Heads, 25) $\wedge$ HoldsAt(Heads, 35) o altri ancora e terminera' quando il giuoco si interrompe Terminates(Toss, Heads, t) $\leftarrow \neg$ HoldsAt(ItsHeads, t)

Un altro caso di azioni che non hanno effetti deterministici e' il lancio della moneta su una scacchiera. Puo' accadere che cada sul quadrato bianco, solo su quello nero, oppure su ambedue. In questa situazione occorrono due determining fluents, (ItsWhite, ItsBlack), che danno origine ai rispettivi fluenti 'bianco' (OnWhite) e 'nero' (OnBlack) in modo da rappresentare i tre possibili risultati:

HoldsAt(ItsWhite, t) $\vee$ HoldAt(ItsBlack, t) $\vee$
(HoldsAt(ItsWhite, t) $\wedge$ HoldsAt(Its Black, t)

Dato che esiste una situazione di partenza (*InitiallyN*) per i fluenti OnBlack e OnWhite ed il lancio avviene al tempo 10 (*Throw*, 10), in tutti i modelli uno dei fluenti si avvera dopo il tempo 10 e in tutti i modelli questi fluenti ritengono i loro valori per sempre.

Azioni composte

Passiamo ora alla ***compound action***, gerarchicamente ordinata: per es., una azione (GoToWork) con due sottoazioni (WalkTo e TrainTo) che proviamo a descrivere cosi':

L'andare al lavoro (GoToWork) avviene nell'intervallo t1, t4 ed implica l'andare a piedi (WalkTo) alla stazione al tempo t1, prendere il treno (TrainTo) per Livorno al tempo t2, prendere il treno (TrainTo) per Cecina al tempo t3, ed andare a piedi (WalkTo) fino al posto di lavoro al tempo t4.
$t1<t2<t3<t4$

Questa semplice sequenza non tiene conto pero' degli effetti di qualche evento inatteso.
Se prima di prendere il treno per Cecina perdo la memoria non avverranno le seguenti azioni: prendere il treno per Cecina e andare a piedi al lavoro.
Quindi perche' la sequenza sia vera devo esplicitare che l'azione composta non termina negli intervalli delle varie soste, inserendo ¬ *clipped condition* ad ogni intervallo in modo da eliminare l'effetto di qualsiasi altro evento. Esempio:

$$\begin{aligned}
&\text{Happens(GoToWork,t1,t4)} \leftarrow \\
&\quad \text{Happens(WalkTo(Station), t1)} \wedge \text{Happens(TrainTo(Livorno), t2)} \wedge \\
&\qquad \text{Happens(TrainTo(Cecina), t3)} \wedge \text{Happens(WalkTo(Work), t4)} \wedge \\
&\qquad\quad t1<t2 \wedge t2< t3 \wedge t3< t4 \wedge \neg\, \text{Clipped(t1,At(Station),t2)} \wedge \neg\, \text{Clipped(t2 At(Livorno),t3)} \wedge \\
&\qquad\qquad \neg\, \text{Clipped(t3, At(Cecina), t4)}
\end{aligned}$$

Il calcolo degli eventi esteso

I simboli di predicato seguenti si riferiscono, in ordine, alle ***azioni concorrenti*** con effetti che interferiscono gli uni sugli altri ed al ***cambiamento continuo*** (come quando il fluire dell'acqua in un recipiente causa il crescere del suo livello nel recipiente):
.

Cancels(a1, a2, f) dice che l'occorrenza di *a1* cancella gli effetti di una simultanea occorrenza di *a2* sul fluente *f*
Cancelled(a, f, t1, t2) dice che l'effetto dell'azione *a* sul fluente *f* e' cancellata da un evento occorso tra il tempo *t1 e t2*
Trajectory(f_1, *t*, f_2, *d*) dice che il fluente f_1 e' iniziato al tempo *t* e che f_2 diviene vero al tempo *t+ d*

Dell'insieme di assiomi che denotano i simboli di predicato, chiamati XC, i primi sette corrispondono a quelli del formalismo completo EC con la differenza che il secondo, terzo, quinto e sesto devono comprendere la condizione di ¬ Cancelled che blocca l'applicabilita' dell'assioma nel caso di occorrenze simultanee di eventi che cancellano gli effetti reciproci. L'ottavo assioma XC8 definisce il predicato Cancelled

$$\text{Cancelled(a1, f, t1, t2)} \leftrightarrow \text{Happens(a2, t1, t2)} \wedge \text{Cancels(a2, a1, f)}.$$

Il nono assioma XC9 e' il rovescio di XC2 e governa il cambiamento continuo

$$\begin{aligned}
&\text{HoldAt(f2, t3)} \leftarrow \\
&\quad \text{Happens(a, t1, t2)} \wedge \text{Initiates(a, f1, t1)} \wedge \neg\, \text{Cancelled(a, f, t1, t2)} \wedge \\
&\qquad t2 < t3 \;\; t3 = t2 + d \wedge \text{Trajectory(f1,t1, t2, d)} \wedge \\
&\qquad\quad \neg\, \text{Clipped(t1, f1, t3)}
\end{aligned}$$

E' da aggiungere che il calcolo dei predicati esteso impiega il nuovo simbolo di funzione "**&**" che esprime gli effetti cumulativi delle azioni concorrenti. Il seguente assioma denotato CA definisce il nuovo simbolo:

Happens(a1&a2, t1, t2) $\leftarrow$ Happens(a1, t1, t2) $\wedge$ Happens(a2, t1, t2)

Il meccanismo della *circumscription* si estende pienamente al nuovo calcolo salvo l'inserimento nella congiunzione Σ dei predicati Trajectory e Cancels, l'inclusione nella congiunzione Δ dell'assioma CA e l'aggiunta della congiunzione Ψ per esprimere i vincoli di stato.

CIRC[Σ ; Initiates, Terminates, Releases, Cancels] $\wedge$
  CIRC[$\Delta \wedge$ (CA); Happens] $\wedge$ XC $\wedge\ \Psi \wedge \Omega$

Per spiegare il funzionamento del formalismo applicato alle azioni concorrenti Shanahan ricorre allo scenario della pentola contenente la zuppa [11].

La pentola colma di zuppa e'sul tavolo. Le azioni contemporanee di sollevamento della parte sinistra e di quella destra della pentola impedisce al contenuto di versarsi e permette di spostare la pentola dal tavolo. Sono richiesti la formula Cancels per annullare gli effetti delle due azioni (ognuna delle quali provocherebbe un versamento della zuppa, da destra o da sinistra) ed il simbolo "&" per cumulare gli effetti delle due azioni e determinare cosi' lo spostamento della pentola dal tavolo.

L'efficacia del calcolo degli eventi esteso, quando si tratti di ***eventi a cambiamento continuo***, viene illustrato in un esempio che usa il predicato Trajectory e riguarda i cosiddetti *triggered events* che si manifestano quando certi fluenti raggiungono determinati valori.

Il dominio comprende l'evento TapOn che da inizio ad un flusso di liquido (Filling) dentro un recipiente. Il fluente Filling e' presente mentre l'acqua scende nel recipiente, e il fluente Level(x), dove x e' un numero reale, segna il crescere del livello dell'acqua nel recipiente. Un'azione di OverFlow occorre quando l'acqua raggiunge il bordo del recipiente al livello 10 e il traboccamento da inizio al fluente Spilling.

Per garantire il calcolo di certi passaggi si utilizzano i seguenti simboli di predicato:

Il predicato Release applicato al fluente Level(x) lo libera dallo stato di inerzia dopo l'azione TapOn
Releases(TapOn, Level(x), t)

Il simbolo Trajectory descrive le variazioni continue dei valori del fluente Level(x) durante il flusso del fluente Filling, nell'assunzione che il livello cresca di una unita' per unita' di tempo
Trajectory(Filling, t, Level(x2), d) $\leftarrow$
  HoldsAt(Level(x1), t) $\wedge$ x2 = x1 + d

Un vincolo di stato garantisce che l'acqua nel recipiente abbia sempre un unico livello
HoldsAt(Level(x1), t) $\wedge$ HoldsAt(Level(x2), t) $\rightarrow$ x1 = x2

Una formula assicura che l'azione OverFlow si verifichi al livello fissato
Happens(OverFlow, t) $\leftarrow$
  HoldsAt(Level(10, t) $\wedge$ HoldsAt(Filling, t)

Davidson e Shanahan: i formalismi

I ragionamenti e gli esiti raggiunti dagli autori dei testi che abbiamo considerato non sono confrontabili. Le impostazioni si collocano in aree diverse e sono diversi l'approccio e le finalita': tuttavia, come ricordato nella nota precedente sia Davidson che Shanahan assegnano una centralita' al concetto di "evento" ed assumono che le azioni e gli eventi siano governati da relazioni causali.
Per D. Davidson l'argomentazione su azioni ed eventi e' rivolta ad individuare la forma logica degli enunciati di azione (parafrasi) su cui applicare la teoria della quantificazione nel contesto di una buona teoria della verita'. Il problema dell'ontologia e' solo prospettato come necessario; d'altra parte, per ammissione dello stesso autore, questo aspetto non rientrava nei suoi interessi immediati.

M. Shanahan, osserva i fenomeni (eventi) nel loro divenire e presenta una robusta teoria assiomatica basata su una ontologia minimale per calcolare i mutamenti che gli eventi nel tempo subiscono o provocano.
Nnondimeno tutti e due i testi consolidano le ragioni per basare lo sviluppo di sistemi informativi sul concetto di "evento".

Donald Davidson: la forma logica

Donald Davidson e' linguista e filosofo. Come filosofo considera le azioni e gli eventi concetti fondanti per esprimere cio' che accade nel mondo; come linguista ritiene che le azioni e gli eventi possano essere spiegati facendo attenzione a come se ne parla analizzando in particolare la forma logica e linguistica degli enunciati che usiamo per descriverli.
L'intuizione fondante e' che per esprimere il significato degli enunciati d'azione si debba ricorrere ad una ontologia di eventi individuali ed irripetibili.

Sono infatti gli eventi individuali ed irripetibili a stare nel rapporto causa effetto e sono le loro descrizioni (leggi per il mondo fisico-naturale; le condizioni di verita' per gli enunciati di azione), che forniscono la spiegazione dell'accaduto. La tesi centrale e' che la nozione di causalita' trova applicazione anche nell'ambito dell'azione umana, sebbene non esistano leggi in senso stretto che possano spiegare le ragioni di un agente.

Nel saggio in cui DD definisce "la forma logica dell'enunciato di azione"[2] e' evidente l'impostazione linguistica del suo tentativo. Lo scopo della ricerca e' evidenziare la forma logica di semplici enunciati riguardanti azioni, e spiegare la funzione logica e grammaticale delle parti o parole di tali enunciati in modo compatibile con le relazioni d'implicazione fra gli enunciati stessi, e la funzione che esse hanno in altri enunciati (non d'azione).

Jones imburro' il panino, deliberatamente, nel bagno, con un coltello a mezzanotte

Per mantenere le relazioni logiche tra le parti di un enunciato come questo DD propone di evidenziare l'elemento sintattico comune "imburro'" dal quale dipendono le relazioni semantiche tra gli enunciati di cui si compone

Jones imburro' il panino (dove, come, quando)

e crea un posto per gli eventi su cui quantificare.

In questa intuizione vi e' una coerenza logica tra concezione filosofica e traduzione linguistica: se il mondo e' popolato di eventi, come dimostra il nostro parlare, allora anche la forma logica degli enunciati su eventi deve ruotare intorno a questo concetto.

Per ottenere la forma logica di un enunciato e' necessario che siano stati analizzati alcuni aspetti: se si tratti di un evento singolo ed irripetibile oppure no, se si tratti di azione e, nel caso affermativo, a quale agente sia riferita l'azione e come venga espressa l'intenzionalita' dell'azione, inoltre, se tra le parti dell'enunciato ve ne siano alcune che implichino altre, se vi siano effetti, e come trattare i modificatori avverbiali ed altro ancora.

- Quando non sia evidente che si tratti di evento singolo e particolare, la forma logica del quantificatore esistenziale nell'enunciato di azione vincolera' la variabile d'azione. Se "Jones imburro' il panino" e' vero, allora c'e' un evento che lo rende vero e che puo' essere descritto.

- Se "Jones imburro'" rappresenta una azione, la soluzione e' quella di trovare la forma logica appropriata per gli enunciati di azione che la distingua da quella usata per gli eventi.

- Una condizione per parlare di azione , dice DD, e' che un posto argomentale venga riempito con un riferimento all'agente; inoltre il verbo deve implicare che la persona ha agito come agente, oppure deve essere usato un verbo non impegnativo aggiungendo che l'atto era intenzionale.

Di seguito la parafrasi dell'enunciato:

Esiste un evento che e' un evento di imburramento di panino (azione) da parte di Jones, Jones l'ha fatto deliberatamente, l'ha fatto col coltello, l'ha fatto a mezzanotte.

Murray Shanahan: la teoria assiomatica

Cinque sono gli elementi che servono a Shanahan per la sua ontologia: *individuals, instants, events-actions(types)*, and *fluents*, essenziali per descrivere un fenomeno: i predicati, e gli assiomi definiscono il loro uso a seconda delle funzioni da implementare. Non viene posto alcun vincolo alla loro estensione, l'autore stesso li indica come elementi di base. Conosciamo, nel campo della linguistica, adattamenti fecondi come l'introduzione dei concetti di *achievements, states, activities, accomplishments*, che specializzano i predicati del calcolo degli eventi [12].
Shanahan non parla di entita'. Nel suo mondo in movimento, dove i fatti sono sempre causati e causanti, importante e' la relazione causale, la dinamica e la tempistica degli effetti (il calcolo del quando, cosa e come), e la generalizzazione dei fenomeni (un certo formalismo si adatta ad un certo tipo di evento). Il largo numero di esempi denota tuttavia la convinzione che nel calcolo degli eventi possano essere rappresentati scenari molto diversi tra loro.
Gli eventi sono intesi da Shanahan come tipi (non si fa distinzione tra eventi ed azioni), possono essere concettualizzati (descritti) in modo diverso ed istanziati, cioe' ancorati ad un istante od ad un intervallo di tempo; essi marcano l'inizio e la fine di altri oggetti (i fluenti) che rappresentano l'altro aspetto degli eventi, quello della trasformazione che essi inducono: la trasformazione delle proprieta' dipendenti dal tempo, o il cambiamento parziale di oggetti.

Conclusioni

Queste ultime caratteristiche offrono spunti notevoli per immaginare svariate applicazioni della teoria di Shanahan in aree eterogenee: in particolare, per quello che ci riguarda, sembrano rispondere a certi requisiti che sono indicati nella premessa [1] come fondamentali per la nostra idea di modello di sviluppo per sistemi informativi.
Si tratta, quindi, di circostanziare esempi su un certo numero di fenomeni per dare conto della effettiva applicabilita' dei formalismi del calcolo degli eventi in campi eterogenei e nel contempo stabilire il grado di conciliabilita' tra i due modelli di rappresentazione degli accadimenti del mondo (l'ontologia di Shanahan e le definizioni di entita' che abbiamo acquisito per il modello). Questa indagine inevitabilmente coinvolge problemi concettuali ed aspetti formali: ed e' su tali questioni che ci vogliamo soffermare.

References

[1] Meghini, C., Venerosi, P., *Dai concetti alla definizione delle entita'*. ISTI-CNR, Memoria, Aprile 2003

[2] Davidson, D., *Azioni ed Eventi*, Bologna, Il Mulino, 1992

[3] Shanahan, M. *The Event Calculus Explained*, in Artificial Intelligence Today, ed. M.J. Wooldridge and M. Veloso, Lecture Notes in Artificial Intelligence, 1600, Springer, 409-430, 1990

[4] Kowalski, R.A., Sergot, M., *A logic-based calculus of events*, New Generation Computing, 4, 65-97, 1986

[5] McCarthy,J., Hayes, P., *Some Philosophical Problems from the Standpoint of Artificial Intelligence.* In D. Michie & B. Meltzer, eds., *Machine Intelligence*, Edinburg University Press, Edinburgh

[6] Hamm, F., Van Lambalgen, M., *Event Calculus, Nominalization, and the Progressive*, in Course 15: The Formal Foundation of Event Semantics, 1-65, University of Trento, 2002

[7] Shanahan, M., *Solving the Frame Problem – A Mathematical Investigation of the Common Sense Law of Inerytia.* MIT Press, Cambridge, MA. 1997

[8] Hanks, S., McDermott, D., *Nonmonotonic Logic and Temporal Projection,* Artificial Intelligence, vol. 33, 379-412, 1987

[9] Baker, A.B., *Nonmonotonic Reasoning in the Framework of the Situation Calculus*, Artificial Intelligence, Vol. 49, 5-23, 1991

[10] McCarthy, J., *Circumscriptium – A Form of Non-monotonic Reasoning*, Artificial Intelligence, Vol. 13, 27-30, 1980

[11] Gelfond, M., Lifschitz, V., Rabinov, A., *What are the Limitations of the Situation Calculus*? In Essays in Honor of Woody Bledsoe, Kluwer Academic, 167-179, 1991
[12] Van Lambalgen, M., Hamm, F. *Intensionality and Coercion*, in Course C15: The Formal Foundation of Event Semantics, 1-29, University of Trento, 2002.