

Università degli Studi di Padova
Dipartimento di Ingegneria dell'Informazione
Tesi di Laurea Magistrale

Ragionamento Temporale Fuzzy
per applicazioni *Elder Care*

Relatore: Prof.ssa Silvana Badaloni
Laureando: Giancarlo Filipetto

Corso di Laurea Magistrale in Ingegneria Informatica
Padova, 16 Luglio 2012

Indice

1	Introduzione	5
2	Panoramica sulle Tecnologie Assistenziali	7
2.1	Sistemi di fall detection	10
2.1.1	Introduzione	10
2.1.2	Soluzioni indossabili al fall detection	14
2.1.3	Soluzioni Ambientali o basate su Videocamera	20
2.2	Sistemi di rilevamento azioni	23
2.3	Sistemi per la gestione dei medicinali	26
2.4	Smart House	28
2.5	Altre proposte di tecnologie assistenziali	29
3	Sistema di Ragionamento Temporale Fuzzy	33
3.1	Relazioni Qualitative	35
3.1.1	Definizione di relazione qualitativa	35
3.1.2	Esempi di vincoli qualitativi	37
3.1.3	Algebra Qualitativa Fuzzy	38
3.2	Relazioni Quantitative	41
3.2.1	Introduzione alle relazioni quantitative	41
3.2.2	Definizione di relazione quantitativa	42
3.2.3	Esempio di relazione quantitativa	44
3.2.4	Operazioni fra relazioni quantitative	45
3.3	Integrazione di vincoli qualitativi e quantitativi	51
3.3.1	Conversione di vincoli quantitativi in qualitativi	52
3.3.2	Conversione di vincoli qualitativi in quantitativi	52
3.3.3	Operatori fra vincoli qualitativi e quantitativi	53
3.4	Algoritmo di Path Consistency	54
3.5	Problema della Frammentazione	56
3.6	Nozione di Evento	60

4	Implementazione	63
4.1	Sistema di ragionamento temporale fuzzy	63
4.1.1	Breve panoramica sul software prodotto	63
4.1.2	Standard per i file di input / output	66
4.2	Interazione con l'utente	67
4.2.1	Interpretazione del linguaggio utente	67
4.2.2	Interfaccia Grafica per reti temporali fuzzy	72
5	Proposta di Supervisore Fuzzy	81
5.1	Introduzione al problema	81
5.2	Ipotesi di sistema a tre livelli	84
5.3	Algoritmo per il controllo	85
5.4	Prototipo	90
5.4.1	Orologio di sistema	90
5.4.2	Sensori e stato degli eventi	91
5.4.3	Rappresentazione degli eventi	92
5.4.4	Interazione con l'utente	92
5.5	Osservazioni e proposte	94
6	Simulazione di scenario reale	97
6.1	Descrizione del Problema	97
6.2	Generazione della rete	98
6.2.1	Vincoli in linguaggio utente	98
6.2.2	Traduzione vincoli dal linguaggio utente	100
6.2.3	File dati	102
6.3	Simulazione del comportamento dell'utente	103
7	Conclusioni	107
A	Tabelle delle relazioni qualitative	109
B	Activity of Daily Living	113
	Bibliografia	115

Capitolo 1

Introduzione

Il concetto di popolazione anziana non può essere definito in modo univoco, perché cambia a seconda dello stato, della cultura, delle esperienze e delle opinioni personali. Risulta difficile stabilire una soglia d'età per delineare questa parte della popolazione, ma si possono ipotizzare i 65 anni (attuale limite minimo per l'età pensionabile in Italia) come il valore di riferimento per iniziare a caratterizzare quantitativamente il fenomeno.

Secondo le stime Istat del 2010 (riportate nel sito internet demo.istat.it) circa il 20% della popolazione residente sul suolo italiano risultava di età superiore ai 65 anni. Negli Stati Uniti d'America, secondo i dati dello United States Census Bureau (l'organo competente per la gestione dei censimenti negli Stati Uniti d'America, i cui dati sono accessibili tramite il sito internet census.gov), nello stesso anno tale valore era di circa il 10% ed è previsto un suo lieve aumento nell'immediato futuro.

Questi dati risultano particolarmente importanti se rapportati con il cambiamento della struttura familiare (che ha ristretto il numero dei suoi componenti) e delle sue dinamiche interne (con i crescenti impegni lavorativi che riducono la presenza in casa).

Il problema dell'assistenza agli anziani viene affrontato ricorrendo a soluzioni di assistenza domiciliare (vedi badanti) o di trasferimento in strutture di ricovero permanente. Sebbene queste scelte rappresentino spesso l'unica opzione possibile, non risultano soddisfacenti in quanto provocano disagi negli anziani ed alzano in modo cospicuo i costi del loro mantenimento. Emerge, quindi, il bisogno di soluzioni alternative che permettano di estendere la permanenza in casa di quei soggetti che presentano ancora un buon grado di indipendenza, fornendo la possibilità di controllo a quanti se ne prendono cura, anche se non fisicamente presenti.

L'interesse a queste soluzioni riguarda un ampio segmento di mercato, capace di attirare investimenti dalle aziende e proporre nuovi spunti di ricerca

nell'ambito accademico.

Con la fine del secolo scorso nuovi progetti sono stati presi in esame da alcuni gruppi di ricerca con lo scopo di esplorare il concetto di “assistenza anziani” (spesso etichettato con il termine “elder care”). Fra le molteplici sfumature che questo campo propone, gli sforzi della ricerca universitaria si sono principalmente concentrati nello studio e nella costruzione di sistemi automatici che sfruttino i più recenti sviluppi tecnologici al fine di monitorare il benessere degli anziani e di rispondere ai problemi in cui un utente può incorrere, come cadere o dimenticarsi di assumere i medicinali che gli sono stati prescritti.

È in quest'ambito che si sviluppa la presente tesi; gli obiettivi sono due: da un lato ottenere una panoramica delle soluzioni presenti in letteratura e dall'altro proporre una possibile applicazione del sistema di ragionamento temporale fuzzy, sviluppato presso il Dipartimento di Ingegneria dell'Informazione di Padova, allo scopo di monitorare la giornata di un soggetto anziano e suggerire eventuali modifiche per renderla il più possibile conforme al comportamento indicato dal medico curante.

Capitolo 2

Panoramica sulle Tecnologie Assistenziali

In letteratura sotto il termine tecnologie assistenziali (in inglese “Assistive Technology”, AT) vengono raggruppati studi che si differenziano sia per obiettivi che per mezzi impiegati. Tale varietà si spiega facilmente se si considerano gli elementi che solitamente sono centrali per questa tipologia di progetti: l’anziano, la sua salute, i problemi che deve affrontare quotidianamente.

Analizzando lo studio [Hirsch et al, 2000] si può notare come (per gli anziani e quanti se ne prendono cura) lo scopo delle tecnologie assistenziali sia quello di mantenere, o se possibile migliorare, la “Qualità della Vita” (in inglese “Quality of Life”, QoL) dei potenziali utenti. Uno studio più accurato non è riuscito a definire in modo formale quali parametri caratterizzino la QoL ma è stato possibile descriverla come un equilibrio fra gli aspetti di indipendenza e di impegno: con il primo ci si riferisce solitamente a tutto ciò che gli anziani possono fare in modo autonomo e senza il controllo esterno, con il secondo alla loro capacità di relazionarsi, di condividere risorse e di aiutarsi reciprocamente.

Questo equilibrio deve essere ricercato alla luce del vissuto e dello stato attuale del potenziale utente. Possiamo notare come, ad esempio, incidenti e malattie non siano dei “momenti passeggeri” nella vita di un anziano ma rischiano, invece, di modificare in modo permanente la vita del soggetto stesso. Allo stesso modo il livello attuale di abilità del soggetto costituisce un punto focale per la vita dello stesso. Com’è facile intuire, il livello delle abilità personali tende a decadere con il passare degli anni, il problema è riconoscerne i segni ed adeguarsi. In [Hirsch et al, 2000] si evidenzia come nei soggetti anziani si possa riscontrare una forte differenza fra il livello percepito e quello reale, che viene prima sovrastimato e poi sottostimato durante la vita dello stesso. Tale cambio è spesso dovuto ad un evento traumatico, come una

caduta o una diagnosi poco favorevole, e questo ha delle forti ripercussioni sul soggetto: finché le abilità sono sovrastimate la possibilità di incidenti aumenta, quando sono sottostimate diminuiscono anche l'indipendenza e la capacità di relazione dell'anziano.

Questi sono solo alcuni degli aspetti su cui focalizzarsi quando si progettano tecnologie assistenziali, altri sono legati allo stato mentale dell'anziano (le proposte variano molto se soffre di un normale decadimento della memoria rispetto a malattie più problematiche come l'Alzheimer) nonché la sua residenza (se continua a vivere nella propria abitazione, piuttosto che con i familiari o in un istituto specializzato) e le distanze che lo separano da conoscenti, familiari, o personale medico.

Come indicato in [Mann et al, 1999] ed in [Giuliani et al, 2005] gli anziani, soprattutto quelli in età non troppo avanzata, sono disposti a modificare il proprio ambiente con tecnologie assistenziali se queste hanno lo scopo esplicito di migliorare la loro QoL perché vengono percepite come un modo per ridurre i tempi persi, incrementare il numero di attività senza stancarsi troppo ed aumentare la propria indipendenza.

Inoltre, sempre lo studio proposto da [Mann et al, 1999] ha evidenziato come gli anziani che arricchiscono il loro ambiente domestico con tecnologie assistenziali presentino un declino delle capacità inferiore rispetto ai coetanei che non ne fanno uso, soprattutto per le funzionalità motorie, con una conseguente riduzione dei costi per la loro cura.

Di contro in [Giuliani et al, 2005] vengono esplicitati alcuni motivi di diffidenza verso le AT, viste spesso come fragili, difficili da utilizzare e dotate di istruzioni poco chiare. Non tutte le proposte vengono accettate nello stesso modo: implicitamente l'anziano organizza le proprie attività in una gerarchia e non ritiene necessario mantenere tutte le abitudini, ma solo quelle a priorità maggiore. In quest'ottica non è utile sostituire una partita di carte con un gioco sul computer (anche perché il bisogno è, solitamente, quello di socializzare e non quello di giocare), mentre sono accettate eventuali modifiche per gli aspetti di sicurezza personale e di gestione dei medicinali.

Per queste ragioni un fattore importante è il livello di confidenza degli utenti con le tecnologie a loro dedicate: rischia di essere inutile sviluppare un sistema di AT quando i potenziali utenti preferiscono soluzioni meno performanti ma più intuitive o vicine alle proprie necessità.

I lavori di [Miskelly, 2001] e [O'Brien, Mac Ruairi, 2009] permettono di individuare alcune linee guida sugli strumenti presenti sul mercato per supportare gli anziani in casa o in istituti specializzati. Fra le componenti facilmente reperibili già da diversi anni vi sono, per esempio:

- l'allarme comunitario: il dispositivo è collegato alla linea telefonica e permette, una volta attivato, di inviare una richiesta pre-registrata di soccorso ad un proprio custode o, se non rintracciato, ad una rete di conoscenti. Oltre al supporto fisso viene fornita anche un'unità mobile, spesso portata al collo (come un ciondolo) o al polso;
- monitor della salute: ispirati alla forma ed al posizionamento agli orologi da polso, questi dispositivi hanno lo scopo di monitorare le pulsazioni cardiache, la temperatura della pelle ed il grado di movimento. Nei primi giorni d'utilizzo il dispositivo registra l'andamento dei segni vitali e, una volta generato un pattern valido, lo utilizza per verificare lo stato dell'utente ed in caso di necessità invia una segnalazione d'allarme;
- sensori di movimento: implementati tramite pannelli sensibili alla pressione, sensori infrarossi, sensori di contatto applicati alle porte, ecc... tutti questi strumenti permettono di monitorare la posizione dell'utente e di inviare delle segnalazioni in caso di spostamenti percepiti (si pensi ad esempio ai movimenti notturni degli anziani per andare in bagno);
- sensori di luminosità per mantenere sempre un livello minimo di luce negli ambienti e sensori per la regolazione della temperatura;
- rilevatori di fumi, di fuoco e rilevatori applicati alla fornitura di gas della cucina sono alcuni esempi di allarmi ad hoc progettati per rilevare o evitare l'insorgere di incendi in casa.
- strumenti per l'assistenza al movimento, come per esempio i bastoni da passeggio o i deambulatori;
- localizzatori del soggetto o degli oggetti da lui utilizzati, che si appoggiano al sistema GPS, alle etichette RFID, ai microfoni ed alle videocamere.

Questi ed altri strumenti hanno lo scopo di favorire il prolungamento della permanenza dei soggetti anziani nelle loro abitazioni, mantenendo comunque un controllo costante in caso di incidenti domestici e migliorando la loro esperienza abitativa, o di agevolare il lavoro di controllo da parte del personale degli istituti di ricovero.

Di seguito sono riportate le descrizioni di alcuni sistemi sperimentali, suddivisi per obiettivi, con lo scopo di analizzare le tecnologie utilizzate e gli algoritmi che sono stati applicati. Vengono poste, di regola, le ipotesi semplificative per cui gli anziani sono sostanzialmente indipendenti (e quindi le AT hanno lo scopo di fornire controllo per le emergenze o una correzione

del comportamento) e in ambiente domestico indipendenti (salvo alcuni casi in cui lo scenario è quello dell'istituti di ricovero).

2.1 Sistemi di fall detection

2.1.1 Introduzione

In questa sezione vengono presentati alcuni approcci al problema della rilevazione delle cadute (in letteratura noto come “Fall Detection”) unitamente ad altre due tematiche: la rilevazione della postura del soggetto (intesa come eretta, seduta, sdraiata) e la posizione che il soggetto occupa nello spazio (ossia la stanza dell'ambiente domestico o la posizione geografica all'esterno). Tali binomi sono, spesso, proposti in letteratura per distinguere una caduta da un movimento volontario da una postura all'altra, o per individuare la posizione del corpo ed indirizzare i soccorsi. Questi studi si rivelano particolarmente interessanti perchè possono ridurre il tempo che intercorre tra un'incidente e la scoperta dello stesso, rendendo i soccorsi più tempestivi e riducendo i rischi di danni permanenti. I primi studi in questo settore sono stati proposti da [Lord, Colvin, 1991], che presentarono dei prototipi sia basati su sensori che su videocamera; è, però, dalla fine degli anni '90 che l'interesse per il problema e le soluzioni proposte sono incrementate.

Un primo grande ostacolo, che si riscontra quando ci si approccia il problema, è individuare una definizione della nozione di caduta. Sebbene il “cadere” sia un concetto noto a tutti, risulta, invece, molto più difficile andare a specificare dei parametri che permettano di identificarlo in una sequenza di posizioni assunte da un soggetto in un breve intervallo di tempo (nell'ordine di pochi secondi). L'incertezza è dovuta, fra gli altri, ad aspetti quali l'interlocutore coinvolto (sia esso un anziano o il personale che si occupa di lui), gli effetti che l'atto provoca sul soggetto interessato (presenza o meno di dolore, tempi necessari all'individuo per riprendere una posizione eretta) e la posizione relativa (in alcuni lavori l'essere sdraiati sul pavimento è condizione necessaria per definire una caduta).

Una generica definizione è stata formulata nel 1987 dal “Kellogg international working group on the prevention of fall in the elderly” che definisce una caduta come una “unintentionally coming to the ground or some lower level and other than as a consequence of sustaining a violent blow, loss of consciousness, sudden onset of paralysis as in stroke or an epileptic seizure” (ossia un non intenzionale andare a terra o ad altro livello inferiore non derivante da un colpo violento, dalla perdita di conoscenza, dall'improvviso insorgere di

paralisi come per un infarto o un episodio epilettico) . Tale definizione, come viene fatto notare nel lavoro di Noury, risulta adeguatamente generale e può essere estesa anche a fenomeni quali vertigini ed altri temporanei problemi medici. Una caduta può essere vista, quindi, come conseguenza di un movimento non controllato da parte del soggetto che inizialmente si presenta in posizione eretta, seduta o sdraiata e che termina, in prima approssimazione, con una posizione distesa a terra o su altro livello inferiore rispetto a quello originario.

Deve essere, infine, considerato l'istinto del soggetto, che lo porta ad evitare la caduta attraverso la generazione di movimenti compensatori, aspetto che spesso comporta la presenza di numerosi stati finali esterni alle definizioni appena descritte. Se si prende, ad esempio, il caso di un individuo che cammina, allora un'eventuale caduta frontale può essere bloccata in principio portando una gamba in avanti, per compensare il movimento e recuperare la posizione eretta, oppure può essere attutita portando le mani in avanti. In modo analogo se il soggetto cade all'indietro è probabile che la sua posizione finale sia quella distesa, ma può tentare di ridurre i danni cercando di sedersi sul pavimento.

I lavori [Noury et al, 2007], [Yu, 2008] riportano alcuni parametri presenti in letteratura che permettono di identificare in modo operativo una caduta. Da un punto di vista macroscopico una caduta si sviluppa nell'arco di pochi secondi (al massimo 3, come indicato nel lavoro di Yu) e porta a modificare in modo sostanziale la sagoma del soggetto passando, per esempio, da una forma principalmente verticale (eretta) ad una orizzontale (distesa). Analizzando nel dettaglio le singole fasi della caduta è possibile notare come, durante l'atto stesso, il baricentro del corpo e la sua testa presentino la componente verticale della velocità che varia secondo un moto uniformemente accelerato. D'altra parte l'impatto dell'individuo, con un qualsiasi altro oggetto o superficie, provoca una brusca variazione del vettore accelerazione. In prima approssimazione, infine, il corpo giace disteso sul pavimento e nel caso in cui la caduta presenti gravi conseguenze sul soggetto stesso, tale posizione viene mantenuta per un significativo lasso di tempo.

In quest'ambito è necessario precisare la fondamentale importanza dei parametri utilizzati e dei relativi valori di soglia applicati. Per esempio la posizione orizzontale del corpo è sicuramente un parametro facile da verificare ma in ambiente domestico, se non utilizzato in combinazione con altri, può generare numerosi falsi allarmi; infatti la posizione distesa può essere adottata anche durante i periodi di sonno, siano o meno quelli abituali e/o programmati. D'altra parte la conferma che una caduta è avvenuta tramite la verifica che il soggetto mantenga una posizione distesa per un certo intervallo

temporale prevede un problema di determinazione dei tempi di osservazione: se troppo stretti allora il rischio di falsi allarmi è alto, d'altra parte se troppo lunghi si può impedire un tempestivo intervento.

Per quanto concerne le tecnologie hardware utilizzate per rilevare alcune delle condizioni sopra espresse, esse si possono dividere in tre grandi categorie, ossia:

- **indossabili:** piccoli dispositivi che possono essere posizionati su indumenti o comunque a stretto contatto con il soggetto come accelerometri e giroscopi. Vantaggiosa è la possibilità di controllare la persona indipendentemente dal luogo fisico in cui si trova. Sono da considerare, però, anche la possibile percezione di invadenza che il sistema genera sul soggetto, i pesanti vincoli sulle posizioni relative del soggetto con lo strumento stesso e la mancanza di una sicura verifica da remoto sullo stato dell'individuo;
- **camera:** permettono di produrre immagini bi e tridimensionali delle zone di interesse. Da un lato risolvono sicuramente il problema della verifica da remoto e non impongono vincoli al soggetto una volta installate; di contro possono essere viste come estremamente invasive della privacy, soprattutto in ambienti quali il bagno e la camera da letto, ed il loro funzionamento è utile fintanto che il soggetto si trova all'interno dell'area di copertura del sistema; devono essere inoltre coordinate secondo la posizione dell'individuo nell'ambiente e le loro rilevazioni sono influenzate dall'illuminazione ambientale e dal materiale delle superfici che possono generare riflessi. Limitatamente al problema della privacy e della necessità di illuminazione per la ripresa video, alcune soluzioni in letteratura prevedono l'utilizzo di videocamere ad infrarossi, che sono meno influenzate dall'illuminazione ambientale e non riproducono con dettaglio le fattezze dell'individuo;
- **ambientale:** ossia la modifica di superfici, come i pavimenti, per poter rilevare pressione ed impatti. Tipiche, di questo approccio, sono le piastrelle dotate di rilevatori di pressione. Presentano il vantaggio dell'assoluta non invasività, ma generano i problemi tipici degli altri sistemi ossia la mancanza di controllo da remoto e la sua limitazione alla sola area di installazione.

Combinazioni di una o più di queste tecnologie sono presenti in letteratura e si differenziano a seconda del soggetto analizzato e della struttura che lo ospita. Soluzioni basate su sensori indossabili sono proposte per soggetti

che sono ospitati in strutture di ricovero. In modo opposto, se l'individuo presenta un buono stato psico-fisico e si vuole massimizzare la sua indipendenza e la sua permanenza in casa, allora è consigliabile una combinazione di tecniche con una parte indossabile, per controllare i movimenti fuori dall'ambiente domestico (per esempio integrando il tutto in indumenti quali la giacca), e da un sistema ambientale/visivo in casa, per svincolare il soggetto dai sensori indossabili.

Prima di concludere la trattazione generica del problema di fall detection e passare ad una panoramica sugli studi pubblicati, è doveroso citare l'articolo [Klenk et al, 2011], che si prefigge di analizzare le variazioni di prestazione dei prodotti fra la fase di collaudo in laboratorio e le performance reali.

Lo studio ha potuto confrontare le misurazioni di cadute di spalle, ottenute tramite accelerometri triassiali, di alcuni pazienti affetti da una particolare forma di Parkinson (per i quali la probabilità di caduta è maggiore rispetto alla media) con i risultati di due simulazioni compiute su soggetti giovani (fra i 20 ed i 30 anni, fascia d'età in cui vengono solitamente selezionati i soggetti per i test dei sistemi di fall detection). Le simulazioni si prefiggevano lo scopo di valutare la capacità dei volontari di "cadere come un anziano", richiedendo una successione di cadute di spalle, e le reazioni degli stessi nel tentativo di prevenire una caduta certa. Le analisi sui dati raccolti hanno evidenziato che i due esperimenti hanno prodotto misurazioni statisticamente diverse rispetto a quelle reali.

Fra le cause di tale divario vi sono, innanzitutto, le diverse condizioni ambientali ed iniziali applicate per le misurazioni; se nei casi reali le cadute sono imprevedibili per il soggetto e terminano solitamente su superfici rigide, nelle simulazioni i soggetti hanno coscienza dei tempi e, per questioni di sicurezza, le cadute terminano su superfici morbide per annullare i danni ai volontari. Altro aspetto emerso dalle simulazioni è che i soggetti non hanno una chiara idea di come poter imitare il comportamento di un soggetto anziano (come richiesto nel primo esperimento). D'altra parte se viene loro richiesto di prevenire una caduta (come nel secondo esperimento), i soggetti giovani presentano movimenti compensatori più accentuati rispetto a soggetti di età più avanzata.

Come affermano gli stessi autori, sebbene non sia certa la possibile estensione delle misurazioni reali raccolte a tutta la fascia di soggetti anziani, è evidente che esiste un naturale divario fra le statistiche reali e quelle simulate. Per poter ridurre tale differenza è necessario compiere una fase di training sui volontari che preveda, per esempio, l'utilizzo di filmati per fornire un'idea verosimile del comportamento atteso.

2.1.2 Soluzioni indossabili al fall detection

Analisi preliminare

In [Noury et al, 2008] viene riportato uno studio preliminare per individuare le richieste di potenziali utenti di un sistema indossabile per la rilevazione di cadute. Lo studio è stato compiuto tramite questionario (470 inviati, 149 ricevuti compilati) su persone che avevano già familiarità con dispositivi di “allarme comunitario”, che vivono sole e/o che sono identificate come “chronic fallers”, soggetti con cronica tendenza alle cadute.

Dall’analisi sui dati si ricava che 2 individui su 3 sono interessati al dispositivo e che andrebbe, in buona parte, ad integrare le funzioni dei medaglioni per la richiesta automatica di soccorso. Per quanto riguarda le caratteristiche del prodotto, esso dovrebbe essere di piccole dimensioni, di peso limitato e possibilmente non visibile ad estranei. Essendo il dispositivo di tipo indossabile, fra le posizioni preferite vi sono il polso, l’anca ed il tronco; soluzioni integrate con braccialetti, vestiti e cinture sono apprezzate ma versioni sottocutanee o sotto l’ascella (stile fondina) non sono molto gradite.

Ne viene rilevata l’utilità in ambienti quali il bagno (ed in particolare sotto la doccia) e, seppure in modo minore, anche durante i periodi notturni e per gli spostamenti fuori casa. In merito ai feedback prodotti dal dispositivo si preferisce la forma visiva per le indicazioni di corretto funzionamento dello stesso ma nelle situazioni di allarme è ritenuto più opportuno il feedback audio. Infine il sistema dovrebbe essere alimentato a batteria, dovrebbe essere garantito l’intervento di tecnici specializzati per le operazioni di manutenzione e, se possibile, sarebbe apprezzata l’integrazione delle funzioni con sensori di altra natura, primi fra tutti quelli per la rilevazione della pressione sanguigna.

Soluzioni basate su acceleratori e giroscopi

In [Bourke et al, 2007] vengono utilizzati due acceleratori tri-assiali posti rispettivamente sul torace e sulla coscia per ricavare il vettore di accelerazione totale nei due punti al variare del tempo. Le misurazioni sono state eseguite su due gruppi di soggetti diversi su attività diverse. Ad un gruppo di giovani (tra i 20 ed i 30 anni) è stato chiesto di simulare, in ambiente sicuro e controllato, una serie di cadute, mentre ad un insieme di anziani (di età compresa fra i 70 e gli 80 anni circa) è stato chiesto di svolgere nel loro ambiente quotidiano, una serie di attività giornaliere (nell’articolo indicate come ADL, activity of daily living) come il sedersi ed alzarsi da sedie, sdraiarsi e rialzarsi, camminare, ecc.

Per ogni accelerometro e per ogni attività i dati sono stati elaborati nel seguente modo:

1. i dati sono stati filtrati tramite filtro di Butterworth passa basso con frequenza di taglio a 250Hz;
2. i dati sono stati aggregati tramite il calcolo del modulo dell'accelerazione complessiva del soggetto (tramite radice della somma dei quadrati);
3. è stato calcolato il valore di picco massimo (indicato come upper peak value UPV) ed il valore di picco minimo (lower peak value LPV).

Analizzando separatamente i dati provenienti da tronco e coscia, e limitatamente ai valori delle simulazioni di cadute, sono stati definiti i valori di upper fall threshold $UFT = \min\{UPV_i\}$ ed in modo analogo il valore di lower fall threshold $LFT = \max\{LPV_i\}$. Fintanto che la misurazione di una attività compiuta dall'utente rimane all'interno dell'intervallo $[LFT, UFT]$ allora l'attività viene classificata come "ordinaria", altrimenti se viola anche uno solo dei due valori di soglia, la caduta viene ritenuta certa. I quattro valori di soglia LFT_{TRUNK} , UFT_{TRUNK} , LFT_{THIGH} , UFT_{THIGH} sono stati testati separatamente portando valori di specificity (rapporto fra le attività correttamente classificate ed il totale) pari rispettivamente a 91.25%, 100.00%, 67.08%, 83.33%. I risultati prodotti, nonostante non contengano misurazioni su reali cadute di soggetti anziani, dimostrano come il tronco sia una delle zone più valide per installare sensori per la fall detection.

In un successivo lavoro [Bourke, Lyons, 2008] lo stesso protocollo per la raccolta dei dati è stato utilizzato sulle misurazioni di un giroscopio bi-assiale che ha lo scopo di misurare le velocità angolari ω_r attorno all'asse frontale ed ω_p sull'asse laterale. Tali rilevazioni sono state, quindi, elaborate tramite filtro di Butterworth passa basso con frequenza di taglio a 100Hz. I valori ottenuti sono stati, quindi, aggregati ottenendo $\omega_{res} = \sqrt{\omega_r^2 + \omega_p^2}$. Come nello studio precedente sono stati valutati i valori di picco di ogni movimento (cadute + ADL) ma in questo caso si è riscontrato una sovrapposizione fra le due categorie di azioni. Per questa ragione, in cascata al controllo $\omega_{res}(t) > FT1$ con $FT1$ minimo valore di picco delle cadute, sono stati posti controlli anche su altri due parametri θ_{res} angolo di rotazione totale e α_{res} accelerazione angolare. Utilizzando il solo test su ω_{res} gli esperimenti hanno prodotto valore di specificity pari al 97.5%; valore che sale al 100% se si aggiungono anche i controlli su θ_{res} e α_{res} .

Approccio basato su due sensori composti da accelerometri e giroscopi tri-assiali è quello proposto in [Li et al, 2009]. In questo caso le informazioni

congiunte dei due sensori, posti rispettivamente su torace e coscia, permettono di accertare lo stato dell'individuo, sia esso una postura statica o una fase di transizione.

Da ogni sensore vengono ricavate l'accelerazione totale e l'angolo totale di rotazione e questi valori sono elaborati in finestre di 1 secondo. Se le variazioni all'interno di una finestra sono contenute allora si considera il soggetto in posizione statica, altrimenti è in atto una transizione fra uno stato e l'altro (per esempio da eretto a seduto). Andando a differenziare le posizioni statiche grazie alle misurazione, l'attenzione viene concentrata solo su quella distesa perché possibile sintomo di una caduta. In questi casi vengono analizzati i 5 secondi precedenti all'istante attuale e se si è verificata una transizione non controllata (differenziata da un movimento volontario secondo alcuni parametri di soglia) allora viene inviato un allarme.

I test sono stati compiuti su un gruppo di giovani ai quali è stato chiesto di cimentarsi sia in attività quotidiane (dalle principali posizioni e le rispettive transizioni ad attività quali camminare, correre, salire e scendere scale) e su loro varianti con caduta. I risultati hanno dimostrato una buona capacità del sistema di classificare le varie azioni, identificando, però l'incapacità di riconoscere cadute verticali, come quelle in cui il soggetto tenta di rallentare la caduta appoggiandosi a pareti, e ad azioni come il "jumping into bad".

Soluzioni che integrano la comunicazione

Nei lavori che seguono l'attenzione è posta sia sulla generazione di un sistema che rilevi le cadute, sia sulla necessità di generare in automatico segnali per la richiesta di soccorso. Risultano interessanti gli scenari rispetto ai quali vengono sviluppati i progetti. Mentre il primo gruppo vuole operare all'interno di strutture ospedaliere e di degenza, facilitando così il compito del personale della struttura e permettendo di focalizzare l'attenzione solo sulle situazioni di reale necessità, il secondo punta sull'aumentare l'indipendenza dei soggetti anziani all'interno delle loro routine quotidiane integrando il prototipo con dispositivi già presenti o facilmente inseribili nella vita quotidiana.

Negli studi sviluppati in [Wang et al, 2008] e [Huang et al, 2009] vengono trattati rispettivamente gli aspetti della rilevazione delle cadute e della localizzazione dei pazienti in difficoltà all'interno di strutture di ricovero.

Il sistema di monitoraggio si basa su accelerometro tri-assiale posizionato vicino all'orecchio. Questa scelta permette di ottenere misurazioni da una parte del corpo che è solitamente più curata, meno soggetta a bruschi movimenti e permette una facile integrazione con dispositivi per migliorare

l'udito, già adottati da parte della popolazione anziana. La segnalazione di allarme viene inoltrata se si verificano alcune delle seguenti condizioni:

1. il modulo dell'accelerazione totale supera il valore di soglia di 6g;
2. il modulo dell'accelerazione sul piano orizzontale è superiore a 2g, segue quindi un periodo in cui l'accelerazione totale è stazionaria attorno ad 1g (coincide con la presenza di una persona ferma distesa per terra) e, la velocità stimata d'impatto sulla superficie è superiore ai 2 m/s (dato ricavato considerando il corpo in caduta libera).

Il sistema di localizzazione e di invio di una richiesta di soccorso si basa su una rete ad hoc, dove è possibile individuare: un nodo "centrale" che ha il compito di interfacciarsi con il personale della struttura (per esempio gli infermieri); nodi fissi sparsi in tutta la struttura di cui è nota e costante la posizione; nodi mobili collegati al dispositivo di rilevazione cadute sui pazienti. Quando un nodo mobile deve inviare una richiesta di soccorso compie i seguenti passi:

1. invia una richiesta broadcast single hop ai nodi fissi per conoscerne l'identità;
2. invia una richiesta broadcast single hop ai nodi noti per la stima del parametro RSSI (misura della potenza ricevuta dal segnale radio), congiuntamente con la posizione fisica degli stessi;
3. stima in locale la propria posizione ed invia la richiesta al nodo centrale tramite collegamento multi-hop sui nodi fissi.

La fase sperimentale ha dimostrato che il sistema di rilevazione riesce a distinguere le situazioni di caduta dalle normali attività, compreso il distendersi velocemente nel letto, anche se i valori di soglia possono essere migliorati. Infine il sistema di richiesta e localizzazione può fornire, se utilizzato, una maggior libertà al personale delle strutture ospedaliere liberandoli dal semplice controllo dei pazienti e permettendo loro di concentrarsi sulle situazioni più urgenti.

Differenti sono le motivazioni e gli scenari analizzati in [Dai et al, 2010], che osserva come i sistemi che si appoggiano su "allarmi comunitari" funzionano soltanto se il soggetto si trova all'interno dell'area di copertura del dispositivo. La soluzione proposta vuole ovviare al problema installando un'applicazione all'interno di uno smartphone già dotato di accelerometri e giroscopi tri-assiali.

Dopo una fase iniziale, nella quale vengono caricate le impostazioni di base dell'utente, il software opera in background fintanto che i parametri calcolati rimangono entro valori di soglia accettabili. Quando una caduta è rilevata viene subito attivato un allarme sul dispositivo e, solo se tale segnale non viene gestito per un prefissato periodo di tempo, allora l'applicazione inizierà ad inoltrare le richieste di assistenza ai numeri prescelti. In questo modo si vogliono diminuire i componenti hardware indossabili ed eventuali falsi positivi possono essere gestiti localmente senza l'intervento del personale di fiducia.

Per quanto riguarda i controlli sui valori degli accelerometri, vengono valutati due parametri: il modulo dell'accelerazione totale

$$|A_T| = \sqrt{|A_x|^2 + |A_y|^2 + |A_z|^2}$$

e il modulo dell'accelerazione verticale

$$|A_v| = |A_x \sin\theta_z + A_y \sin\theta_y - A_z \cos\theta_y \cos\theta_z|$$

. I due parametri sono, quindi elaborati nel medesimo modo:

1. si valuta se la differenza di $|A_i|$ all'interno della finestra win_{ti} eccede la soglia Th_{ti} ;
2. se 1 è verificato si controlla che nella finestra win_{ci} , che segue win_{ti} , la differenza di $|A_i|$ è inferiore alla soglia Th_{ci} .

Se le due condizioni sono verificate per entrambi i parametri allora viene lanciato un segnale di allarme perché si rileva una caduta seguita da una fase di inerzia. Il sistema è stato testato su giovani dai 20 ai 30 anni ai quali è stato chiesto di eseguire cadute ed attività quotidiane, posizionando il cellulare sulla vita (cintura), sul petto (tasca della maglietta) o sulla coscia (tasca dei pantaloni). Le sue prestazioni sono state comparate con altre tecniche presenti in letteratura ed in commercio evidenziando risultati simili. E' stata, inoltre, valutata l'efficienza energetica della soluzione proposta rispetto al funzionamento del dispositivo stesso in assenza dell'applicazione in background: i dati hanno evidenziato che il consumo della batteria non viene significativamente influenzato.

Approccio basato su machine learning

I prototipi finora presentati possono essere classificati come sistemi esperti ma in letteratura il problema del fall detection è stato approcciato anche con tecniche di machine learning.

Un esempio a tal proposito è il lavoro [Zhang et al, 2006]. Da un punto di vista hardware il sistema costruito può essere visto come un predecessore di [Dai et al, 2010] avendo previsto di collegare ad un cellulare un apparato basato su accelerometro tri-assiale, per garantire gli aspetti di rilevazione caduta e trasferimento allarme verso un apposito server. Il criterio centrale è basato sul fatto che una caduta prevede una fase iniziale di normale attività giornaliera, seguita dalla caduta vera e propria e da una fase finale di mantenimento della posizione raggiunta. Seguendo a ritroso tale sequenza di stati il flusso di operazioni utilizzate è il seguente:

1. viene analizzato se nell'ultimo secondo vi è un periodo di stasi, verificando che il modulo dell'accelerazione totale sia intorno a $|g|$;
2. se il primo punto è verificato il soggetto potrebbe essere caduto e vengono, quindi, analizzati i dati prodotti degli ultimi 1,5 secondi. Tali misurazioni vengono classificate tramite Support Vector Machine (SVM) per distinguere i casi di potenziale caduta dalle normali attività giornaliere;
3. se il controllo tramite SVM ha successo i dati vengono mappati su un features space secondo il Kernel Fisher Discriminant e classificato come caduta o meno secondo l'algoritmo k-Nearest Neighbors.

Gli autori evidenziano come, secondo la teoria, un SVM possa distinguere le cadute dagli altri eventi ma, in tal caso, la dimensione del feature space sarebbe risultata troppo elevata. Per tale ragione si è preferito distinguere fra attività giornaliera e potenziali situazioni di pericolo ed inserire in cascata un ulteriore controllo. Per quanto riguarda i dati rilevati in fase sperimentale, misurati su soggetti giovani, anziani ed un manichino, sono stati ripartiti in training e test set rilevando, infine, un rapporto di correttezza (elementi classificati correttamente sul totale dei test) intorno al 93%.

L'approccio proposto in [Dinh, Struck, 2009] prevede di combinare logica fuzzy e reti neurali per il fall detection. Nel dettaglio si ha che:

1. un accelerometro tri-assiale viene utilizzato per seguire gli spostamenti del soggetto in esame;
2. i dati prodotti (in coordinate cartesiane) sono convertiti in coordinate polari;
3. i dati sono filtrati per ottenere i valori medi di accelerazione e variazioni angolari;

4. le informazioni così prodotte sono convertite secondo logica fuzzy per ottenere indicazione del tipo di movimento prodotto, classificato secondo le tipologie “rest” ed “heavy”;
5. i dati prodotti sono analizzati in finestre di 5 secondi ed utilizzati come input di una rete neurale allo scopo di valutare la presenza o meno di una caduta all’interno della finestra dati esaminata.

I risultati sperimentali sono stati prodotti considerando le varie tipologie di caduta (di fronte, di spalle, laterali e verticali/collasso) producendo una precisione media del 90% circa.

Prototipo indossabile

Per concludere la panoramica sulle soluzioni indossabili è doveroso citare l’articolo [Niazmand et al, 2010], a rappresentanza dei lavori che tentano di integrare i sensori con l’abbigliamento di tutti i giorni. Il prototipo presentato è un capo d’abbigliamento, nello specifico una maglia, nella quale sono stati integrati dei sensori per la rilevazione di cadute. Il sistema si basa su 8 accelerometri situati sulla zona toracica e sulla parte inferiore delle braccia. Tali sensori trasmettono i dati ad una unità elettronica, la quale analizza le misurazioni e valuta l’eventuale caduta. Il prodotto, alimentato a batterie ricaricabili, presenta i vantaggi di essere invisibile agli estranei e, una volta privato dell’alimentazione, di poter essere tranquillamente lavato perché tutte le componenti hardware sono protette.

2.1.3 Soluzioni Ambientali o basate su Videocamera

Un primo caso di soluzioni basate su singola camera è [Rougier et al, 2006]. Nel loro studio gli autori tentano di ricostruire un modello 3D della testa del soggetto a partire dai suoi dati biometrici e dal flusso video. Riconoscendo che la testa di un individuo è ben approssimata tramite un’ellissi, allora l’algoritmo proposto procede nel seguente modo:

1. viene individuata sul frame la posizione del capo del soggetto attraverso ricerche successive, a partire dall’ultima posizione riscontrata e verso le zone limitrofi;
2. viene stimata la nuova posizione nello spazio dell’ellissi 3D, attraverso la posizione sull’immagine e sui dati biometrici già memorizzati;
3. vengono calcolate la velocità verticale V_v ed orizzontale V_h a partire dai modelli nello spazio;

4. una caduta è rilevata se i valori V_v e/o V_h superano valori di soglia prefissati.

Il prototipo illustrato presentava, al momento della pubblicazione dell'articolo, alcune problematiche, prima fra tutte la necessità di inizializzare manualmente la posizione della testa sul flusso video. Altri aspetti negativi si presentavano nella differenza fra modello stimato e realtà, soprattutto durante le fasi finali della caduta. Questo elemento non alterava il successo del sistema perché la notifica di allarme veniva già segnalata nei frame precedenti. Le simulazioni hanno evidenziato che il sistema è in grado di riconoscere buona parte delle cadute dalle normali attività quotidiane.

L'approccio utilizzato nei prossimi lavori vuole essere più legato ad un'analisi della figura umana nel suo complesso, andando ad individuare la silhouette del soggetto e la sua modifica nella successione dei frame. Per tali ragioni, le immagini sono spesso pre-processate con algoritmi di background subtraction, rilevazione del bordo della figura umana e suo miglioramento tramite processi di erosione, dilatazione ed assotigliamento.

In [Shieh, Huang, 2009] l'attenzione viene posta non solo sulla generazione di un fall detector tramite camera, ma anche sulla possibilità di analizzare più flussi video contemporaneamente grazie al multi-threading ed ai nuovi processori multi-core. Per quanto concerne la rilevazione della caduta, il contorno della figura in ogni frame viene convertito in una successione di direzioni, che indicano le posizioni relative dei pixel marcati rispetto al precedente, attraversando i punti sul contorno in senso orario. Tali successioni sono, quindi, confrontate con alcune già note e presenti in un database. L'algoritmo è stato, quindi, suddiviso in 4 thread per poter gestire altrettanti flussi video in pipeline. La fase di sperimentazione è stata suddivisa in tre parti: innanzitutto sono state confrontate le prestazioni di un sistema basato su parallelismo con 4 thread che gestiscono in modo separato i 4 flussi video, quindi il pipeline con thread a carico di lavoro bilanciati e non. È emerso che, nonostante l'uso di thread provochi aumenti nelle comunicazioni, se bilanciati, il throughput generato supplisce a tale carenza. Il sistema di rilevazione cadute presenta un tasso di precisione superiore al 90% ed un ritardo nell'inoltrare l'allarme di circa 13 secondi (di cui 10" utilizzati per verificare l'effettivo mantenimento della "posizione caduta").

La tecnica proposta in [Chen et al, 2010] elabora le informazioni della silhouette di ogni immagine per generare un'approssimazione dello skeleton della persona. Quindi, i dati così generati possono essere utilizzati per valutare la distanza fra skeleton ad istanti temporali successivi e per identificare

la miglior ellissi che approssima la figura riconosciuta, tramite l'analisi dei momenti e dei momenti centrali per il calcolo del centro dell'ellissi, della sua rotazione e della dimensione dei suoi assi. Tutti questi dati vengono, infine, aggregati per generare tre parametri che indicano la variazione di posizione e di dimensioni della figura su intervallo di tempo. Quando tali valori non sono conformi alle soglie stabilite, si ipotizza vi sia stata una brusca variazione nella forma dell'individuo, che rappresenta una delle caratteristiche per valutare l'avvenuta caduta. La tecnica proposta è stata testata su un insieme di azioni ordinarie e di cadute, producendo un valore di detection rate di poco superiore al 90% e di false alarm rate intorno al 6%. L'aspetto negativo riscontrato è l'incapacità di distinguere cadute "rallentate" e movimenti volontari troppo veloci (come cadere appoggiandosi ad una parete o sedersi bruscamente su una sedia).

Rispetto agli approcci fin'ora analizzati, quello in [Tzeng et al, 2010] prevede l'utilizzo di un sistema misto che unisce sensori a pressione posti sul pavimento con immagini ad infrarossi. Il prototipo prevede un controllo costante sui valori di pressione prodotti dai sensori LVDT (linear variable differential transformer) e se la differenza di pressione, in due istanti successivi, è superiore ad un valore di soglia prestabilito, allora viene attivata la videocamera ad infrarossi. Dalle immagini prodotte vengono individuati gli elementi che si differenziano dallo sfondo e per ognuno vengono calcolati gli istogrammi delle deviazioni standard sui pixel analizzati per righe e colonne, nonché l'aspect ratio (rapporto tra larghezza e altezza dell'immagine). I valori prodotti dai sensori di pressione e dalle immagini elaborate sono utilizzati come input per un sistema esperto allo scopo di determinare la possibile caduta del soggetto. I risultati sperimentali hanno prodotto valori di accuratezza, sensitivity e specificity superiori al 90%; hanno anche evidenziato come l'approccio utilizzato sia preferibile in termini di privacy ma che i suoi risultati sono fortemente influenzati dall'abbigliamento utilizzato dalla persona inquadrata (un abbigliamento troppo pesante potrebbe impedire il riconoscimento del soggetto rispetto al background).

L'approccio proposto in [Foroughi et al, 2008] vuole generare un sistema robusto per la rilevazione di cadute e, più in generale, il riconoscimento di posizioni, combinando dati provenienti dall'elaborazione dei frame video e l'uso di Support Vector Machine per classificare le varie posizioni. In particolare, dopo aver individuato la silhouette umana all'interno del frame ne vengono estratti tre parametri:

- l'ellissi che meglio approssima la figura umana rilevata, considerando il

centro, l'angolo di rotazione θ e le misure degli assi (a, b);

- gli istogrammi di proiezione della figura sugli assi orizzontale e verticale normalizzati;
- la posizione della testa ed il suo andamento temporale.

Questi dati sono stati applicati su SVM ed i risultati sperimentali hanno prodotto valori di sensitivity attorno al 90% e di specificity del 95%.

Un ultimo esempio di sistema è quello proposto in [Snoek et al, 2007] per la rilevazione delle cadute di soggetti che scendono le scale. Il metodo proposto prevede l'utilizzo di una videocamera per catturare il flusso video del soggetto mentre si muove sulle scale. Da tali dati vengono ricavate le informazioni sulla posizione dei piedi (se sono visibili nelle immagini) o una stima di tale posizione tramite una rete bayesiana. Parte dei pattern sono stati utilizzati per allenare delle reti di Markov nascoste al fine di ottenere dei modelli sulle posizioni dei piedi nel caso in cui un soggetto scenda correttamente le scale. Il confronto con tali stime viene, infine, utilizzato per valutare il comportamento dell'utente. Fintanto che il soggetto produce un comportamento considerato simile a quello previsto, allora l'azione si considera compiuta in modo normale, se invece la somiglianza è assente, allora una caduta è ritenuta probabile.

Sebbene la fase sperimentale presenti dei risultati favorevoli, è stata riscontrata la difficoltà del sistema a determinare la posizione dei piedi se tale zona è parzialmente coperta da altre persone, o se il modello di scarpe non corrisponde a quelli utilizzati in fase sperimentale. Inoltre, l'utilizzo di alcuni esempi per allenare la catena di Markov può portare il sistema a classificare in modo non corretto flussi video che, semplicemente, non sono conformi a quanto atteso (come nel caso in cui si salti un gradino mentre si scendono le scale).

2.2 Sistemi di rilevamento azioni

Secondo la definizione della comunità medica, con il termine "Attività del vivere quotidiano" (in inglese Activity of Daily Living, ADL) si intendono tutte quelle azioni che vengono comunemente compiute durante la giornata tipo di un soggetto. Queste includono la cura personale (tra cui l'alimentazione, la pulizia personale, il vestirsi), il lavoro, le attività casalinghe ed il tempo libero. A questi comportamenti è data particolare importanza perché sono un buon indicatore del livello di indipendenza delle persone anziane.

Fra i vari metodi utilizzati per valutare il grado di indipendenza dei soggetti anziani e basati sull'osservazione delle attività quotidiane degli stessi, è possibile citare le proposte di:

- Katz e Akporn, che valuta attività quali lavarsi, vestirsi, deambulare;
- Lawton e Brody, che si basa sulle attività che prevedono l'interazione con gli oggetti, quali comunicare telefonicamente, fare shopping, preparare degli alimenti, autogestirsi in casa, muoversi con mezzi propri o pubblici, gestirsi con i medicinali;

le tabelle riepilogative di tali metodi sono riportate in appendice.

Stimare le azioni che vengono compiute da una persona nell'arco di una giornata è un'attività utile per quanti si prendono cura della stessa ed è un compito che può essere eseguito integrando diverse tecnologie.

Un primo esempio di sistema per il riconoscimento delle ADL è il progetto PROACT (Proactive Activity Toolkit) descritto in [Philipose et al, 2004]. Il suo scopo è riconoscere le attività compiute da un soggetto tramite la sequenza degli oggetti da lui utilizzati.

Il sistema è costituito da un guanto dotato di lettore attivo di RFID che permette di individuare gli oggetti etichettati con cui l'utente interagisce. Sempre in pre-processing vengono ricavate le sequenze di oggetti necessari per svolgere particolari compiti tramite inserimento manuale e data-mining sul web. In run-time la sequenza di oggetti con la quale si interagisce viene valutata tramite una rete bayesiana dinamica (allenata con i dati ricavati), allo scopo di inferire le informazioni sulle azioni svolte.

Il sistema è stato testato in un periodo di sei mesi in una vera abitazione. Posta in osservazione la durata media dei dispositivi installati, è stato riscontrato un tempo di vita media di circa 2-3 mesi delle etichette RFID. Successivamente è stata verificata la capacità del lettore di riconoscere gli oggetti con i quali si interagisce; infine è stata valutata la capacità di riconoscere 14 ADL ottenendo le seguenti misurazioni:

- Precision: probabilità che una data inferenza sull'attività sia corretta

$$TP/(TP + FP) = 88\%;$$

- Recall: probabilità che PROACT possa inferire correttamente una data attività

$$TP/(TP + FN) = 73\%.$$

Un secondo esempio è proposto nei documenti [Berenguer et al, 2008], [Noury et al, 2010] e [Noury et al, 2011].

L'idea alla base del progetto è monitorare lo stato complessivo del “sistema casa” tramite un unico dispositivo, posizionato all'inizio del sistema di fornitura di energia elettrica, che ha lo scopo di monitorare i consumi. È stato rilevato, infatti, che l'accensione e lo spegnimento di diversi dispositivi all'interno della casa genera diverse firme nel segnale potenza e tali modelli possono essere utilizzati per valutare quali dispositivi sono stati accesi/spenti.

Nella fase di installazione del dispositivo di controllo della potenza, vengono memorizzate le firme dei singoli dispositivi. Le varie attività ADL sono state, quindi, collegate ai vari dispositivi elettrici andando a definire una misura di correlazione fra attività e strumento tramite un parametro $p_{ij} \in \{0, 1, 2, 3\}$ che indica quanto alta è la connessione fra lo strumento i e l'attività j . Per esempio fra la colazione e la macchina del caffè vi è un legame molto stretto e quindi il parametro $p_{caffè,colazione} = 3$ mentre $p_{lucecucina,colazione} = 1$ perché la luce della cucina può essere accesa anche per altri motivi, come la cena o la lettura del giornale. Definito come $r_i \in \{0, 1\}$ l'indicazione se l'utensile i è acceso o spento, allora in un ogni istante t si presume in esecuzione l'attività j che massimizza

$$ADL_j = \max \sum_{i=0}^n p_{ij} * r_i.$$

Le stime così ottenute per ogni istante vengono, quindi, raffinate confrontandole con le letture precedenti e successive al fine di produrre letture continue delle ADL nei vari intervalli di tempo.

Il sistema è stato valutato su soggetti anziani che risiedono nella propria abitazione da soli. L'esperimento è stato eseguito per ogni soggetto per 90 giorni. Nel suo complesso il sistema proposto viene percepito come non invasivo, ma allo stesso tempo anche molto informativo per coloro che assistono l'anziano. È pensabile un miglioramento del sistema per meglio adattarlo alle necessità del singolo utente ma questo richiede una raccolta di informazioni che possono essere percepite come troppo invasive.

Un ultimo esempio è proposto in [Fleury et al, 2009] e [Fleury et al, 2010] è un esempio di riconoscimento di ADL tramite diversi sensori installati in ambiente domestico.

A questo scopo è stato utilizzato un vero appartamento di $47m^2$ composto di cucina, soggiorno, bagno, camera da letto ed equipaggiato con

- sensori infrarossi, per la localizzazione di persone nell'ambiente;

- sensori di contatto per il frigorifero, gli armadi ed i cassettoni;
- microfoni per la classificazione dei suoni o del parlato;
- un sensore composto da acceleratore triassiale per la classificazione delle posizioni e dei movimenti;
- videocamera per avere indicazioni temporali sulle ADL

ed utilizzate per il machine learning supervisionato.

Sono state selezionate diverse categorie di attività da rilevare quali dormire, preparare e fare colazione, cambiarsi d'abito, igiene personale, comunicazione. Per ogni sensore sono stati preprocessati i dati per ottenere delle features da utilizzare per il training ed il testing di SVM. Per esempio dall'accelerometro si è ricavata la percentuale di tempo spesa nelle varie posizioni (in piedi, seduto, disteso) e per la camminata.

I dati raccolti corrispondono alle simulazioni compiute da 13 volontari in un intervallo di un'ora ciascuno e la classificazione tramite Support Vector Machines con kernel RBF (o gaussiano) portando a rapporti di classificazione corretta intorno al 86%.

In [Fleury et al, 2010] si è tentato di migliorare le prestazioni del sistema tramite l'aggiunta di conoscenze quali la stanza utilizzata e la finestra temporale durante la quale l'azione ha luogo. A dispetto delle aspettative il lavoro proposto ha dimostrato che le prestazioni generali del sistema non sono migliorate, anzi in alcuni casi sono addirittura peggiorate. Tali risultati sono dovuti a problemi relativi ai sensori di posizione che, in alcuni casi, non individuavano anche per lunghi periodi temporali, gli spostamenti del volontario. Questo fenomeno ha portato a mal-classificare frame temporali e tale fenomeno ha prodotto un peggioramento delle prestazioni.

2.3 Sistemi per la gestione dei medicinali

Il problema del rispetto delle prescrizioni mediche è un'importante aspetto della vita dell'anziano ed un caso particolare del controllo delle attività quotidiane appena descritto. Per controllare l'assunzione dei farmaci sono stati proposti diversi dispositivi in commercio ed in letteratura.

Strumenti come il "portapillole" a cadenza settimanale sono stati pensati per catalogare e dosare le medicine da prendere. Organizzati in vari contenitori, a seconda del giorno e della fascia oraria, sono un utile strumento per aiutare gli anziani a non sbagliare i dosaggi (perché già predisposti all'inizio della settimana) e per fornire conferma dell'avvenuta ingestione (tramite la

loro assenza dal contenitore).

In progetto proposto in [Vinjumur et al,] si appoggia, invece, su un contenitore (in cui conservare tutti i medicinali) ed un servizio web per il monitoraggio, anche da remoto, delle azioni dell'anziano.

Il cofanetto dei medicinali è dotato di sensori per verificare se è aperto o chiuso, di un lettore RFID, e di una bilancia di precisione per valutare il peso del contenuto. Ad ogni medicinale è, inoltre, associata una targhetta RFID per permetterne il riconoscimento e la presenza (o l'assenza) all'interno del cofanetto.

Il sistema opera nel seguente modo:

- se l'utente necessita di un medicinale, apre il cofanetto e ne estrae il contenitore. Queste azioni vengono rilevate tramite lo switch aperto / chiuso e dall'incapacità del lettore di rilevare la targhetta RFID del farmaco corrispondente;
- quanto l'utente ha preso il medicinale lo ripone nel cofanetto. Queste azioni sono percepite tramite il lettore RFID che riprende a rilevare la presenza della confezione, e dalla bilancia che può calcolare la variazione di peso del farmaco e ricavare il numero di pastiglie prese.

Tutte queste informazioni sono rese disponibili tramite un servizio web che permette all'utente di gestirsi con le medicazioni e, a quanti si prendono cura di lui, di verificare le sue assunzioni di farmaci.

Un altro esempio di sistemi per il monitoraggio dei farmaci è proposto in [Ammouri, Bilodeau, 2008] e [Bilodeau, Ammouri, 2009].

Esso si appoggia ad una rete di Petri per descrivere la successione di azioni che il soggetto deve compiere per assumere un farmaco, e su un flusso video per verificare le azioni compiute dall'utente.

In ogni immagine sono rilevati gli elementi fondamentali, come le mani, il volto ed i barattoli dei medicinali, e le occlusioni fra componenti (per esempio quando la mano viene portata alla bocca o spostata sul farmaco) per individuare le interazioni fra le varie componenti. Tali informazioni sono utilizzate per far procedere la rete di Petri e derivare le informazioni sull'avvenuta assunzione del medicinale.

I risultati sperimentali dimostrano una buona capacità del sistema di rilevare faccia e mani con prestazioni superiori al 90%. Il sistema complessivo è stato poi testato su 20 sequenze video, 12 di ingerimento medicinali e 8 altre azioni. I risultati sperimentali hanno portato al corretto riconoscimento di 8

su 12 azioni di assunzione di pillole e 7 su 8 azioni non legate all'ingerimento. L'analisi dei dati ha rilevato che le azioni non correttamente classificate sono dovute, nel primo caso a mancati riconoscimenti di posizioni corrette di mani e testa, mentre nel secondo caso ad una sequenza di riconoscimenti poco verosimigliante (polo scollata che ha prodotto una figura della testa più grande del normale che ha portato ad individuare un falso contatto fra le due mani e la testa).

2.4 Smart House

Con il termine Smart House si intende un ambiente domestico opportunamente progettato ed arricchito di tecnologie avanzate, allo scopo di svolgere funzioni in modo autonomo o di assistere quanti la abitano.

Nell'ambito delle tecnologie assistenziali, le case intelligenti sono viste come un possibile modo per migliorare la vita indipendente degli anziani, garantendone comunque il controllo. Le proposte presenti in letteratura prevedono l'utilizzo di dispositivi che, per esempio, permettano di localizzare l'utente o gli oggetti che normalmente utilizza; controllino gli accessi all'abitazione o che generino statistiche sui segni vitali.

Un'applicazione di tale approccio è il progetto "Gator Tech Smart House", di cui si può trovare una descrizione in [Helal, Chen, 2009]. Progettato in Florida, questo ambiente reale integra i servizi appena descritti per controllare e facilitare il vissuto dei soggetti anziani. Durante il suo sviluppo, è stata verificata la difficoltà di integrare i vari dispositivi in un'unica architettura per permettere il controllo di tutta l'abitazione. Per tale ragione è stato predisposto un nodo hardware, da incorporare nelle singole spine di alimentazione, per individuare il tipo di oggetto ad esso collegato (tramite etichette RFID) ed il suo stato (acceso o spento); questo permette di gestire dispositivi, come le lampade, che possono essere facilmente spostati e sono facilmente reperibili in commercio. Per integrare tutti i componenti è stata creata la piattaforma Atlas; la sua funzione è quella di generare una rappresentazione verosimigliante della casa e di gestire i comportamenti dei vari componenti, siano essi sensori o attuatori.

Sempre in ambito domestico e per utenti indipendenti, è stato ideato il progetto Robocare, descritto per esempio in [Cesta et al, 2011]. L'idea da cui è nato il progetto è fornire un ambiente intelligente che sappia osservare il soggetto anziano che lo abita ed interagire con lui tramite un robot. Algoritmi di people tracking e per l'individuazione delle azioni del soggetto sono

stati pensati per generare lo stato dell'utente. Queste informazioni sono, poi, state utilizzate come input di un sistema di ragionamento temporale basato su CSP che ha lo scopo di verificare il comportamento dell'anziano rispetto a quanto prescritto da familiari o dal medico curante. La fase sperimentale del progetto si è orientata, da un lato per la verifica delle capacità del robot di interagire con l'ambiente e dall'altro per verificare, tramite sondaggi, le reazioni di potenziali utenti alla proposta. È stato riscontrato, in quest'ultimo caso, che i soggetti anziani vedono in modo positivo l'utilizzo delle tecnologie coinvolte nelle situazioni di emergenza, utilizzate per accertare la presenza di malesseri prima di inoltrare la richiesta ai familiari, e per ricordare eventuali appuntamenti; non sono, invece, graditi eventuali consigli generati dal sistema per "meglio investire" il tempo dell'utente.

Ultimo esempio di ambiente intelligente è proposto in [Stanford, 2002]. La struttura presa in esame è un complesso abitativo per anziani capaci di vivere in modo indipendente ma che preferiscono, per affrontare la propria vecchiaia al meglio, un controllo non invasivo. Ogni utente è dotato di un localizzatore per monitorare i suoi spostamenti e gestire gli accessi alle singole zone abitative. Il monitoraggio degli spostamenti permette ai dipendenti della struttura di controllare l'anziano e di intervenire, o richiedere l'aiuto del personale medico, solo nei casi di necessità. Fra gli altri servizi resi disponibili vi sono il monitoraggio del peso e degli spostamenti notturni, tramite sensori di pressione posti sotto al letto, l'utilizzo di computer dotati di touch screen per la gestione delle comunicazioni, come mail o videocchiamate, ed un database centralizzato per la memorizzazione dei dati dei singoli pazienti (al fine di fornire al medico curante lo storico dei parametri vitali dei singoli utenti).

2.5 Altre proposte di tecnologie assistenziali

Per concludere questa panoramica sulle tecnologie assistenziali è doveroso citare esempi di altri sistemi per l'assistenza degli anziani. Nei casi sotto riportati vengono rilassate le ipotesi viste in precedenza, come il buono stato psico-fisico dell'utente o la funzione di controllo del sistema stesso; quelli che si ottengono sono sistemi con obiettivi più specifici e scenari limitati. Nei primi due casi i soggetti in esame presentano gravi forme di malattie mentali, nel terzo l'attenzione viene posta su soggetti con scarsa mobilità e costretti in sedia a rotelle, seguono infine due casi in cui i robot vengono utilizzati per supportare il personale di case di riposo in attività a basso rischio o per ipotizzare nuovi sistemi di tele-presenza.

SenseCam

Lo studio proposto in [Hodges et al, 2006] osserva come in letteratura siano presenti vari sistemi pensati per soggetti con gravi problemi di memoria ma il cui obiettivo principale è quello di garantire l'esecuzione di azioni future e non sul ricordare quanto è già stato vissuto. Per studiare una possibile applicazione tecnologica che venga in aiuto di tale categoria di anziani è stato proposto SenseCam, un dispositivo portatile che ha lo scopo di riprendere i momenti più significativi della vita dell'utente da utilizzare in un secondo tempo per "allenare la mente". SenseCam è costituito da un piccolo apparecchio, da portare come un pendente, dotato di fotocamera con lente fish-eye, sensori e memoria. Una volta acceso, il dispositivo scatta foto con cadenza periodica, nonché ogni volta che vi è un possibile cambio di ambiente (rilevato dai sensori) o su richiesta dell'utente (come una normale fotocamera). L'apparecchio è associato ad un software installato su computer allo scopo di salvare i dati raccolti, modificare le immagini (per eliminare le distorsioni dovute alla lente che permette di avere una panoramica più ampia dell'ambiente circostante ma che, di contro, produce immagini poco chiare), integrarle con commenti forniti dall'utente e rivedere gli eventi a cui l'anziano ha partecipato. In fase sperimentale si è riscontrato che rivedere le immagini ad intervalli regolari di tempo nel periodo successivo al momento vissuto, permette all'utente di allenare la mente e mantenere vivi i ricordi, con risultati migliori rispetto a strumenti come il diario personale, contrastando il normale decadimento della memoria.

Automated handwashing assistance

Lo studio proposto in [Hoey et al, 2010] è orientato al garantire una sorveglianza attiva a soggetti che soffrono di gravi malattie mentali (come l'Alzheimer) nell'azione, apparentemente semplice, di lavarsi le mani. L'input di questo sistema di controllo sono le riprese delle azioni svolte dall'utente, allo scopo di comprendere il suo comportamento e di suggerire eventuali modifiche tramite messaggi audio o video. Fra i problemi affrontati per la realizzazione di questo progetto vi sono la necessità di tracciare, tramite il supporto di una rete bayesiana, i movimenti delle singole componenti di interesse (come le mani o l'asciugamano) che per loro natura non presentano una forma unica, ma variano nel tempo, e determinare le eventuali reazioni del sistema ai gesti dell'utente, appoggiandosi ad un processo di Markov parzialmente osservabile. I test compiuti si sono concentrati su tre problemi successivi: inizialmente è stato verificato il funzionamento della componente che individua la posizione delle mani, quindi che gli output prodotti dal siste-

ma rispecchino un comportamento atteso ed infine si è passati ad uno studio sperimentale in un istituto per soggetti che soffrono di forme medio-acute di demenza.

Robotic Wheelchair

Lo scopo del progetto proposto in [Yanco, 2001] è quello di agevolare gli spostamenti in ambiente esterno per i soggetti costretti su sedia a rotelle. Il prototipo è stato dotato di una videocamera stereo per raccogliere le immagini di quanto si trova di fronte al dispositivo. Gli elementi che vengono presi in considerazione sono due: la presenza di ostacoli per modificare la velocità del dispositivo in loro presenza ed il percorso che la sedia a rotelle ha preso per scelta dell'utente. Sfruttando le informazioni di profondità della telecamera stereo vengono rilevati i punti in cui vi è una forte variazione di profondità che vengono marcati come ostacoli. Per semplicità si è ipotizzato che gli ostacoli presenti sul marciapiede siano, con alta probabilità degli oggetti mobili, come persone o animali, che tendono a muoversi più velocemente della sedia stessa, nonché ad evitarla. Per questa ragione quando un ostacolo è rilevato nel centro della traiettoria del veicolo, allora la sedia tenderà a ridurre la sua velocità e, se necessario, anche a fermarla. Il secondo problema studiato è l'aiuto alla navigazione per l'utente. In questo caso, tramite l'immagine rilevata dalla videocamera, vengono individuati i potenziali bordi del marciapiede e sulla base di tali stime vengono valutate eventuale variazioni della direzione, affinché l'utente identifichi la direzione da seguire e sia la sedia a rotelle a muoversi seguendo le indicazioni date. L'utilizzo congiunto di tali tecniche ha dimostrato, in fase sperimentale, che il soggetto riduce gli sforzi compiuti per manovrare la sedia a rotelle durante i suoi spostamenti.

Pearl

Scopo del progetto proposto in [Pineau et al, 2003] ed in [Pollack, 2005] è quello di investigare il possibile utilizzo di un robot per gestire gli ospiti di una casa di riposo, in modo da affiancare le infermiere nelle attività meno complesse e problematiche. Il sistema è costituito da diversi moduli che permettono di gestire le informazioni sugli impegni dei vari pazienti (gestiti tramite il software Autominder e basato sul un CSP), di localizzare e seguire i singoli soggetti, di interagire verbalmente e tramite touch screen... A supervisionare il tutto vi è un controllore che determina l'ordine delle azioni da compiere; se per esempio un paziente deve compiere una data attività, il robot dovrà individuare la propria posizione, quella del soggetto da contattare, quindi raggiungerlo ed iniziare l'interazione. Il sistema è stato testato

per alcuni giorni in una casa di riposo, inizialmente con lo scopo di verificare le sue capacità di interazione con la popolazione residente e successivamente come agente che ricordi a singoli pazienti del loro incontro con il dottore e che li guidi verso la loro destinazione. Tra i principali problemi che sono stati riscontrati vi sono stati la difficoltà di interagire verbalmente con il robot e la sua incapacità di adattare la propria velocità quando un anziano lo deve seguire.

ExCite

Come indicato in precedenza, fra gli obiettivi delle tecnologie assistenziali vi è la ricerca di strumenti che permettano di mantenere un buon grado di interazione sociale. Con tale proposito viene sviluppato ExCITE, di cui si può trovare una descrizione in [Cesta et al, 2010], che si prefigge di ridurre il senso di solitudine e di isolamento dei soggetti anziani tramite l'utilizzo di un robot modificato per permettere la tele-presenza. Alla base dello studio vi è l'idea di adattare i sistemi attualmente utilizzati per l'interazione in ambienti pericolosi affinché possano essere utilizzati in ambiente domestico per far visita agli anziani. Il sistema si compone di due componenti installate nel domicilio del soggetto da visitare e nelle case dei suoi familiari. Nel primo viene posizionato il robot Giraff che ha la capacità di muoversi in modo sicuro nell'ambiente domestico e di fornire gli strumenti per le video-chiamate; nei secondi viene installato un software per interagire con il robot sui singoli computer di casa. Quando un familiare vuole far visita all'anziano, tramite il suo terminale si connette al robot ed ha la possibilità vedere il soggetto a cui si fa visita e con la possibilità di muoversi all'interno della residenza. Questo permette di simulare in modo più realistico una visita di persona, senza i problemi di staticità di altri strumenti come Skype. Particolarità del progetto è la modalità di sviluppo dello stesso: per rendere il prodotto finale più vicino alle necessità dei potenziali utenti, gli sviluppatori hanno predisposto dei cicli di test e modifica per modificare il prototipo secondo i feedback ottenuti. Per ogni soggetto del test, infatti, vengono proposti almeno due cicli di utilizzo: il primo per familiarizzare con lo stato attuale dello strumento ed il secondo per valutare sul campo le modifiche richieste nel periodo precedente.

Capitolo 3

Sistema di Ragionamento Temporale Fuzzy

Analizzando il linguaggio parlato si può notare come le informazioni temporali siano uno degli elementi che lo caratterizzano. Non solo i tempi verbali ci permettono di fornire implicitamente informazioni sugli eventi che coinvolgono la nostra realtà ma è possibile indicare quando alcuni avvenimenti sono occorsi, quanto sono durati e che relazioni sono presenti fra di loro.

Per esemplificare tali concetti si considerino le seguenti espressioni:

- il pranzo viene servito dopo la colazione;
- l'infuso di erbe deve essere preso fra le 12:00 e le 20:00;
- l'aspirina deve essere presa a stomaco pieno e preferibilmente entro venti minuti dalla fine della cena;
- cenare è un'attività che dura circa 1 ora.

Questi semplici esempi ci permettono di intuire alcuni degli elementi essenziali che possono caratterizzare un sistema di ragionamento temporale, e che sono stati recepiti nel sistema fuzzy sviluppato presso il Dipartimento di Ingegneria dell'Informazione dell'Università di Padova.

In riferimento agli eventi ci si può esprimere sia in termini di punti che di intervalli rispetto all'asse temporale; con il termine "colazione", ad esempio, ci si riferisce all'intero arco di tempo in cui l'azione viene compiuta mentre con "fine della cena" si vuole indicare l'istante esatto in cui l'azione ha termine. Come già indicato in [Allen, 1983] fra le due categorie di elementi vi sono dei legami molto forti: un intervallo può essere espresso tramite i suoi punti di inizio e fine, e viceversa un punto può essere visto come estremo di un

intervallo. Mantenere entrambe le tipologie permette di rendere il sistema più espressivo e le informazioni in esso contenute di più facile lettura.

Allo stesso modo è possibile osservare come le relazioni fra gli eventi possano essere espresse in modo diverso. Mentre il primo esempio indica una relazione di precedenza (ma senza indicare quanto tempo intercorra fra la colazione ed il pranzo) nelle altre vengono indicate le tempistiche da seguire per svolgere tali attività (espresse sia in termini di finestra temporale valida sia in termini di durata). Tali legami possono, quindi, essere di tipo qualitativo, quando indicano solo un ordine di precedenza fra gli elementi, o di tipo quantitativo, se presentano delle informazioni numeriche.

Ritornando agli esempi sopra citati è, inoltre, possibile riflettere sul fatto che non tutte le indicazioni temporali hanno lo stesso peso: è diverso affermare che un'azione dura 1 ora dall'indicare che ne dura circa 1; la prima ipotizza una tempistica esatta, la seconda può essere vista come una media di valori che converge, in un qualche modo, a quello indicato. Questo suggerisce la necessità di integrare quanto detto finora con un parametro che permetta di descrivere aspetti quali la vaghezza, l'imprecisione, le diverse priorità e i gradi di preferenza; tutto questo è possibile se si introduce la possibilità di modellare le informazioni in modo fuzzy.

Riorganizzando le osservazioni fin qui raccolte in forma grafica è possibile notare come si possano facilmente rappresentare in un grafo orientato, nel quale punti ed intervalli fungono da vertici mentre le relazioni che li vincolano possono essere rappresentate come archi orientati.

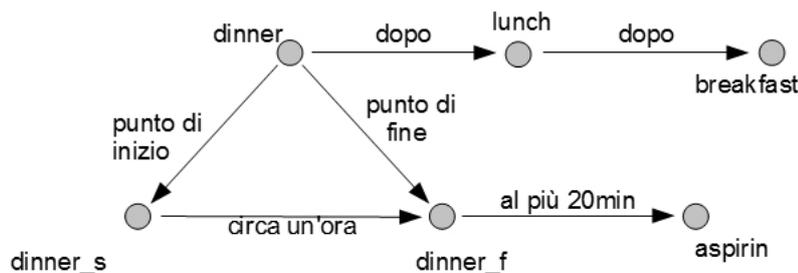


Figura 3.1: Esempio di come le informazioni temporali possono essere tradotte in un grafo orientato

Molte delle ipotesi appena citate hanno permesso, nel corso degli anni, ad autori diversi di riflettere su come sia possibile integrare tali dati in un unico sistema di ragionamento, al fine di ottenere una potenza informativa maggiore.

Il sistema sviluppato dal Dipartimento di Ingegneria dell'Informazione dell'Università di Padova si ispira alle ricerche proposte in [Allen, 1983], in [Vilain et al, 1989], in [Meiri, 1995] ed è possibile trovarne una descrizione in [Badaloni et al, 2007], per esempio. Tale strumento, noto come "Sistema di Ragionamento Temporale Fuzzy" permette di gestire punti ed intervalli temporali, relazioni qualitative e quantitative ed indicazioni fuzzy sul grado di preferenza dei vincoli.

3.1 Relazioni Qualitative

3.1.1 Definizione di relazione qualitativa

Una relazione qualitativa fra due oggetti temporali i e j , siano essi punti o intervalli, consiste in una disgiunzione del tipo

$$(i \ r_1[\alpha_1] \ j) \vee (i \ r_2[\alpha_2] \ j) \vee \dots \vee (i \ r_n[\alpha_n] \ j)$$

che, per semplicità di notazione, può essere espressa anche nella forma

$$i \ \{r_1[\alpha_1], r_2[\alpha_2], \dots, r_n[\alpha_n]\} \ j = i \ R_{ij} \ j.$$

Il parametro r_k è una delle possibili relazioni elementari esistenti fra i e j e può essere ricondotto ad una delle seguenti tipologie:

- relazioni intervallo - intervallo (II): definite in [Allen, 1983], sono le relazioni che possono intercorrere tra una coppia di intervalli. Sono costituite dalle relazioni di *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, le corrispondenti relazioni inverse *after*, *met-by*, *overlapped-by*, *started-by*, *contains*, *finished-by*, nonché la relazione *equals-to* di uguaglianza, per un totale di 13 (vedi tabella 3.1).
- relazioni punto - punto (PP): introdotte in [Vilain et al, 1989], individuano tutti i modi in cui si possono ordinare due punti. Esse sono *before*, *equals-to*, *after* (vedi tabella 3.2).
- relazioni punto - intervallo (PI): definite in [Meiri, 1995], rappresentano i 5 modi in cui è possibile posizionare sull'asse temporale un punto rispetto ad un intervallo. Esse sono *before*, *starts*, *during*, *finishes*, *after*. Invertendo le relazioni appena descritte si hanno le relazioni intervallo - punto (IP) che sono *after*, *started-by*, *contains*, *finished-by* e *before* (vedi tabella 3.3).

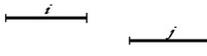
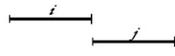
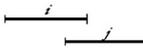
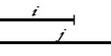
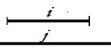
<i>Relazione (i, j)</i>	<i>Rappresentazione</i>	<i>Relazione Inversa (j, i)</i>
before { b }		after { a }
meets { m }		met-by { ml }
overlaps { o }		overlapped-by { oi }
starts { s }		started-by { si }
during { d }		contains { di }
finishes { f }		finished-by { fi }
equals-to { eq }		equals-to { eq }

Tabella 3.1: Relazioni di tipo intervallo - intervallo

<i>Relazione (i, j)</i>	<i>Rappresentazione</i>	<i>Relazione Inversa (j, i)</i>
before { b }		after { a }
equals-to { eq }		equals-to { eq }

Tabella 3.2: Relazioni di tipo punto - punto

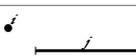
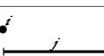
<i>Relazione (i, j)</i>	<i>Rappresentazione</i>	<i>Relazione Inversa (j, i)</i>
before { b }		after { bi }
starts { s }		started-by { si }
during { d }		contains { di }
finishes { f }		finished-by { fi }
after { a }		before { ai }

Tabella 3.3: Relazioni di tipo punto - intervallo

Il valore $\alpha_k \in [0, 1]$ permette di introdurre il grado fuzzy di appartenenza della relazione elementare r_k al vincolo fra i e j . Si può definire come grado di soddisfacimento del vincolo i, j il valore

$$\text{deg}(R_{ij}) = \max_{k=1, \dots, n} \alpha_k.$$

Come già accennato in precedenza, l'idea di introdurre tale informazione serve a modellare quegli aspetti di vaghezza, imprecisione o semplicemente di preferenza, che sono tipiche del linguaggio parlato e costituiscono il punto focale degli studi sviluppati a partire da [Badaloni et al, 2003].

È doveroso far notare che i vincoli qualitativi proposti in letteratura in [Allen, 1983], [Vilain et al, 1989] e [Meiri, 1995] trovano una naturale rappresentazione nel sistema di ragionamento temporale fuzzy, in quanto le relazioni r_k che appartengono al vincolo presentano $\alpha_k = 1$ mentre le altre $\alpha_k = 0$.

Secondo la teoria degli insiemi fuzzy, per ogni coppia (i, j) dovrebbe essere indicato il grado di appartenenza al vincolo di ogni relazione elementare; per non appesantire la notazione è stata adottata la convenzione secondo la quale le relazioni a grado 0 possono non essere indicate mentre quelle a grado 1 il valore α_k può essere omissso.

3.1.2 Esempi di vincoli qualitativi

Per comprendere al meglio la potenza espressiva delle relazioni qualitative appena descritte riprendiamo un paio degli esempi visti in precedenza ed osserviamo come questi possano essere tradotti in relazioni qualitative.

Un primo esempio di facile traduzione è l'espressione "il pranzo viene servito dopo la colazione". Considerando la presenza di due intervalli etichettati come "breakfast" e "lunch", la relazione che li lega è di tipo "before" perché le azioni avvengono una dopo l'altra e fra le due esiste un intervallo di tempo non nullo che le separa. Il vincolo può, quindi, essere riscritto nella forma

$$\text{breakfast } \{ b \} \text{ lunch.}$$

Un secondo interessante spunto di riflessione può essere l'espressione "l'aspirina deve essere presa a stomaco pieno e preferibilmente entro venti minuti dalla fine della cena". Premettendo la doverosa osservazione che, con il solo utilizzo delle relazioni qualitative, non è possibile modellare l'aspetto dei "venti minuti", si possono comunque fare alcune considerazioni sulle relazioni che intercorrono fra gli intervalli della cena e del prendere l'aspirina, indicati come "dinner" ed "aspirin intake". Ragionando sul senso della frase è possibile fare le seguenti osservazioni sulla relazione

$$\text{aspirin intake} \rightarrow \text{dinner}$$

- le opzioni “after” e “met-by” sicuramente rispettano l’espressione;
- le relazioni “before”, “meets”, “overlaps”, “starts” sono da escludere perché non rispettano la parte della prescrizione che prevede di prendere il medicinale “a stomaco pieno”;
- le opzioni “equals-to”, “started-by”, “finished-by”, “contains” sono da escludere per ragioni del senso comune, dato che sicuramente la cena ha una durata superiore dell’azione di ingerire una pillola;
- le relazioni “during”, “finish” ed “overlapped-by” si possono considerare valide, visto che presuppongono che il soggetto abbia già iniziato a cenare prima di prendere l’aspirina. D’altra parte si può desiderare di differenziare questi casi dai primi, perché da un punto di vista logico si ha lo stomaco “più pieno” una volta concluso il pasto. Per questa ragione tali relazioni si potrebbero considerare valide ma, sfruttando gli aspetti fuzzy del sistema di ragionamento, si può ipotizzare di assegnare loro un valore di preferenza $\alpha \in (0, 1)$ in modo da distinguerle dalle altre opzioni.

Quanto detto può essere formalizzato nel seguente modo:

$$\text{aspirin intake} \left\{ \begin{array}{cccc} b[0], & m[0], & o[0], & s[0], \\ eq[0], & si[0], & fi[0], & oi[0], \\ d[\alpha], & f[\alpha], & oi[\alpha], & \\ a[1], & mi[1] & & \end{array} \right\} \text{dinner} = \\
 \text{aspirin intake} \{ a, mi, d[\alpha], f[\alpha], oi[\alpha] \} \text{dinner}$$

3.1.3 Algebra Qualitativa Fuzzy

Al fine di facilitare l’elaborazione dei dati sui vincoli qualitativi è stata definita un’ Algebra Qualitativa Fuzzy (QA^{fuz}) i cui elementi sono tutti i possibili vincoli qualitativi mentre le operazioni definite sulle relazioni sono quelle di:

- inversione, operatore unario identificato dal simbolo $^{-1}$;
- intersezione, operatore binario identificato dal simbolo \otimes ;
- unione, operatore binario identificato dal simbolo \oplus ;
- composizione, operatore binario identificato dal simbolo \circ .

Le definizioni che sono state adottate per le operazioni in QA^{fuz} estendono le loro controparti proposte in [Allen, 1983], [Vilain et al, 1989] e [Meiri, 1995], mantenendo invariati i risultati riportati in tali articoli.

Operazione di Inversione

Si considerino due oggetti temporali i e j vincolati dalla relazione

$$i R_{ij} j = i \{r_1[\alpha_1], r_2[\alpha_2], \dots, r_n[\alpha_n]\} j$$

allora il vincolo inverso $R_{ji} = R_{ij}^{-1}$ risulta

$$j R_{ji} i = j R_{ij}^{-1} i = j \{inv(r_1)[\alpha_1], inv(r_2)[\alpha_2], \dots, inv(r_n)[\alpha_n]\} i$$

dove, per ogni $x = 1, \dots, n$, $inv(r_x)$ è la relazione inversa di r_x come indicato nelle tabelle 3.1, 3.2, 3.3. Si può facilmente intuire che l'inverso di un vincolo II è II, quello di un vincolo PP è PP mentre una relazione PI ha per inverso un vincolo IP e viceversa.

Applicando l'operatore appena descritto si possono ricavare i seguenti esempi:

- l'inverso del vincolo II $\{a, mi, d[\alpha], f[\alpha], oi[\alpha]\}$ è $\{b, m, di[\alpha], fi[\alpha], o[\alpha]\}$;
- l'inverso del vincolo PP $\{eq, a[\alpha]\}$ è il vincolo $\{b[\alpha], eq\}$;
- l'inverso del vincolo PI $\{b[\alpha_1], s[\alpha_2], d\}$ è il vincolo IP $\{bi[\alpha_1], si[\alpha_2], di\}$;
- l'inverso del vincolo IP $\{fi[\alpha], ai\}$ è il vincolo PI $\{f[\alpha], a\}$.

Operazione di Intersezione

Si considerino due oggetti temporali i e j sui quali sono definiti i vincoli

$$i R_{ij}^1 j = i \{r_1[\alpha_1^1], r_2[\alpha_2^1], \dots, r_n[\alpha_n^1]\} j$$

$$i R_{ij}^2 j = i \{r_1[\alpha_1^2], r_2[\alpha_2^2], \dots, r_n[\alpha_n^2]\} j$$

allora il vincolo di intersezione $R_{ij} = R_{ij}^1 \otimes R_{ij}^2$ risulta

$$i R_{ij} j = i R_{ij}^1 \otimes R_{ij}^2 j = i \{r_1[\alpha_1], r_2[\alpha_2], \dots, r_n[\alpha_n]\} j$$

dove $\alpha_x = \min\{\alpha_x^1, \alpha_x^2\}$ per ogni $x = 1, \dots, n$.

Alcuni esempi dell'applicazione dell'operatore di intersezione sono:

- dati i vincoli II $R_1 = \{a, mi, d, f\}$ ed $R_2 = \{o, s, d, f\}$ allora

$$R_1 \otimes R_2 = \{d, f\};$$

- dati i vincoli PP $R_1 = \{eq[\alpha], a\}$ e $R_2 = \{b, eq[\alpha]\}$ allora

$$R_1 \otimes R_2 = \{eq[\alpha]\};$$

- dati i vincoli PI $R_1 = \{b[0.7], s[0.9], d[0.3]\}$ e $R_2 = \{b[0.5], s, d[0.6]\}$ allora

$$R_1 \otimes R_2 = \{b[0.5], s[0.9], d[0.3]\};$$

- dati i vincoli IP $R_1 = \{bi, si, di[0.8], fi[0.5]\}$ e $R_2 = \{di[0.3], fi[0.8], ai\}$ allora

$$R_1 \otimes R_2 = \{di[0.3], fi[0.5]\}.$$

Operazione di Unione

Si considerino due oggetti temporali i e j sui quali sono definiti i vincoli

$$i R_{ij}^1 j = i \{r_1[\alpha_1^1], r_2[\alpha_2^1], \dots, r_n[\alpha_n^1]\} j$$

$$i R_{ij}^2 j = i \{r_1[\alpha_1^2], r_2[\alpha_2^2], \dots, r_n[\alpha_n^2]\} j$$

allora il vincolo di unione $R_{ij} = R_{ij}^1 \oplus R_{ij}^2$ risulta

$$i R_{ij} j = i R_{ij}^1 \oplus R_{ij}^2 j = i \{r_1[\alpha_1], r_2[\alpha_2], \dots, r_n[\alpha_n]\} j$$

dove $\alpha_x = \max\{\alpha_x^1, \alpha_x^2\}$ per ogni $x = 1, \dots, n$.

Alcuni esempi dell'utilizzo dell'operatore di unione sono:

- dati i vincoli II $R_1 = \{a, mi, d, f\}$ ed $R_2 = \{o, s, d, f\}$ allora

$$R_1 \oplus R_2 = \{a, d, o, mi, s, f\};$$

- dati i vincoli PP $R_1 = \{eq[\alpha], a\}$ e $R_2 = \{b, eq[\alpha]\}$ allora

$$R_1 \oplus R_2 = \{b, eq[\alpha], a\};$$

- dati i vincoli PI $R_1 = \{b[0.7], s[0.9], d[0.3]\}$ e $R_2 = \{b[0.5], s, d[0.6]\}$ allora

$$R_1 \oplus R_2 = \{b[0.7], s, d[0.6]\};$$

- dati i vincoli IP $R_1 = \{bi, si, di[0.8], fi[0.5]\}$ e $R_2 = \{di[0.3], fi[0.8], ai\}$ allora

$$R_1 \oplus R_2 = \{bi, si, di[0.8], fi[0.8], ai\}.$$

Operazione di Composizione

Si considerino tre oggetti temporali i , j e k sui quali sono definiti i vincoli

$$i R_{ij} j = i \{r_1[\alpha_1^{ij}], r_2[\alpha_2^{ij}], \dots, r_{n_1}[\alpha_{n_1}^{ij}]\} j$$

$$j R_{jk} k = j \{r_1[\alpha_1^{jk}], r_2[\alpha_2^{jk}], \dots, r_{n_2}[\alpha_{n_2}^{jk}]\} k$$

allora il vincolo di composizione $R_{ik} = R_{ij} \circ R_{jk}$ risulta

$$i R_{ik} k = i R_{ij} \circ R_{jk} k = i \{r_1[\alpha_1], r_2[\alpha_2], \dots, r_n[\alpha_n]\} k$$

dove

$$\alpha_z = \max_{x,y:r_z \in \{r_x \circ r_y\}} \min\{\alpha_x^{ij}, \alpha_y^{jk}\}$$

per ogni $x = 1, \dots, n_1$, $y = 1, \dots, n_2$, $z = 1, \dots, n$ e dove l'operazione $r_x \circ r_y$ è definita secondo le tabelle riportate in Appendice A. In modo equivalente è possibile definire l'operazione di composizione come:

$$R_{ik} = R_{ij} \circ R_{jk} = \bigoplus_{x,y} \{r_x \circ r_y[\min\{\alpha_x^{ij}, \alpha_y^{jk}\}]\}$$

per ogni $x = 1, \dots, n_1$, $y = 1, \dots, n_2$.

Per esempio, dati i vincoli II $R_1 = \{a, di\}$ ed $R_2 = \{o\}$ allora:

$$R_1 \circ R_2 = \{a, oi, mi, d, f\} \oplus \{o, di, fi\} = \{a, o, oi, mi, d, di, f, fi\}.$$

In modo analogo dati i vincoli PP $R_1 = \{a, eq[0.5]\}$ ed $R_2 = \{b[0.3], eq\}$ allora:

$$R_1 \circ R_2 = \{b[0.3], eq[0.3], a[0.3]\} \oplus \{b[0.3]\} \oplus \{a\} \oplus \{eq[0.5]\} = \{b[0.3], eq[0.5], a\}.$$

Come si può facilmente intuire dall'analisi delle tabelle riportate in Appendice A, non è possibile applicare l'operazione di composizione su tutte le coppie di relazioni qualitative, ma solo su quelle che hanno in comune un estremo. È per questa ragione che nella tabella A.1 alcune combinazioni, come II - PP, sono state marcate come impossibili tramite il simbolo \circledast .

3.2 Relazioni Quantitative

3.2.1 Introduzione alle relazioni quantitative

Finora è stato osservato come sia possibile definire ed operare con vincoli che permettano di descrivere le posizioni reciproche fra oggetti temporali.

Richiamando, però, gli esempi visti all’inizio del capitolo ci si può rendere conto che la potenza espressiva indicata finora non è sufficiente per modellare completamente relazioni come:

- l’infuso di erbe deve essere preso fra le 12:00 e le 20:00;
- l’aspirina deve essere presa a stomaco pieno e preferibilmente entro venti minuti dalla fine della cena;
- cenare è un’attività che dura circa 1 ora.

Per raggiungere questo obiettivo è necessario vincolare i punti del sistema all’asse temporale, traducendo così gli aspetti quantitativi delle espressioni.

Prendendo spunto dagli esempi proposti è possibile notare come i vincoli che devono essere modellati siano di due tipologie diverse:

- i vincoli assoluti caratterizzati dal fatto che viene vincolata la posizione del punto direttamente sull’asse temporale. Ne è un esempio l’azione del prendere l’infuso di erbe i cui istanti di inizio e fine devono avvenire fra mezzogiorno e le otto di sera;
- i vincoli relativi servono a definire la distanza tra coppie di punti. Ne è un esempio la durata dell’attività del cenare che sottintende una distanza fra l’inizio e la fine dell’azione di circa 1 ora.

Tutti questi problemi possono essere risolti applicando due semplici modifiche alla rete temporale finora descritta. Inserendo un punto fittizio di “origine dei tempi” all’interno della rete è possibile costruire una corrispondenza fra l’asse temporale ed i punti, permettendo, quindi, di modellare i vincoli assoluti come distanze dei singoli dall’origine dei tempi. È, inoltre, possibile costruire una relazione biunivoca fra l’asse temporale e l’insieme dei numeri reali se viene definito un valore di granularità (o “quanto temporale”) che ne permetta la traduzione.

3.2.2 Definizione di relazione quantitativa

Con tali premesse è possibile descrivere una relazione quantitativa fra due punti temporali come un vincolo sulla loro distanza. Siano, per esempio, i e j due punti della rete, allora una relazione fra di loro ha la forma $i R_{ij} j$ che può essere tradotta nell’espressione $j - i = d \in D_{ij}$ dove D_{ij} è un insieme di valori reali. Tale definizione è proposta in [Meiri, 1995] e si ispira ad una versione “crisp” del problema; è necessario passare a [Badaloni et al, 2003] per poter ottenere una formulazione fuzzy dei vincoli quantitativi allo scopo

di modellare espressioni verbali più vaghe ed imprecise. Per raggiungere tale scopo è sufficiente definire una funzione di appartenenza $\mu_{R_{ij}} : \mathbb{R} \rightarrow [0, 1]$ che esprima quanto un generico valore d soddisfa la relazione fra i e j . Ovviamente un grado di appartenenza pari a 0 indica che il valore d non soddisfa la relazione, mentre un grado 1 indica un soddisfacimento pieno della relazione.

Al fine di garantire una buona potenza espressiva ai vincoli quantitativi e mantenere bassi i costi computazionali, le relazioni sono state modellate tramite unione di trapezi T_k nella forma $R_{ij} = \{T_1, T_2, \dots, T_n\}$. Un generico trapezio si presenta come $T_k = \ll_k a_k, b_k, c_k, d_k \gg_k [\alpha_k]$ dove:

- $a_k, b_k \in \mathbb{R} \cup \{-\infty\}$;
- $c_k, d_k \in \mathbb{R} \cup \{+\infty\}$;
- $\alpha_k \in [0, 1]$;
- \ll_k assume la forma (o [;
- \gg_k assume la forma) o];

e secondo le condizioni per cui:

- $a_k \leq b_k \leq c_k \leq d_k$;
- se $a_k = -\infty$ allora $b_k = -\infty$ e $\ll_k = (;$
- se $a_k < b_k$ allora $\ll_k = (;$
- se $a_k = d_k$ allora $\ll_k = [$ e $\gg_k =]$;
- se $d_k = +\infty$ allora $c_k = +\infty$ e $\gg_k =)$;
- se $c_k < d_k$ allora $\gg_k =)$.

Ogni trapezio sottintende una funzione di appartenenza μ_{T_k} che assume i valori:

$$\mu_{T_k}(x) = \begin{cases} 0 & \text{se } x < a_k \vee (x = a_k \wedge \ll_k \text{ is}) \\ & \vee (x = d_k \wedge \gg_k \text{ is}) \vee x > d_k \\ \alpha_k \left(\frac{x-a_k}{b_k-a_k} \right) & \text{se } a_k < x < b_k \\ \alpha_k \left(\frac{x-d_k}{c_k-d_k} \right) & \text{se } c_k < x < d_k \\ \alpha_k & \text{altrimenti} \end{cases}$$

mentre la funzione $\mu_{R_{ij}}$ può essere definita come:

$$\mu_{R_{ij}}(x) = \max_{k=1, \dots, n} \mu_{T_k}(x).$$

Il vantaggio di queste definizioni è la possibilità di modellare forme quali trapezi, rettangoli, triangoli, ma anche trapezi aperti sul semiasse negativo o positivo, come si può notare nella figura 3.2.

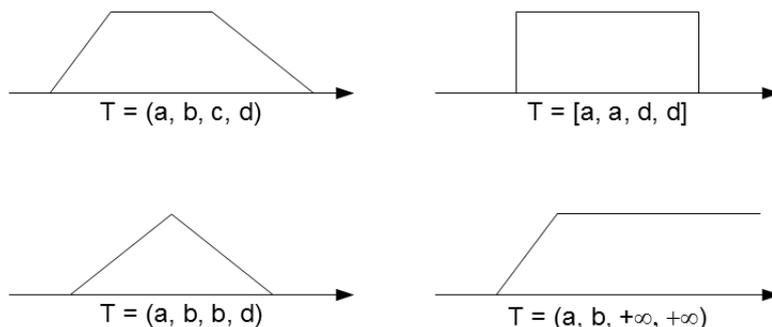


Figura 3.2: Esempi di possibili modellazioni di trapezi

Fra le informazioni utili per descrivere una relazione quantitativa si possono ricordare le seguenti:

- grado di soddisfacimento del vincolo: $deg(R_{ij}) = \max_{k=1,\dots,n} \alpha_k$;
- supporto del trapezio: $supp(T_k) = \{x : \mu_{T_k}(x) > 0\}$;
- nucleo del trapezio: $core(T_k) = \{x : \mu_{T_k}(x) = 1\}$.

3.2.3 Esempio di relazione quantitativa

Per intuire la potenza espressiva della formulazione dei vincoli appena descritta, si ipotizzi di dover modellare l'espressione "John arriva ogni giorno in ufficio fra le 7:30 e le 8:00 circa, ma a volte arriva entro le 9:00". Tale vincolo sull'arrivo di John in ufficio può essere modellato in forma quantitativa come:

$$\text{origin of time} \left\{ \begin{array}{l} (7:20, 7:30, 8:00, 8:10), \\ (7:20, 7:30, 9:00, 9:00)[0,5] \end{array} \right\} \text{John arrives}$$

dove:

- le indicazioni temporali sono mantenute per facilitare la lettura (una volta scelto il valore dell'origine dei tempi e di quanto temporale, tali valori devono essere tradotti in numeri reali);
- il primo grado di preferenza è omissso perché posto ad 1 (come accadeva anche per le relazioni qualitative).

Da un punto di vista grafico, la relazione può essere visualizzata come in Figura 3.3 (dove la linea grigia indica il grado di $\mu_{R_{ij}}$ mentre le altre linee indicano i due trapezi che compongono il vincolo).

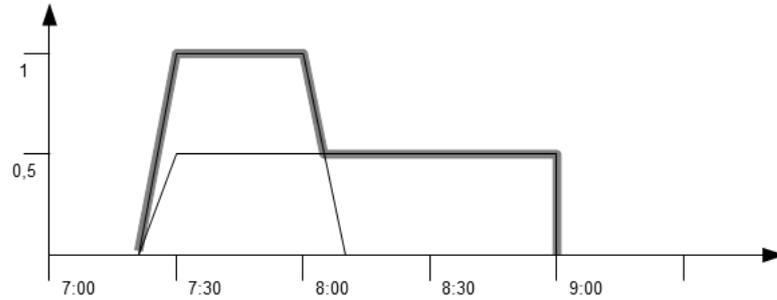


Figura 3.3: Rappresentazione grafica dell'esempio del vincolo quantitativo

3.2.4 Operazioni fra relazioni quantitative

Come per le relazioni qualitative, sono stati definiti quattro operatori sui vincoli e sono:

- inversione, operatore unario identificato dal simbolo $^{-1}$;
- intersezione, operatore binario identificato dal simbolo \otimes ;
- unione, operatore binario identificato dal simbolo \oplus ;
- composizione, operatore binario identificato dal simbolo \circ ;

che sono definiti di seguito.

Operazione di inversione

Si considerino due oggetti temporali i e j sui quali è definito il vincolo

$$i R_{ij} j = i \{T_1, T_2, \dots, T_n\} j$$

allora il vincolo inverso R_{ji} viene definito come

$$j R_{ji} i = j R_{ij}^{-1} i = j \{T_1^{-1}, T_2^{-1}, \dots, T_n^{-1}\} i$$

dove $T_k^{-1} = \lll'_k -d_k, -c_k, -b_k, -a_k \ggg'_k [\alpha_k]$ e dove \lll'_k e \ggg'_k sono dello stesso tipo (parentesi tonde o quadre) rispettivamente di \ggg_k e \lll_k .

Per esempio, dato il vincolo quantitativo

$$R_1 = \left\{ \begin{array}{l} (5, 6, 8, 10), \\ [4, 4, 12, 14][0,5] \end{array} \right\}$$

allora $R_2 = R_1^{-1}$ risulta pari a

$$R_2 = \left\{ \begin{array}{l} (-10, -8, -6, -5), \\ (-14, -12, -4, -4)[0,5] \end{array} \right\}$$

come rappresentato in figura 3.4.

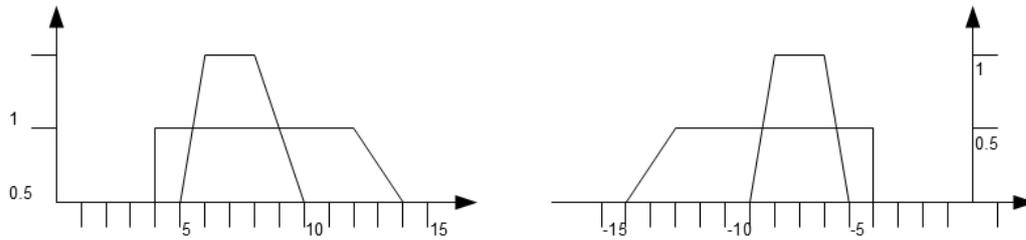


Figura 3.4: Le relazioni R_1 ed $R_2 = R_1^{-1}$

Operazione di unione

Si considerino due oggetti temporali i e j sui quali sono definiti i vincoli

$$i R_{ij}^1 j = i \{T_1^1, T_2^1, \dots, T_{n_1}^1\} j$$

$$i R_{ij}^2 j = i \{T_1^2, T_2^2, \dots, T_{n_2}^2\} j$$

allora il vincolo di unione $R_{ij} = R_{ij}^1 \oplus R_{ij}^2$ risulta

$$i R_{ij} j = i R_{ij}^1 \oplus R_{ij}^2 j = i \{T_1^1, T_2^1, \dots, T_{n_1}^1, T_1^2, T_2^2, \dots, T_{n_2}^2\} j$$

ossia le due liste di trapezi vengono fuse in un'unica relazione R_{ij} .

Si considerino, per esempio, i vincoli quantitativi R_1 ed R_2

$$R_1 = \left\{ \begin{array}{l} (5, 6, 9, 10), \\ [4, 4, 12, 14][0,5] \end{array} \right\}$$

$$R_2 = \{ [3, 3, 8, 10) \}$$

allora il vincolo che si ottiene unendo R_1 ed R_2 risulta pari a

$$R_3 = R_1 \oplus R_2 = \left\{ \begin{array}{l} (5, 6, 9, 10), \\ [4, 4, 12, 14)[0,5] \\ [3, 3, 8, 10) \end{array} \right\} = \left\{ \begin{array}{l} [3, 3, 9, 10), \\ [4, 4, 12, 14)[0,5] \end{array} \right\}$$

dove i vincoli $(5, 6, 9, 10)$ e $[3, 3, 8, 10)$ sono stati uniti in $[4, 4, 9, 10)$, come rappresentato in figura 3.5 perché tale sostituzione non altera la capacità informativa del vincolo.

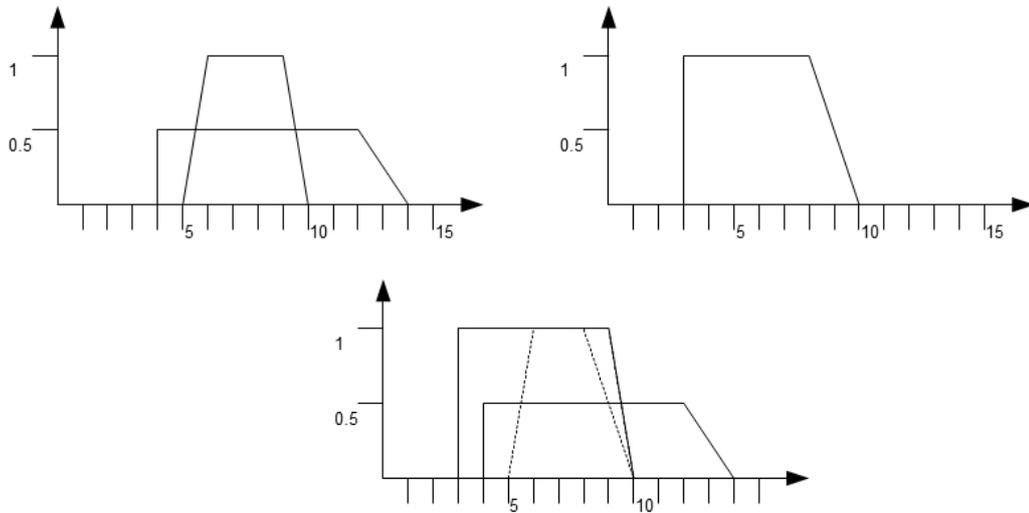


Figura 3.5: Le relazioni R_1, R_2 ed $R_3 = R_1 \oplus R_2$

Operazione di composizione

Si considerino tre oggetti temporali i, j e k sui quali sono definiti i vincoli

$$i R_{ij} j = i \{T_1^{ij}, T_2^{ij}, \dots, T_{n_1}^{ij}\} j$$

$$j R_{jk} k = j \{T_1^{jk}, T_2^{jk}, \dots, T_{n_2}^{jk}\} k$$

allora il vincolo di composizione $R_{ik} = R_{ij} \circ R_{jk}$ risulta

$$i R_{ik} k = i R_{ij} \circ R_{jk} k = i \{T_1, T_2, \dots, T_{n_1 \times n_2}\} k$$

dove il generico trapezio T_z^{ik} , con $z = 1, \dots, n_1 \times n_2$ è ottenuto tramite composizione dei trapezi T_x^{ij} e T_y^{jk} , con $x = 1, \dots, n_1$ ed $y = 1, \dots, n_2$.

La composizione di due trapezi $T_z^{ik} = T_x^{ij} \circ T_y^{jk}$ si ottiene tramite i seguenti passaggi:

1. se $\alpha_x \neq \alpha_y$, e si supponga senza perdita di generalità che $\alpha_x < \alpha_y$, allora è necessario considerare la versione troncata di T_y al valore $\alpha_y = \alpha_x$ ossia

$$TR_{\alpha_x}(T_y^{jk}) = \llcorner_y a_y, a_y + \frac{\alpha_x}{\alpha_y}(b_y - b_x), d_y - \frac{\alpha_x}{\alpha_y}(d_y - c_x), d_y \ggcorner_y$$

2. quando entrambi i trapezi presentano medesimo grado massimo allora il trapezio T_z^{ik} che si ottiene per composizione di T_x^{ij} con T_y^{jk} risulta

$$T_k^{ik} = T_x^{ij} \circ T_y^{jk} = \llcorner_k (a_x^{ij} + a_y^{jk}), (b_x^{ij} + b_y^{jk}), (c_x^{ij} + c_y^{jk}), (d_x^{ij} + d_y^{jk}) \ggcorner_k [\alpha]$$

dove $\alpha = \min\{\alpha_x, \alpha_y\}$, \llcorner_k è di tipo (se \llcorner_x^{ij} o \llcorner_y^{jk} sono di tipo (e dove \ggcorner_k è di tipo) se \ggcorner_x^{ij} o \ggcorner_y^{jk} sono di tipo).

Si considerino, per esempio, i vincoli quantitativi R_1 ed R_2

$$R_1 = \left\{ \begin{array}{l} (5, 6, 9, 10), \\ [4, 4, 8, 11][0,5] \end{array} \right\}$$

$$R_2 = \{ (0, 2, 3, 3) \}$$

allora il vincolo che si ottiene componendo R_1 con R_2 risulta pari a

$$R_3 = R_1 \circ R_2 = \left\{ \begin{array}{l} (5, 6, 9, 10), \circ (0, 2, 3, 3] \\ [4, 4, 8, 11][0,5] \circ (0, 1, 3, 3][0,5] \end{array} \right\} = \left\{ \begin{array}{l} (5, 8, 12, 13), \\ (4, 5, 11, 14)[0,5] \end{array} \right\}$$

come rappresentato in figura 3.6.

Operazione di intersezione

La definizione dell'operatore di intersezione risulta più difficile da formalizzare perché l'intersezione dell'area di due trapezi a priori non è un trapezio (come si può osservare nella figura 3.7). È necessario, quindi, prestare la massima attenzione nella sua definizione per produrre soluzioni che siano ancora dei vincoli quantitativi.

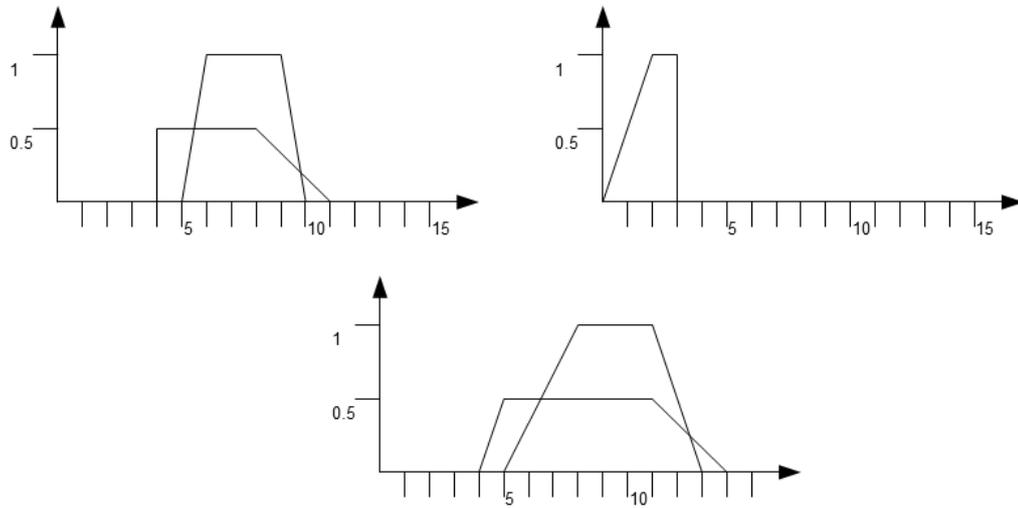


Figura 3.6: Le relazioni R_1, R_2 ed $R_3 = R_1 \circ R_2$

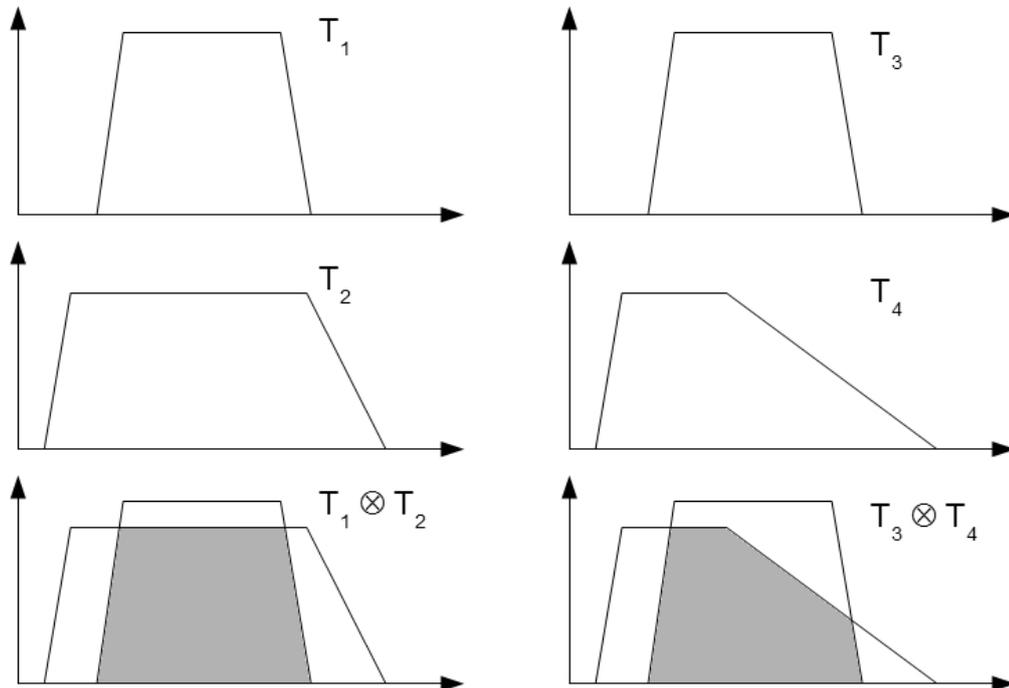


Figura 3.7: Sulle due colonne sono riportati due esempi in cui l'intersezione di due trapezi può essere o meno un trapezio

Per ovviare a tale problema in [Badaloni et al, 2003] sono proposti tre possibili approssimazioni che si ottengono:

- mantenendo il primo trapezio, ossia $T_k = T_x \otimes T_y \cong T_x$

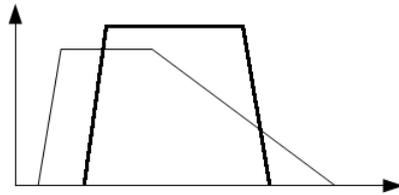


Figura 3.8: L'operazione $T_k = T_x \otimes T_y \cong T_x$

- approssimando T_k con il più piccolo trapezio contenuto in T_x che contenga l'intersezione di $T_x \otimes T_y$, ed indicato come $T_k = T_x \triangleleft T_y$

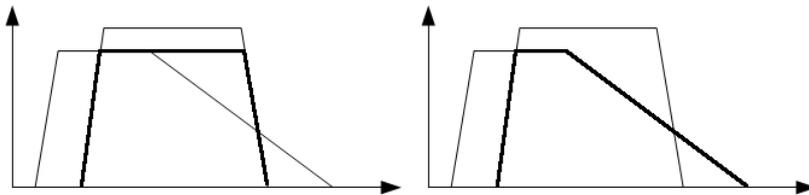


Figura 3.9: Le operazioni $T_k = T_x \triangleleft T_y$ e $T_k = T_y \triangleleft T_x$

- approssimando T_k con il più grande trapezio contenuto nell'intersezione e che mantenga supporto e grado massimo, indicato come $T_k = T_x \otimes_a T_y$

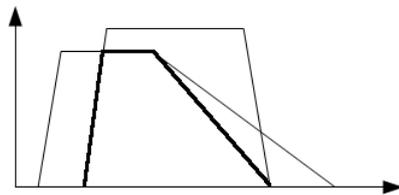


Figura 3.10: L'operazione $T_k = T_x \otimes_a T_y$

Delle tre alternative è stata scelta l'ultima per definire l'operazione di intersezione, perché:

- mantiene il supporto ed il grado simili a quelli reali;
- non aumenta il grado delle soluzioni ma, al più le sottostima;
- è l'unica definizione che gode della proprietà commutativa.

Per il calcolo di $T_z = T_x \otimes T_y$ sono, quindi, necessari i seguenti passaggi:

1. se $\alpha_x \neq \alpha_y$, allora si deve considerare la versione troncata del trapezio a grado maggiore, come già indicato per l'operazione di composizione;
2. $a_z = \max\{a_x, a_y\}$;
3. $b_z = \max\{b_x, b_y\}$;
4. $c_z = \min\{c_x, c_y\}$;
5. $d_z = \min\{d_x, d_y\}$;
6. se $a_x > d_z$ allora l'intersezione è nulla;
7. se $b_z > c_z$ allora i due valori devono essere ricalcolati al punto di intersezione fra i lati dei trapezi T_x e T_y con grado minore;
8. \ll_z e \gg_z devono essere poste in modo coerente con l'intersezione ottenuta.

Per esemplificare quanto detto si faccia riferimento alla figura 3.7.

Siano $T_1 = T_3 = (3, 4, 8, 9)$, $T_2 = (1, 2, 10, 12)[0.9]$ e $T_4 = (1, 2, 5, 13)[0.9]$ allora si ottengono:

$$T_1 \otimes T_2 = (3, 3.9, 8.1, 9) \text{ e}$$

$$T_3 \otimes T_4 = (3, 3.9, 5, 9).$$

3.3 Integrazione di vincoli qualitativi e quantitativi

Nelle sezioni precedenti sono stati considerati in modo separato vincoli qualitativi e quantitativi, dove i secondi possono essere utilizzati solo su coppie punto - punto. E' possibile integrare tutte queste informazioni in un unico sistema di ragionamento temporale, come proposto in [Badaloni et al, 2003].

Per raggiungere tale scopo è necessario definire gli operatori $QUAL^{fuz}(R_{ij})$ e $QUAN^{fuz}(R_{ij})$ per trasformare rispettivamente i vincoli quantitativi in qualitativi e viceversa; inoltre deve essere definito il comportamento degli operatori di intersezione, unione e composizione nei casi in cui si debba operare su vincoli non omogenei.

3.3.1 Conversione di vincoli quantitativi in qualitativi

Sia $R = \{T_1, T_2, \dots, T_n\}$ un vincolo quantitativo definito fra due punti temporali, allora il vincolo qualitativo a lui associato si ricava tramite l'operatore $QUAL^{fuz}(R)$.

Per ogni trapezio T_k viene valutata la funzione di appartenenza μ_{T_k} sul semiasse positivo, sull'origine e sul semiasse negativo determinando i valori $h^+(T_k)$, $h^0(T_k)$ ed $h^-(T_k)$ come segue.

$$h^+(T_k) = \sup_{x>0} \mu_{T_k}(x) = \begin{cases} 0 & \text{se } d_k \leq 0; \\ \frac{d_k \alpha_k}{d_k - c_k} & \text{se } c_k < 0 < d_k \\ \alpha_k & \text{altrimenti} \end{cases}$$

$$h^0(T_k) = \mu_{T_k}(0) = \begin{cases} 0 & \text{se } (a_k > 0) \vee (a_k = 0 \wedge \ll_k = ()) \\ & \vee (d_k < 0) \vee (d_k = 0 \wedge \gg_k = ()); \\ \frac{a_k \alpha_k}{a_k - b_k} & \text{se } a_k < 0 < b_k \\ \frac{d_k \alpha_k}{d_k - c_k} & \text{se } c_k < 0 < d_k \\ \alpha_k & \text{altrimenti} \end{cases}$$

$$h^-(T_k) = \sup_{x<0} \mu_{T_k}(x) = \begin{cases} 0 & \text{se } a_k \geq 0; \\ \frac{a_k \alpha_k}{a_k - b_k} & \text{se } a_k < 0 < b_k \\ \alpha_k & \text{altrimenti} \end{cases}$$

L'operatore $QUAL^{fuz}(R)$ può essere, ora, definito nel seguente modo:

$$QUAL^{fuz}(R) = \left(b[\max_{k=1, \dots, n} \{h^+(T_k)\}], eq[\max_{k=1, \dots, n} \{h^0(T_k)\}], a[\max_{k=1, \dots, n} \{h^-(T_k)\}] \right).$$

Tale comportamento può essere facilmente spiegato se si riflette sul fatto che il vincolo $i \{[a, a, d, d]\} j$ comporta che la differenza $j - i \in [a, d]$ ossia che $a < j - i < d$. Ponendo $a > 0$ allora $j - i > 0$ e da questo si ricava che $j > i$ ossia che vale la relazione $i \{b\} j$. In modo analogo se $d < 0$ allora si ricava $j - i < 0$ e quindi vale la relazione $i \{a\} j$.

3.3.2 Conversione di vincoli qualitativi in quantitativi

Sia $R = \{b[\alpha_1], eq[\alpha_2], a[\alpha_3]\}$ una relazione qualitativa, allora la sua versione quantitativa, che si ottiene tramite la funzione $QUAN^{fuz}$ presenterà la seguente forma:

- se $\alpha_1 > 0$ allora $(0, 0, +\infty, +\infty)[\alpha_1] \in QUAN^{fuz}(R)$;
- se $\alpha_2 > 0$ allora $[0, 0, 0, 0][\alpha_2] \in QUAN^{fuz}(R)$;
- se $\alpha_3 > 0$ allora $(-\infty, -\infty, 0, 0)[\alpha_3] \in QUAN^{fuz}(R)$;

3.3.3 Operatori fra vincoli qualitativi e quantitativi

Una volta definiti gli operatori che permettono la conversione fra vincoli qualitativi e quantitativi è possibile definire gli operatori di intersezione, unione e composizione nei casi in cui uno dei due vincoli coinvolti sia qualitativo e l'altro quantitativo.

Gli unici casi che devono essere gestiti sono i seguenti:

- sia R_1 , vincolo punto - punto espresso in forma qualitativa, ed R_2 , vincolo quantitativo e sia applicato l'operatore di intersezione, allora risulta:

$$R_1 \otimes R_2 = R_2 \otimes R_1 = QUAN^{fuz}(R_1) \otimes R_2;$$

- sia R_1 , vincolo punto - punto espresso in forma qualitativa, ed R_2 , vincolo quantitativo e sia applicato l'operatore di unione, allora risulta:

$$R_1 \oplus R_2 = R_2 \oplus R_1 = QUAN^{fuz}(R_1) \oplus R_2;$$

- sia R_1 , vincolo punto - punto espresso in forma qualitativa, ed R_2 , vincolo quantitativo e sia applicato l'operatore di composizione, allora si ottiene:

$$R_1 \circ R_2 = QUAN^{fuz}(R_1) \circ R_2;$$

secondo la medesima metodologia si ottiene che:

$$R_2 \circ R_1 = R_2 \circ QUAN^{fuz}(R_1);$$

- sia R_1 , vincolo intervallo - punto, ed R_2 , vincolo quantitativo e sia applicato l'operatore di composizione, allora risulta:

$$R_1 \circ R_2 = R_1 \circ QUAL^{fuz}(R_2);$$

- sia R_1 , vincolo punto - intervallo, ed R_2 , vincolo quantitativo e sia applicato l'operatore di composizione, allora risulta:

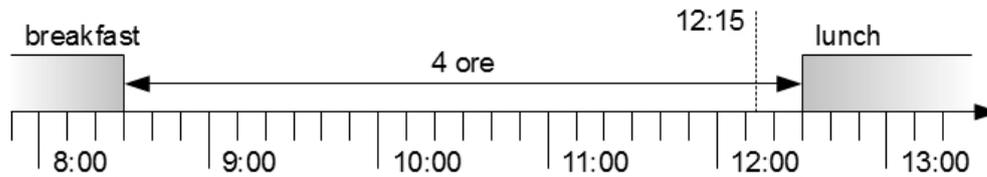
$$R_2 \circ R_1 = QUAL^{fuz}(R_2) \circ R_1.$$

Come si può notare, l'idea alla base di tali scelte è quella mantenere i vincoli in forma quantitativa fintanto che si opera su relazioni che coinvolgono punti temporali, perché tale forma permette di mantenere una potenza espressiva maggiore; si passa alla forma qualitativa solo quando le relazioni risultanti coinvolgono almeno un intervallo.

3.4 Algoritmo di Path Consistency

Data una rete temporale N sulla quale sono stati definiti “pochi” vincoli fra i nodi, allora possiamo essere interessati all’effetto globale che tali relazioni hanno sulla rete.

Per esempio, se affermiamo che la colazione finisce dopo le 8:30 e che il pranzo inizia dopo 4 ore dalla colazione allora deve essere possibile ricavare in modo automatico che le 12:30 sono il primo istante utile in cui cominciare il pasto rispettando tutti i vincoli imposti. Inserendo, oltre ai vincoli sopra riportati, anche quello per cui il pranzo deve iniziare prima delle 12:15 allora deve risultare palese che non esiste una soluzione che rispetti tutti e tre i vincoli.



In modo più formale ciò che si desidera determinare è il grado di consistenza della rete N (ossia il minimo fra i massimi gradi dei vincoli della rete) e calcolare la così detta “rete minimale”. Quest’ultima è definita come una rete equivalente ad N in cui i vincoli sono i più espliciti possibili, il che significa che ogni assegnazione di due variabili, che soddisfa i corrispondenti vincoli binari con un certo grado, può essere estesa ad una soluzione globale che rispetta tale grado di soddisfacimento.

Siccome un tradizionale problema di TCSP (Temporal Constraint Satisfaction Problem, ossia problema di soddisfacimento vincoli temporali organizzato nella forma di CSP) può essere modellato tramite una rete temporale fuzzy i cui gradi di consistenza sono 0 o 1, allora l’intrattabilità del problema nella sua versione crisp comporta che tutti gli obiettivi sopra riportati siano anch’essi problemi NP-Difficili.

Una soluzione approssimata al problema di determinare la rete minimale si può ricavare tramite l’algoritmo di path-consistency. La sua formulazione iniziale è stata proposta in [Allen, 1983] e la sua struttura è stata mantenuta pressoché invariata nei lavori di [Vilain et al, 1989], [Meiri, 1995], [Badaloni et al, 2003] salvo per la definizione delle operazioni fra vincoli temporali.

L’algoritmo di Path-Consistency per reti temporali fuzzy può essere così definito:

```

PCfuz(N)
  Q ← { (i, j) : 1 ≤ i < j ≤ n }
  while (Q ≠ ∅)
    select and delete (i, j) from Q
    for k ← 1 to n: k ≠ i and k ≠ j do
      t ← Rik ⊗ (Rij ∘ Rjk)
      if (t ≠ Rik) then
        Rik ← t
        Rki ← t-1
        Q ← Q ∪ { (i, k) }
        t ← Rkj ⊗ (Rki ∘ Rij)
      if (t ≠ Rkj) then
        Rkj ← t
        Rjk ← t-1
        Q ← Q ∪ { (k, j) }
  return min(i,j):1≤i<j≤n max{deg(Rij)}

```

Come dimostrato in modo sperimentale in [Allen, 1983], l'algoritmo di path-consistency non individua tutti i vincoli perché si appoggia, durante la sua computazione, solo su terne di oggetti temporali. Inoltre, come ipotizzato da [Allen, 1983] e dimostrato in [Vilain et al, 1989] tale algoritmo presenta complessità polinomiale (in particolare è stato calcolato che la sua complessità è $O(n^3)$ nel numero di operazioni di intersezione e composizione).

Per questa ragione lo studio di [Vilain et al, 1989] si focalizza non tanto sull'algebra degli intervalli (sotto-algebra della QA^{fuz} che si appoggia solo su relazioni II nella loro versione crisp) introdotta in [Allen, 1983], ma sull'algebra dei punti (sotto-algebra della QA^{fuz} che si appoggia solo su relazioni PP nella loro versione crisp) ed in particolare in una sua sotto-algebra denominata Continuous Point Algebra(PA_C).

La PA_C è costituita da tutte le relazioni PP tali per cui presi due punti x e y che verificano il vincolo, allora anche un qualsiasi altro punto in (x, y) verifica il medesimo vincolo. Per tale definizione quasi tutte le relazioni possibili nella PA fanno parte della PA_C , con la sola eccezione di $\{b, a\}$. Come indicato in [Vilain et al, 1989] l'algoritmo di path-consistency permette di calcolare la rete minimale se si limitano le operazioni alla sola PA_C .

Lo studio proposto in [?], dopo aver introdotto l'algebra IA^{fuz} , estensione fuzzy dell'algebra degli intervalli IA ha posto l'attenzione sulla trattabilità del problema di individuazione della rete minimale. Per farlo sono state definite la point algebra fuzzy (PA^{fuz}), (estensione fuzzy della PA) e la continuous point algebra fuzzy (PA_C^{fuz}) (sotto algebra di PA^{fuz} costituita da tutte le re-

lazioni del tipo $\{b[\alpha_1], eq[\alpha_2], a[\alpha_3]\}$ per le quali vale che $\alpha_2 \geq \min\{\alpha_1, \alpha_2\}$. Sono state definite, inoltre, le sotto-algebre SA^{fuz} e SA_C^{fuz} che sono costituite da tutti i vincoli IA^{fuz} per i quali è possibile una traduzione esatta, rispettivamente, in termini di PA^{fuz} e PA_C^{fuz} utilizzando i punti di inizio e fine intervallo.

Sempre in [?] si dimostra che il problema di individuare una soluzione minimale per una rete con vincoli PA_C^{fuz} si risolve tramite l'algoritmo di path-consistency in tempo polinomiale e si dimostra, inoltre, che tale proprietà vale anche per reti con vincoli SA_C^{fuz} .

3.5 Problema della Frammentazione

Con il termine "Frammentazione" ci si riferisce a quel fenomeno per cui un vincolo quantitativo presenta al suo interno numerosi trapezi, che non possono essere sostituiti da un insieme più piccolo mantenendo inalterata la potenza informativa (ossia la funzione di appartenenza del vincolo).

Questo fenomeno risulta particolarmente importante se abbinato all'algoritmo di path-consistency, dove tali informazioni vengono propagate su tutta la rete, rischiando così di aumentare la dimensione di altri vincoli quantitativi e rischiando di appesantire la computazione.

Fra le cause palesi al problema della frammentazione vi sono la formulazione originale dei vincoli (già frammentati) e l'operazione di intersezione fra vincoli qualitativi e quantitativi. Vediamo entrambi gli scenari tramite degli esempi.

Il primo caso riguarda i vincoli originali; si prenda, ad esempio, una rete di tre punti (A, B, C) sui quali sono definite le relazioni:

- $A\{[1, 1, 2, 2], [11, 11, 12, 12], [21, 21, 22, 22]\}B$
- $A\{[0, 0, 22, 22], [23, 23, 33, 33], [34, 34, 50, 50]\}C$
- $A\{(0, 1, 2, 3), [16, 16, 17, 17], [23, 23, 24, 24]\}C$

allora, dopo aver applicato l'algoritmo di path-consistency il risultato è il seguente:

- $A\{[1, 1, 2, 2], [11, 11, 12, 12], [21, 21, 22, 22]\}B$
- $A \left\{ \begin{array}{l} (1, 2, 4, 5), (11, 12, 14, 15), [17, 17, 19, 19], \\ (21, 22, 22, 22), [23, 23, 26, 26], [27, 27, 29, 29], \\ [34, 34, 36, 36], [37, 37, 39, 39], [44, 44, 46, 46] \end{array} \right\} C$
- $A\{(0, 1, 2, 3), [16, 16, 17, 17], [23, 23, 24, 24]\}C$

dove la frammentazione sul vincolo (A, C) è palese se consideriamo le operazioni $R_{AC} \otimes (R_{AB} \circ R_{BC})$ che vengono compiute durante l'algoritmo di path-consistency.

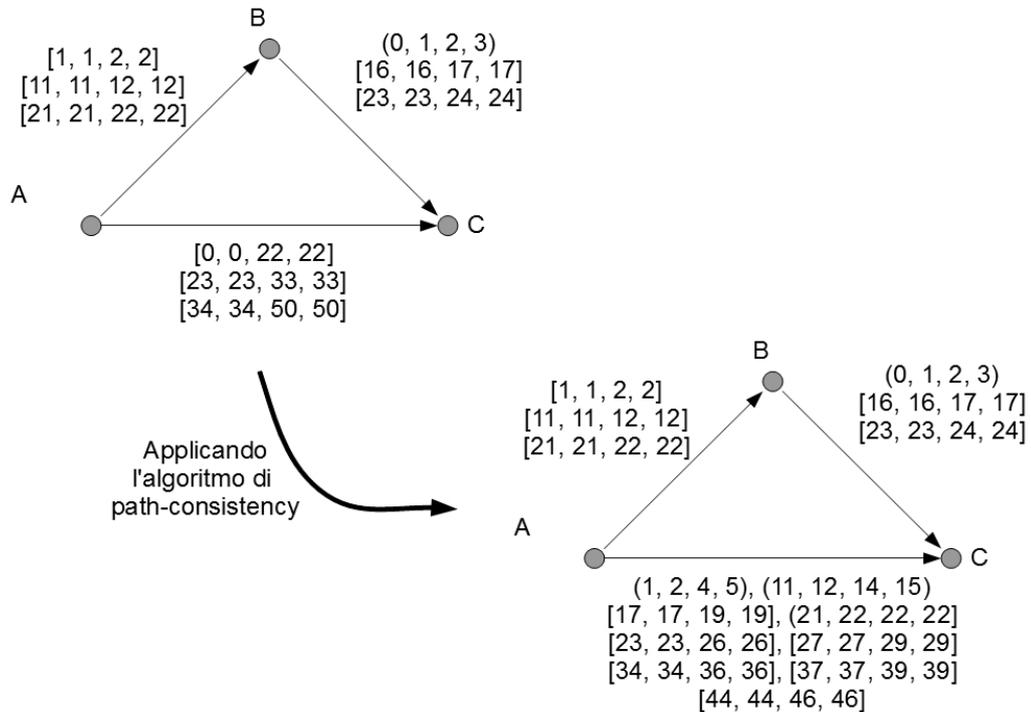


Figura 3.11: Esempio di frammentazione dovuta alla formulazione del problema prima e dopo l'invocazione dell'algoritmo di path-consistency

Il secondo caso riguarda l'operazione di intersezione; si prenda ad esempio la rete generata dai seguenti vincoli (e riportata in figura 3.12):

- la cena non dovrebbe iniziare prima delle 19:30 e dura circa 1 ora;
- l'aspirina non dovrebbe essere presa prima della fine della cena, e preferibilmente fra i 5 ed i 20 minuti dalla sua conclusione. L'attività dura solitamente 5 minuti.
- si ipotizzi l'origine dei tempi alle 18:00 ed un quanto temporale di 1 minuto.

Fra le iterazione dell'algoritmo di path-consistency vengono compiuti, fra gli altri, i tre passaggi riportati di seguito:

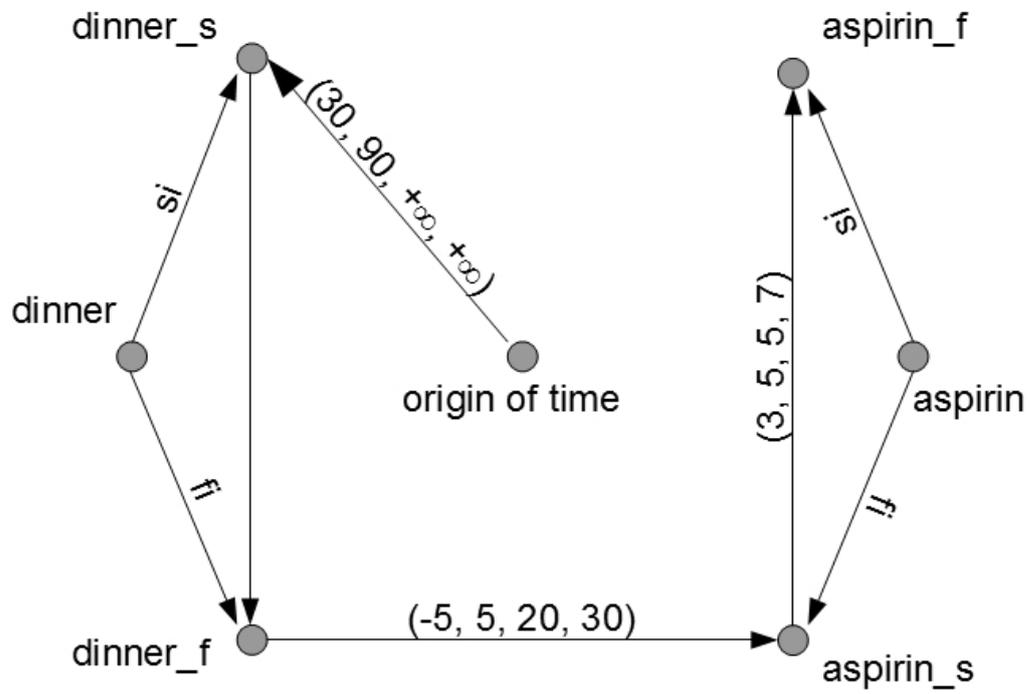


Figura 3.12: Rappresentazione della rete che genera frammentazione a causa dell'operazione di intersezione

- propagando i vincoli su $(dinner_f, aspirin_s)$ e $(aspirin_s, aspirin_f)$ si ottiene il vincolo $dinner_f\{(-2, 10, 25, 37)\}aspirin_f$;
- propagando i vincoli su $(dinner, dinner_f)$ e $(dinner_f, aspirin_f)$ si ottiene il vincolo $dinner\{ai, di[0, 166], fi[0, 166]\}aspirin_f$;
- propagando i vincoli su $(dinner_f, dinner)$ e $(dinner, aspirin_f)$ si ottiene il vincolo $dinner_f\{b, eq, a[0, 166]\}aspirin_f$ che deve essere confrontato con $dinner_f\{(-2, 10, 25, 37)\}aspirin_f$.

L'operazione di intersezione che deve essere compiuta nell'ultimo passaggio genera, secondo la teoria riportata finora, un vincolo composto da due trapezi nella forma $dinner_f\{(-2, 0, 0, 0)[0, 166], (0, 10, 25, 37)\}aspirin_f$ e questo comporta la frammentazione del vincolo fra $dinner_f$ ed $aspirin_f$. Un'analisi più attenta della rappresentazione grafica (riportata in figura 3.13) dei due vincoli, fa, però, emergere che la reale intersezione fra le aree sottese dal vincolo qualitativo e quello quantitativo coincide con il vincolo quantitativo stesso.

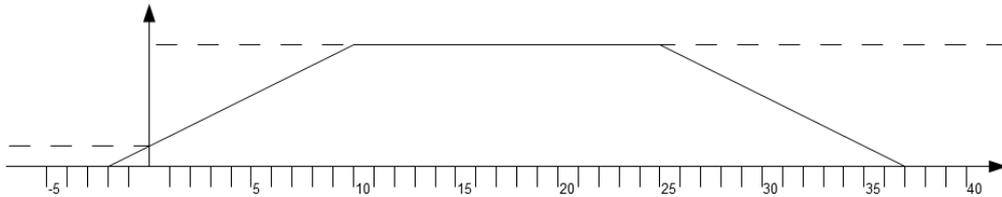


Figura 3.13: Rappresentazione del vincolo che genera frammentazione

In letteratura sono state proposte almeno due soluzioni per ovviare al problema della frammentazione e sono le approssimazioni ULT e LPC.

L'approssimazione ULT (Upper Lower Tightening) si prefigge lo scopo di semplificare le operazioni fra vincoli quantitativi non considerando il vincolo originale, ma una sua versione “semplificata” costituita dal più piccolo trapezio che contiene tutti i frammenti originali. Tale metodo non genera soluzioni accettabili a priori quando viene applicato l'algoritmo di path-consistency, ma, se questo restituisce il grado 0, allora anche la rete originale non è consistente.

Una seconda approssimazione è quella LPC (Loose Path Consistency) che riduce il problema della frammentazione, limitando il numero massimo di trapezi che si generano in fase di intersezione fra vincoli. In questo caso, dati due vincoli quantitativi R_1 ed R_2 , si sostituisce l'operazione $R_1 \otimes R_2$ con

$R_1 \triangleleft R_2$ definita come

$$R_1 \triangleleft R_2 = \{T'_1, T'_2, \dots, T'_n : T'_k = [L_k, L_k, U_k, U_k]\}$$

dove L_k, U_k sono rispettivamente l'estremo inferiore e superiore dell'intersezione fra $T_k \in R_1$ ed R_2 . Anche in questo caso se la rete è inconsistente, allora anche la rete originaria è inconsistente.

In questo lavoro viene proposta un'alternativa, ispirata dallo studio di fenomeni come quello indicato nel secondo esempio (vedi figura 3.12). L'idea è quella di modificare la definizione dell'operazione di intersezione nel caso in cui uno dei vincoli sia qualitativo e l'altro quantitativo.

Come visto in figura 3.13, e comunque facilmente intuibile, l'area sottesa dalla funzione di appartenenza del vincolo $QUAN^{fuz}(QUAL^{fuz}(R_{quan}))$, dove R_{quan} è un vincolo quantitativo, è un limite superiore all'area sottostante della funzione $\mu_{R_{quan}}$.

Per questa ragione di fronte ad un'operazione $R_{qual} \otimes R_{quan}$ fra un vincolo qualitativo ed uno quantitativo prima bisogna verificare se

$$R_{qual} \otimes QUAL^{fuz}(R_{quan}) \geq QUAL^{fuz}(R_{quan})$$

perché in questo caso risultato dell'operazione è il vincolo R_{quan} stesso, altrimenti si procede applicando l'operatore di intersezione come visto in precedenza.

In fase sperimentale si è riscontrato come l'applicazione di questa variante permetta di ridurre i tempi di computazione dell'algoritmo di path-consistency, già in reti con una ventina di oggetti temporali (di cui solo un terzo sono intervalli) e la frammentazione dei vincoli qualitativi.

3.6 Nozione di Evento

È stato osservato come gli oggetti gestiti nel sistema di ragionamento temporale fuzzy siano punti ed intervalli e come fra di loro vi sia una forte corrispondenza. In particolare possiamo osservare come i fenomeni che vengono modellati nella rete temporale siano costituiti da un intervallo e dai suoi punti di inizio e di fine. Questo particolare raggruppamento di oggetti presenta dei vincoli fissi:

- fra intervallo e punto di inizio vi è un vincolo IP del tipo $\{si\}$;
- fra intervallo e punto di fine vi è un vincolo IP del tipo $\{fi\}$;

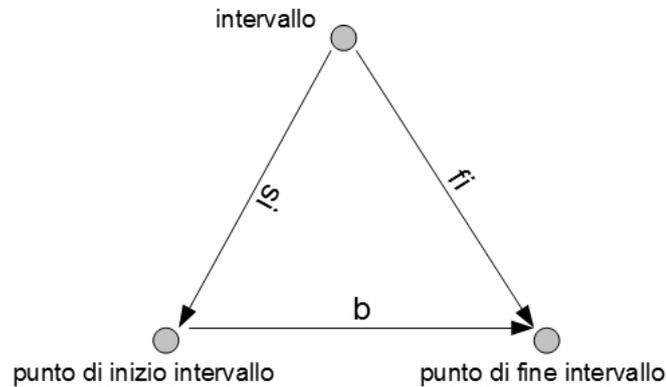


Figura 3.14: Esempio di Evento Completo

- fra punto di inizio e di fine si può ricavare una relazione di $\{b\}$;

come riportato in figura 3.14.

Per questa ragione in alcune applicazioni, come vedremo di seguito, non si è interessati al singolo oggetto, ma ad un raggruppamento di essi, legati insieme dal significato.

Per questa ragione si definisce come *Evento*, in una rete temporale fuzzy, un raggruppamento di, al più, tre oggetti temporali (un intervallo e due punti) legati fra loro perché sfumature diverse del medesimo fenomeno modellato.

In particolare si definisce come *Evento Completo* un evento che presenta al suo interno sia l'intervallo che i punti che fungono da estremi.

Per evitare ambiguità e per facilitare la ricerca degli eventi viene utilizzata la convenzione di etichettare gli oggetti di un evento nel seguente modo:

- l'intervallo viene etichettato tramite il nome del fenomeno che si vuole modellare, come "breakfast", "dinner", "aspirin intake";
- il punto di inizio del fenomeno viene etichettato componendo il nome dell'intervallo ed il suffisso "_s", a memoria della relazione $\{s\}$ che esiste fra il punto e l'intervallo;
- il punto di fine del fenomeno viene etichettato componendo il nome dell'intervallo ed il suffisso "_f", a memoria della relazione $\{f\}$ che esiste fra il punto e l'intervallo.

Un algoritmo "stupido" ma utile per raggruppare gli oggetti in eventi all'interno di una rete temporale fuzzy, è il seguente:

1. il punto etichettato “origin of time” se presente fa parte di un evento a sé stante;
2. ogni intervallo fa parte di un evento diverso dagli altri finora individuati;
3. se un punto verifica la convenzione (descritta sopra) rispetto ad un intervallo, allora appartiene allo stesso evento dell'intervallo individuato;
4. ogni punto non legato ad un intervallo si considera appartenente ad un evento diverso da quelli finora considerati.

Tali convenzioni sono state utilizzate in fase di implementazione per alcune applicazioni.

Capitolo 4

Implementazione

4.1 Sistema di ragionamento temporale fuzzy

4.1.1 Breve panoramica sul software prodotto

Il sistema di ragionamento temporale fuzzy descritto nel capitolo precedente è stato implementato tramite il linguaggio di programmazione C++, secondo lo schema UML riportato, seppure in forma semplificata, in figura 4.1.

Elemento centrale del software è la classe “Network” che ha lo scopo di generare, modificare e salvare reti temporali fuzzy. Fra le principali variabili d’istanza della classe vi sono la lista dei vertici che costituiscono la rete, i vincoli definiti su di essi, nonché le informazioni su “origin of time” e quanto temporale. Fra i metodi disponibili si possono ricordare quelli per:

- generare una nuova rete (a partire dall’insieme dei suoi vertici);
- caricare una rete precedentemente generata;
- salvare le elaborazioni su file;
- esplorare la rete ed i vincoli che la compongono;
- modificare le relazioni esistenti inserendone di nuove;
- lanciare l’algoritmo di path-consistency.

Rispetto a quest’ultimo punto è doveroso esplicitare che tale algoritmo opera secondo lo pseudo-codice descritto in precedenza e restituisce il grado massimo di consistenza della rete. Nel caso sia rilevata una situazione di inconsistenza, il metodo blocca la sua elaborazione al primo vincolo inconsistente generato e restituisce un grado nullo; non vengono, però, modificati

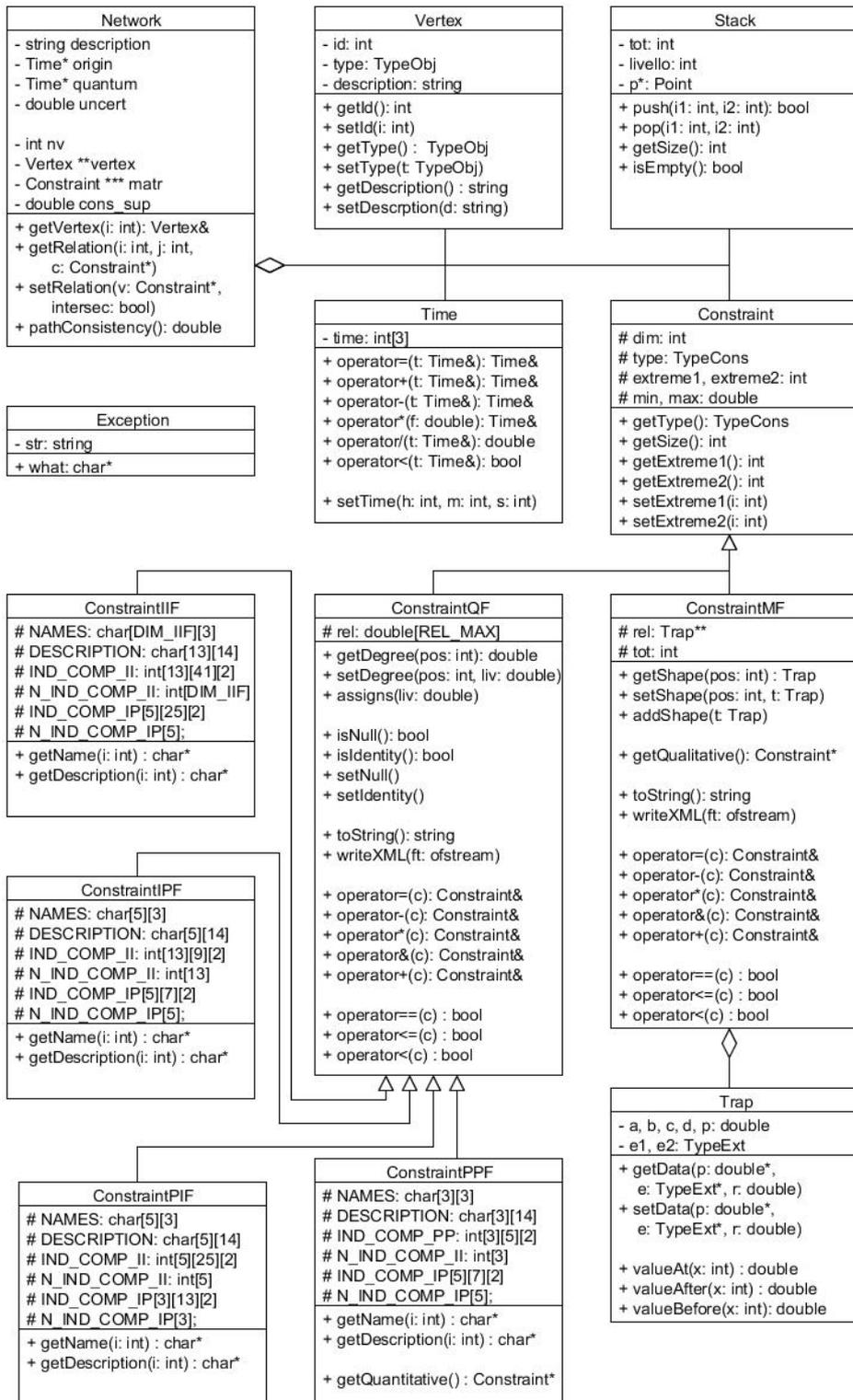


Figura 4.1: Schema UML semplificato

gli altri vincoli finora calcolati essendo l'inconsistenza resa palese dal valore restituito.

Le classi su cui si appoggia Network per le sue elaborazioni sono tre e servono a rappresentare le informazioni inerenti il tempo, i vertici della rete temporale e i vincoli definiti su di loro; oltre a queste si utilizza la libreria standard per la gestione di stringhe, i file di input / output e le operazioni matematiche.

In particolare, gli oggetti della classe "Time" rappresentano istanti temporali, espressi come ore, minuti e secondi ($hh : mm : ss$); sono stati, inoltre, implementati i metodi per manipolare tali informazioni e permettere il passaggio dalle indicazioni temporali ai numeri reali (utilizzati nei vincoli quantitativi). A questo scopo sono stati ridefiniti gli operatori di addizione e sottrazione, di divisione e di moltiplicazione per costante. In questo modo è possibile passare da un istante temporale t al suo corrispondente valore reale n tramite la formula $n = (t - origin_time)/quanto$ ed in modo opposto $t = n * quanto + origin_time$.

Come visto in precedenza, i vertici della rete fuzzy sono formati da punti ed intervalli. Gli oggetti della classe "Vertex" rappresentano tali vertici e presentano, al loro interno, tre variabili d'istanza: un descrittore del tipo di oggetto temporale (punto o intervallo), un identificativo numerico che tiene traccia dell'Id assegnato all'oggetto all'interno della rete ed una descrizione dello stesso.

I vincoli sono, invece, elaborati tramite la classe "Constraint" e le sue derivate. Tale classe è di tipo astratto e contiene solamente l'implementazione dei metodi comuni a tutte le relazioni, come quelli relativi agli estremi del vincolo, al tipo di vincolo ed alla sua dimensione (intesa come numero di relazioni elementari o di trapezi presenti). I metodi astratti sono inseriti per descrivere le operazioni comuni fra le relazioni, come gli operatori definiti sui vincoli e le operazioni di confronto fra relazioni dello stesso tipo.

Da questa classe ne derivano due che rappresentano i vincoli quantitativi e qualitativi e sono, rispettivamente, "ConstraintMF" e "ConstraintQF". La prima implementa quanto riportato nel capitolo precedente, unitamente alla modifica per la gestione della frammentazione basata sulla verifica di

$$R_{qual} \otimes QUAL^{fuz}(R_{quan}) \geq QUAL^{fuz}(R_{quan}).$$

La seconda classe è stata a sua volta derivata per specializzarne i comportamenti a seconda del tipo di relazioni che rappresentano e per mantenere le informazioni relative ai vincoli elementari, ai loro simboli e alle tabelle per le operazioni di composizione. Per questa ragione sono state implementate le classi "ConstraintIIF", "ConstraintIPF", "ConstraintPIF" e "ConstraintPPF".

Le classi “ConstraintPPF” e “ConstraintMF” mettono, inoltre, a disposizione i metodi che implementano rispettivamente le funzioni $QUAN^{fuz}(R)$ e $QUAL^{fuz}(R)$.

Tutto il codice che è stato brevemente introdotto fino a questo momento (al fine di gestire i parametri ricevuti in input non conformi alle aspettative) utilizza istanze della classe “Exception” per generare, propagare e gestire le eccezioni.

Il software è integrato con un programma (il cui codice è riportato in `ftcn.cpp`) che riceve in input il nome di un file su cui sono salvate le informazioni relative ad una rete temporale fuzzy, ne applica l’algoritmo di path-consistency e riporta i risultati su shell di comando ed in un apposito file di output.

4.1.2 Standard per i file di input / output

Come già evidenziato in precedenza, uno degli aspetti fondamentali del sistema di ragionamento temporale è la possibilità di salvare le elaborazioni compiute e riprenderle in un secondo momento tramite l’utilizzo di file. Questo impone la scelta di uno standard secondo il quale memorizzare tali informazioni; già nelle prime implementazioni (ad opera del dott. Falda) questa è ricaduta sull’utilizzo di file XML (ma con estensione `.xfn` per rimandare al tipo di informazioni in esso contenute) e secondo una struttura dettata da uno XMLSchema reperibile nel file `ftcn.xsd`.

Tra le motivazioni di questa scelta vi sono la facile comprensione dei dati salvati, per un utente con conoscenze del sistema, nonché la possibilità di interagire facilmente con gli stessi. Inoltre i file XML presentano un numero limitato di regole per essere ben formati e, grazie alla scrittura di un XMLSchema, è possibile validare i dati stessi tramite opportuni validatori presenti in commercio.

La formulazione originaria prevedeva l’inserimento di informazioni relative ai vari vertici della rete temporale e ai vincoli definiti fra di loro; a tali dati sono stati aggiunti tre campi opzionali che permettono di memorizzare il valore dell’origine dei tempi, del quanto temporale e del parametro di incertezza. Lo scopo dei primi due è stato già esplicitato in precedenza, il terzo è un parametro che è legato al problema di tradurre le espressioni del linguaggio umano in vincoli temporali e verrà definito in seguito.

4.2 Interazione con l'utente

La tematica dell'interazione con l'utente comporta la ricerca di soluzioni che permettano ad un soggetto, solitamente non esperto, di utilizzare facilmente il sistema di ragionamento temporale fuzzy presumendo una sua scarsa conoscenza del software con cui deve interagire. Fra le strategie che vengono comunemente adottate a tale scopo vi sono l'utilizzo di strumenti che interpretano il linguaggio umano e la costruzione di interfacce grafiche.

Entrambi gli approcci verranno affrontati di seguito.

4.2.1 Interpretazione del linguaggio utente

Già nel 1983 nell'articolo [Allen, 1983], l'autore osserva come il linguaggio quotidiano contiene informazioni di natura temporale sugli avvenimenti e sui legami che possono intercorrere fra di loro. Come è stato evidenziato in precedenza, i dati temporali possono indicare specifici istanti o intervalli di tempo, elementi caratterizzanti i singoli eventi o relazioni che esprimono legami fra più fenomeni.

Richiamando gli esempi visti in precedenza:

- il pranzo viene servito dopo la colazione;
- l'infuso di erbe deve essere preso fra le 12:00 e le 20:00;
- l'aspirina deve essere presa a stomaco pieno e preferibilmente entro venti minuti dalla fine della cena;
- la cena è un'attività che dura circa 1 ora.

ed ipotizzando una rete temporale che contenga almeno gli eventi "origin of time", "breakfast", "lunch", "dinner", "herbal tea" ed "aspirin intake" allora le espressioni sopra indicate potrebbero essere tradotte, almeno in prima istanza, come:

- breakfast $\{b\}$ lunch;
- origin of time $\{[12:00, 12:00, 20:00, 20:00]\}$ herbal tea_s;
- dinner_f $\{(0 - \varepsilon_1, 0, 20min, 20min + \varepsilon_1)\}$ aspirin intake_s;
- dinner_s $\{(1h - \varepsilon_2, 1h, 1h, 1h + \varepsilon_2)\}$ dinner_f.

Questi esempi ci permettono di intuire come il linguaggio umano possa essere tradotto in vincoli qualitativi o quantitativi da utilizzare in una rete temporale e come questa strategia possa essere utile per semplificare il problema dell'inserimento di vincoli da parte di un utente non esperto.

Sebbene la creazione di un interprete per ogni possibile espressione del linguaggio umano sia difficile da codificare, è tuttavia possibile semplificare il problema se questo viene posto come traduzione di un'espressione da un linguaggio naturale ridotto ad un vincolo applicabile ad una rete temporale. Tale compito è possibile tramite l'utilizzo di due componenti:

1. un traduttore di espressioni da un linguaggio utente (semplificato) in vincoli temporali, applicando un mapping fra termini noti e le corrispondenti caratteristiche della relazione;
2. una componente che tenga conto di aspetti quali la granularità del tempo, le traduzioni linguistiche e la presenza di eventuali sinonimi che trasformi le frasi espresse da un utente medio nel linguaggio semplificato.

Tale scomposizione era stata introdotta in [Badaloni, Falda, 2010] ed è stata ripresa nell'attuale implementazione.

Linguaggio utente e sua traduzione

La potenza espressiva del linguaggio utente deve essere tale da permettere di tradurre le frasi in vincoli. Per ottenere questo risultato, il linguaggio deve contenere gli elementi per individuare gli eventi coinvolti e le relazioni che vi sono fra di loro.

Ipotizzando la presenza di soli eventi completi all'interno della rete, oltre all'origine dei tempi, allora:

- ogni intervallo può essere indicato tramite la sua etichetta o descrizione, come per esempio l'espressione "breakfast" indica l'intervallo che rappresenta l'azione di fare colazione;
- i punti di inizio o fine evento possono essere individuati ponendo rispettivamente l'espressione "begins" o "ends" dopo il nome dell'evento;
- la relazione fra inizio e fine evento può essere espressa in modo compatto tramite la parola "length" posta dopo il nome dell'evento.

Per descrivere le relazioni qualitative è possibile rifarsi ai vincoli elementari fra coppie di intervalli, coppie di punti e coppie fra un punto ed

un intervallo, proposti rispettivamente in [Allen, 1983], [Vilain et al, 1989] e [Meiri, 1995]. Analizzando i tre lavori è possibile individuare 13 espressioni linguistiche utilizzabili nel nostro linguaggio semplificato e riproposte nella tabella 4.1.

Espressione	Relazione II	Relazione IP	Relazione PI	Relazione PP
before	{b}	{ai}	{b}	{b}
after	{a}	{bi}	{a}	{a}
during	{d}		{d}	
contains	{di}	{di}		
overlaps	{o}			
overlapped by	{oi}			
meets	{m}			
met by	{mi}			
starts	{s}		{s}	
started by	{si}	{si}		
finishes	{f}		{f}	
finished by	{fi}	{fi}		
equals to	{eq}			{eq}

Tabella 4.1: Espressioni linguistiche e loro traduzione in relazioni elementari

Per quanto riguarda le relazioni quantitative, in [Badaloni, Falda, 2010] sono state proposte 10 espressioni, combinabili fra loro, per modellare tali vincoli. In figura 4.2 sono stati riportati tali modelli da un punto di vista grafico, unitamente alle frasi ad essi associate e alle parti di trapezi che essi modellano.

Mentre i costrutti “at”, “at about”, “on”, “on about” modellano totalmente il trapezio della relazione che si vuole esprimere, gli altri possono essere utilizzati da soli, per indicare dei trapezi aperti a più o meno infinito, o uniti dalla congiunzione “but” per essere abbinati e rappresentare una forma chiusa.

Avendo definito i costrutti fondamentali per esprimere una relazione tra oggetti temporali rimane solo da definire la forma delle espressioni nel linguaggio utente. Un'espressione si definisce valida quando espressa nella forma (espressa come pseudo-espressione regolare):

$$\langle \text{oggetto}_1 \rangle (is)? \langle \text{vincolo} \rangle (or(is)? \langle \text{vincolo} \rangle)* \\ (\text{compared to})? \langle \text{oggetto}_2 \rangle ?$$

dove con l'espressione $\langle \text{oggetto}_i \rangle$ si vuole indicare il nome di un evento mentre con $\langle \text{vincolo} \rangle$ ci si riferisce ad una relazione elementare modellata

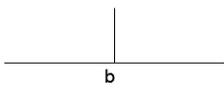
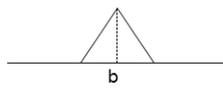
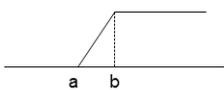
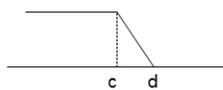
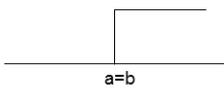
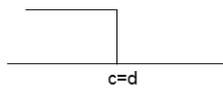
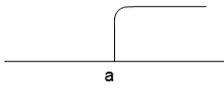
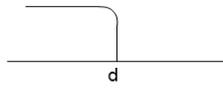
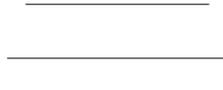
Espressione	Rappresentazione	Rappresentazione	Espressione
at b on b [b, b, b, b]			at about b on about b (b - ε, b, b, b + ε)
after about b not less than about b (a, b, -, -)			before about c not more than about c (-, -, c, d)
not before b not less than b [b, b, -, -)			not after c not more than c (-, -, c, c)
after a more than a			before d less than d
(-∞, -∞, -, -)			(-, -, +∞, +∞)

Figura 4.2: Espressioni di relazioni quantitative e loro traduzione

come sopra. L'oggetto $\langle \text{oggetto}_2 \rangle$, se non indicato, viene inteso come origine dei tempi, salvo nel caso in cui il primo oggetto temporale utilizzi il costrutto "length", nel qual caso l'indicazione dei vertici viene modificata per considerare la relazione di durata fra l'inizio e la fine dell'evento.

Per esempio, le espressioni appena citate possono essere tradotte in linguaggio utente nel seguente modo:

- il pranzo viene servito dopo la colazione → "lunch is after breakfast";
- l'infuso di erbe deve essere preso fra le 12:00 e le 20:00 → "herbal tea begins after about 12:00 but before about 20:00";
- l'aspirina deve essere presa a stomaco pieno e preferibilmente entro venti minuti dalla fine della cena → "aspirin intake begins after 0 but before about 20min compared to dinner ends";
- la cena è un'attività che dura circa 1 ora → "dinner length is on about 1h".

Traduttore basato su conoscenza temporale

Il secondo elemento di traduzione ha almeno due funzioni principali: da un lato deve tradurre le indicazioni temporali espresse come ore, minuti e secondi in un valore numerico coerente all'interno della rete temporale e dall'altro deve sostituire eventuali termini, utili per aumentare la potenza espressiva concessa all'utente, in sinonimi presenti nel linguaggio utente.

Per far fronte al problema della traduzione delle indicazioni temporali è possibile utilizzare le informazioni inerenti l'origine dei tempi ed il quanto temporale. L'unica accortezza da applicare in questo passaggio è quella di sottrarre al valore temporale, indicato dall'utente, il valore dell'origine dei tempi se e solo se il secondo oggetto temporale è omesso. In questo modo è possibile comprendere se i valori indicati sono assoluti o se sono relativi a differenze temporali fra i due punti presi in esame.

Sempre in questo livello è possibile riportare le espressioni utilizzate nel linguaggio comune in termini accettati dal linguaggio utente. Ne sono un esempio le espressioni “the beginning of <oggetto>” e “the end of <oggetto>” che possono essere tradotte rispettivamente con “<oggetto> begins” ed “<oggetto> ends” o l'espressione “takes” che traduce “length”.

Implementazione

Le componenti descritte in questa sezione sono state implementate in C++ tramite le classi “UserLanguageTranslator” e “BasicKnowledgeTranslator”; la prima con lo scopo di tradurre le espressioni da linguaggio utente in vincoli da applicare alla rete a cui si riferiscono, mentre la seconda serve a ridurre la potenza espressiva fornita dall'utente e a tradurre le indicazioni temporali in numeri reali. Gli oggetti di “BasicKnowledgeTranslator” hanno al loro interno un'istanza di “UserLanguageTranslator” allo scopo di convertire le espressioni semplificate in relazioni.

Entrambi sono dotati di un metodo per impostare la rete corrente, ossia la rete per cui generare i vincoli, assieme alle indicazioni sulla struttura della rete e degli eventi a lei collegati.

Un altro metodo comune è “translate” che riceve in ingresso una stringa, che contiene l'espressione da tradurre, e produce un vincolo coerente con la rete corrente o lancia un'eccezione se è stato riscontrato qualche termine non previsto.

In fase di scrittura del codice sono state compiute alcune scelte che vengono qui riportate per trasparenza:

- le espressioni quantitative “not before”, “not less than”, “after” e “more than” vengono tutte tradotte con il vincolo $[a, a, -, -)$. Tale scelta è

stata compiuta perché la forma proposta in letteratura per le ultime due espressioni non può essere rappresentata tramite trapezio, mentre le formulazioni $(a - \varepsilon_1, a + \varepsilon_1, -, -)$ o $(a, a + \varepsilon_2, -, -)$, con ε_1 ed ε_2 parametri da valutare, apparivano troppo arbitrarie. In modo analogo si è operato con le espressioni “not after”, “not more than”, “before” e “less than”.

- le espressioni del tipo “after about b but before about c” sono state modellate con $a = b - u \times (c - b)$ e $d = c + u \times (c - b)$ dove u è un parametro di incertezza definito dall’utente;
- le espressioni riconducibili ad “after about b” o “before about c” sono state modellate con $a = b \times (1 - u)$ o $d = c \times (1 + u)$, volendo fornire un margine fuzzy ai vincoli indicati dall’utente;
- le espressioni “at about b” e “on about b” sono state modellate con vincoli del tipo $(b \times (1 - u), b, b, b \times (1 + u))$.

Tali scelte permettono, quindi, di modellare vincoli di tipo crisp come “il tempo di cottura della torta varia tra la mezzora e l’ora”, vincoli fuzzy del tipo “la colazione inizia fra le 8:00 e le 8:30 circa” e loro varianti.

Il codice riportato in “translate.cpp” permette di testare, su shell di comando, le funzionalità delle classi descritte. Sulla riga di comando, il programma riceve il nome del file in cui è salvata la rete ed ogni espressione successivamente inserita viene tradotta e ne viene visualizzato il vincolo corrispondente.

4.2.2 Interfaccia Grafica per reti temporali fuzzy

L’applicazione “Ftcn_gui” è un’interfaccia grafica creata con lo scopo di fornire uno strumento che permetta ad utenti più o meno esperti di utilizzare il software di ragionamento temporale e gestire reti temporali.

L’applicazione è stata scritta in C++ e si appoggia al framework Qt per la gestione della parte grafica. Tale scelta è dovuta alle caratteristiche del software prodotto dalla TrollTech che prevede implementazioni della libreria per i principali sistemi operativi. Le differenze che si possono riscontrare fra le varie versioni dipendono principalmente dall’aspetto grafico dei singoli componenti che si adatta all’ambiente su cui è compilato ma il loro comportamento e le loro proprietà rimangono inalterate.

Il nucleo dell’applicazione è formato da due reti temporali: la prima, denominata original, ha lo scopo di gestire e mantenere le informazioni fornite dall’utente e ricavate da eventuali file di input; la seconda, denominata path, è

una copia dei dati forniti dall'utente sui quali viene applicato di volta in volta l'algoritmo di path-consistency. La scelta di gestire in parallelo entrambe le versioni della rete è dettata dal desiderio di fornire all'utente la possibilità di visualizzare sia i vincoli originali posti sulla rete che gli effetti che tali informazioni portano a livello globale di rete temporale.

Passando alla descrizione dell'applicazione si può notare come essa si componga di tre parti con i fini di:

1. generare reti temporali, caricare reti esistenti e memorizzare le elaborazioni su file;
2. rappresentare la rete attualmente in uso;
3. gestire gli input forniti dall'utente;

tali componenti possono essere individuate anche nell'immagine 4.3.

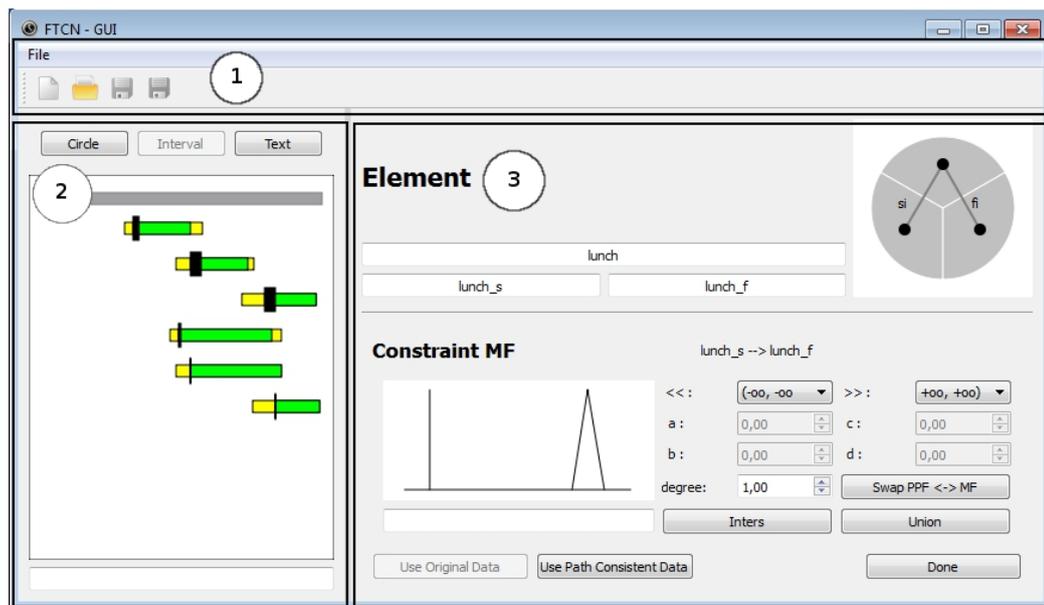


Figura 4.3: L'applicazione Ftcn_Gui e le sue componenti

Gestione delle reti temporali

Per quanto riguarda la gestione delle reti temporali, è stato fornito all'applicazione il classico menù File con le possibilità di:

- creare una nuova rete;

- aprire una rete già esistente;
- salvare la rete ed i vincoli attualmente espressi dall'utente;
- salvare la medesima rete in altro file.

I comportamenti delle opzioni fornite rispecchiano gli effetti attesi; si differenzia il solo metodo di generazione di una nuova rete temporale che, comporta una procedura che comprende le seguenti fasi:

1. inserimento di dati generali sulla rete, quali la sua descrizione, il valore di incertezza utilizzato dall'interprete del linguaggio umano (come già spiegato in precedenza e di default posto al 10%) ed i valori di origine e quanto temporale (posti di default rispettivamente a mezzanotte e ad 1 minuto);
2. inserimento dei dati relativi al numero di vertici della rete, quali l'opzione di inserimento del punto dell'origine dei tempi, il numero di intervalli (con l'opzione di generare in automatico anche i punti di inizio e fine intervallo) e gli eventuali punti aggiuntivi;
3. inserimento delle etichette degli eventi;

come riportati, a fini esemplificativi, in figura 4.4. La rete che viene così generata presenterà in automatico un vincolo $\{b, s\}$ fra l'origine dei tempi e tutti gli intervalli, vincoli $\{b, eq\}$ fra origine e tutti gli altri punti presenti nella rete, nonché le eventuali relazione di $\{si\}$ e $\{fi\}$ fra ogni intervallo ed i rispettivi punti di inizio e fine.

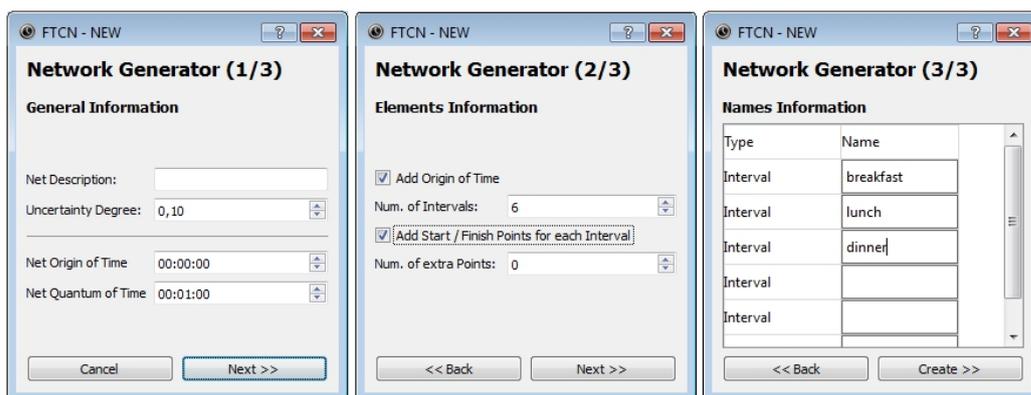


Figura 4.4: Successione dei tre passaggi per generare una nuova rete

Rappresentazione della rete in uso

Il secondo modulo che compone l'interfaccia grafica è costituito da tre rappresentazioni ispirate alla rete temporale che permettono di visualizzare informazioni tratte da "original" e "path", nonché di accedere, tramite interazione con il mouse, ai singoli vincoli della rete.

Una prima rappresentazione è quella testuale e visualizza le reti "original" e "path" in forma testuale grezza, indicando tramite sigle e valori numerici tutte le relazioni presenti.

Una seconda rappresentazione è quella "circolare" che si appoggia principalmente alla rete "original", ma prende in considerazione i risultati forniti dalla sua versione path-consistency nel caso in cui essa individui dei vincoli inconsistenti. L'idea che l'ha ispirata è quella di rappresentare la rete sottostante come grafo di eventi connessi fra loro; ogni evento è rappresentato da un punto che vuole riassumere il raggruppamento di vertici descritto in precedenza, mentre gli archi non orientati rappresentano possibili legami fra gli eventi stessi.

Allo scopo di semplificare la rappresentazione sono state fatte le seguenti scelte:

- i punti che rappresentano gli eventi sono normalmente colorati di nero. Nei casi in cui nelle reti "original" o "path" vi siano delle relazioni inconsistenti fra oggetti temporali appartenenti a tale evento, allora il punto viene colorato di rosso, per porre l'attenzione dell'utente su tale problema;
- gli archi fra eventi possono avere tre forme diverse a seconda dei legami presenti fra gli stessi: se vi è almeno una relazione inconsistente fra oggetti di due eventi distinti nella rete "original" o "path" allora l'arco fra i due eventi viene colorato di rosso; se fra due eventi l'utente ha posto almeno un vincolo allora l'arco viene tracciato in grigio scuro; infine, se fra due eventi l'utente non ha indicato nessun vincolo allora non viene indicato alcun arco, per evitare di appesantire la rappresentazione;
- facendo click con il mouse su un evento viene inoltrata la richiesta di accedere ai dati del singolo evento mentre trascinando il mouse fra un evento e l'altro vengono, invece, richiamati i dati della relazione fra i due eventi; infine, facendo click su un punto esterno al grafo vengono richiamate le informazioni generali della rete;
- la rappresentazione è completata con un anello diviso in settori centrati sui punti che rappresentano gli eventi, in modo tale che qualsiasi operazione compiuta nel settore sia rimandata all'evento di appartenenza.

La scelta di rendere visibili tali settori è dettata da un lato dalla volontà di evitare che l'utente sia obbligato ad essere troppo preciso nell'interazione con l'immagine; dall'altro, però, c'è anche la volontà di indicare in modo trasparente le zone di interesse, per non lasciare al soggetto l'incertezza sui limiti posti alle sue azioni.

La terza rappresentazione è quella tramite intervalli; vuole essere una raffigurazione visiva dei vincoli metrici presenti fra eventi. Per tale ragione, viene visualizzata se è presente il punto di origine dei tempi ed almeno un evento della rete è completo. Se entrambe le condizioni sono soddisfatte allora viene suddiviso lo spazio in fasce orizzontali, una per ogni evento, e se questo è completo viene rappresentato tramite tre rettangoli sovrapposti:

- uno giallo, che rappresenta la zona temporale massima in cui l'evento può compiersi;
- uno verde, che indica la zona temporale in cui l'evento si compie con grado massimo;
- uno nero, che indica la prima possibile esecuzione dell'evento con la durata minima.

Se all'interno dell'evento vi sono vincoli inconsistenti la rappresentazione viene sostituita con una fascia rossa, se, invece, l'evento non è completo allora la fascia è di colore grigio. Anche per questa rappresentazione, un click su una fascia orizzontale rimanda all'evento corrispondente mentre trascinare il mouse fra eventi diversi permette di accedere ai dati sui vincoli fra i due eventi.

Eventuali informazioni sul nome dell'evento su cui è posizionato il mouse o sull'istante temporale (limitatamente alla rappresentazione ad intervalli) preso in considerazione, vengono visualizzati a pedice della rappresentazione stessa.

Esempi delle tre rappresentazioni sono proposti in figura 4.5.

Gestione input forniti dall'utente

Il terzo componente dell'interfaccia grafica è il modulo per la gestione dei vincoli fra oggetti temporali. A seconda dell'interazione dell'utente con le rappresentazioni vengono visualizzate tre tipologie differenti di informazioni. La prima schermata è composta da:

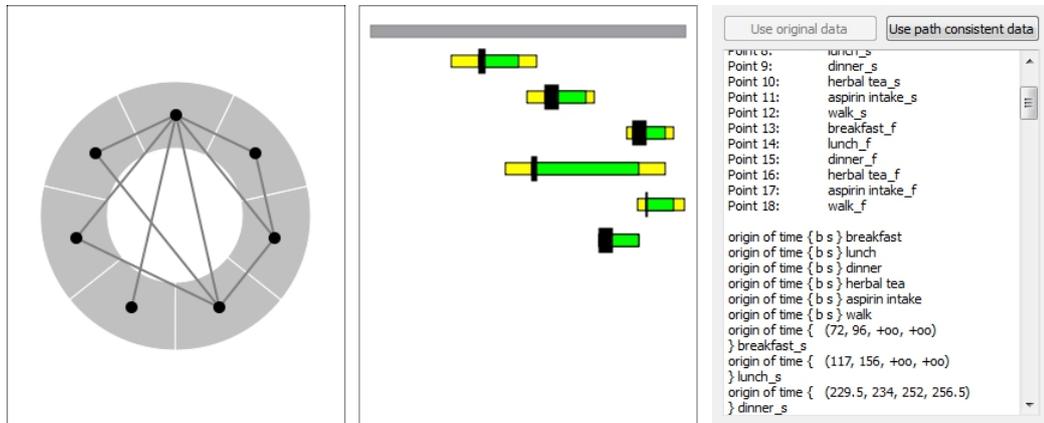


Figura 4.5: Esempi della medesima rete temporale secondo le tre rappresentazioni, i dati utilizzati verranno spiegati nel dettaglio in seguito

- le informazioni generali della rete, quali descrizione, grado di consistenza della rete dopo la propagazione dei vincoli, grado di incertezza e valori dell'origine dei tempi e del quanto temporale in uso. Di tutti questi parametri solo la descrizione della rete è modificabile;
- la componente per l'introduzione dei vincoli espressi in linguaggio utente. Seguendo le regole indicate in precedenza è possibile inserire tramite tastiera i vincoli che si desidera inserire nella rete. Tali vincoli andranno a sostituire quelli già esistenti.

La seconda schermata riguarda i singoli eventi e contiene una rappresentazione dell'evento e dei nodi che lo compongono. Se l'evento è completo viene, inoltre, visualizzato di default il vincolo fra i punti di inizio e fine intervallo, perché è l'unica informazione non banale presente nell'evento. La terza schermata riguarda le relazioni fra eventi temporali. Nella parte alta della schermata vengono espansi i singoli eventi e, sempre tramite mouse, è possibile accedere alle indicazioni esatte dei legami che li compongono. Sia nel secondo che nel terzo caso è possibile non solo visualizzare i vincoli fra gli oggetti, nella loro forma originale o path-consistency, ma anche di modificarli. In quest'ultimo caso le modifiche devono essere espressamente salvate altrimenti al cambio di schermata vengono dimenticate. La scelta è dovuta al fatto che ogni modifica della rete avviene "in tempo reale"; sebbene questa operazione sia facile per la rete "original", per quanto riguarda l'aggiornamento di "path" è prevista l'invocazione dell'algoritmo di path-consistency, che è computazionalmente pesante.

Di seguito sono riportate le immagini 4.6, 4.7 e 4.8 che rappresentano le tre schermate del modulo per la gestione dei vincoli.

Network General Information

Net Description: elder schedule

Consistency Degree: 1

Uncertainty Degree: 0,10

Origin of time: 00:00:00

Quantum of time: 00:10:00

Write a new Constraint:

Add Relation!

Figura 4.6: Esempio della componente che riassume le informazioni generali

Element

breakfast

breakfast_s breakfast_f

Constraint MF breakfast_s --> breakfast_f

Graph showing two peaks.

<< : (-∞, -∞) >> : (+∞, +∞)

a : 0,00 c : 0,00

b : 0,00 d : 0,00

degree: 1,00

Swap PPF <-> MF

Inters Union

Use Original Data Use Path Consistent Data Done

Figura 4.7: Esempio della componente per la gestione di un evento della rete

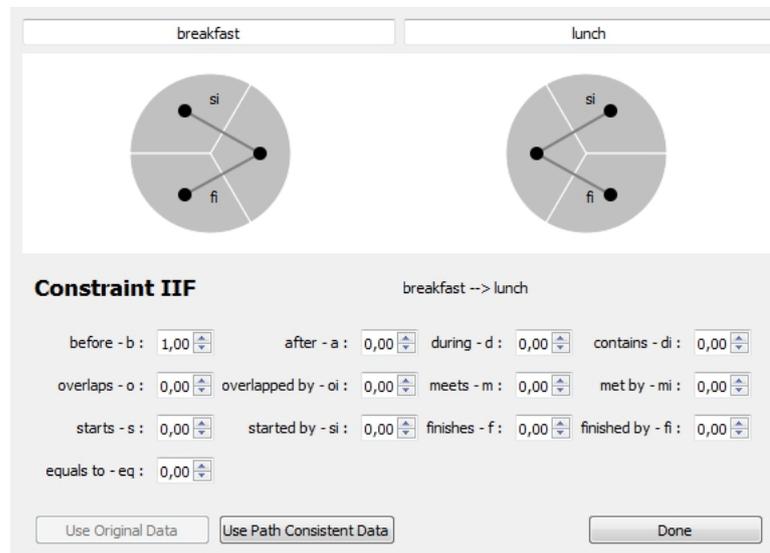


Figura 4.8: Esempio della componente per la gestione di una relazione fra eventi

Capitolo 5

Proposta di Supervisore Fuzzy

5.1 Introduzione al problema

I possibili utilizzi delle reti temporali fuzzy presentati in letteratura si orientano principalmente nel settore della validazione di possibili sviluppi di eventi sulla base dei vincoli temporali che li legano. Fra gli esempi possibili si ricordano:

- le applicazioni nell'ambito medico, per verificare se lo sviluppo di determinati sintomi sia o meno ricollegabile a specifiche malattie, descritte in [Badaloni, Falda, 2010];
- le applicazioni in ambito spaziale, per valutare le tempistiche delle applicazioni satellitari come descritto in [Badaloni et al, 2007];
- le applicazioni in ambito investigativo, per validare i possibili spostamenti dei sospettati e verificare la bontà degli scenari ipotizzati, come in [Badaloni et al, 2004b].

L'applicazione attualmente allo studio è legata, invece, alla tematica dell'assistenza agli anziani, con il particolare scopo di verificare il rispetto delle prescrizioni imposte all'utente e, in caso di necessità, assistendolo tramite indicazione delle attività da svolgere. Un sistema di questo tipo, per operare al meglio, deve essere accettato dall'utente: non deve essere visto come un rigido controllore, ma piuttosto come uno strumento per assicurare lo svolgimento di quelle azioni che permettono al soggetto stesso di continuare una vita indipendente.

Una simile applicazione è già stata proposta nell'ambito del progetto RoboCare [Cesta et al, 2011], alla quale questo prototipo si ispira, ma si differenzia perché in quest'ultima gli aspetti temporali furono gestiti tramite TCSP.

Il passaggio da una formulazione esatta ad una fuzzy presenta almeno un paio di vantaggi sia per l'utente dell'applicazione, sia per coloro che devono prendersi cura dell'anziano, come i familiari ed il personale medico.

Una prima motivazione è legata alla tipologia di vincoli utilizzati comunemente per descrivere il problema. A differenza di applicazioni industriali, in cui i tempi di processo e di reazione ai segnali sono, in prima approssimazione, noti e fissati, nel caso di assistenza di un soggetto anziano i tempi per compiere le azioni possono presentare ampie variazioni temporali dovute, per esempio allo stato di salute dell'utente.

Una seconda motivazione è legata alla capacità di tradurre i suggerimenti temporali forniti dall'utente, o da quanti se ne prendono cura, tramite un modulo per l'interpretazione del linguaggio. Questo compito risulta più difficile se si utilizza una formulazione crisp al problema perché non modella le tempistiche secondo gradi di preferenza diversi.

Si prenda a riguardo il caso in cui l'utente debba iniziare a pranzare tra le 12.00 e le 13.00, con la possibilità di discostarsi da tale intervallo di al più 30 minuti. In una formulazione del problema tramite TCSP, il dominio dell'inizio del pranzo è un intervallo $[a, b]$ con $a \in [11:30, 12:00]$ e $b \in [13:00, 13:30]$. Una formulazione rigida del problema prevede, quindi, che le soluzioni 11:40 o 13:10 non siano considerate valide, mentre in una formulazione più flessibile le soluzioni 11:40, 12:30 e 13:10 risulterebbero tutte valide a pari grado.

Queste sfumature possono essere gestite se si utilizza il sistema di ragionamento fuzzy. Modellando il vincolo come la relazione fra l'origine dei tempi e l'inizio del pranzo secondo la forma $\{(11:30, 12:00, 13:00, 13:30)\}$ le soluzioni fra 12:00 e 13:00 sono viste come ottime (perché a grado massimo), ma vengono considerate accettabili le soluzioni fra le 11:30 e le 12:00 o fra le 13:00 e le 13:30 anche se, in questi casi, il grado di preferenza risulta minore.

Esempi di tali scenari sono riportati in figura 5.1.

Queste sono alcune delle motivazioni del passaggio dalla formulazione crisp – TCSP a quella fuzzy, le altre sono legate all'interazione del sistema stesso con l'utente e la possibilità di modellare tali comunicazioni, anche tramite canali diversi, a seconda del grado di rispetto dei vincoli. Ritornando all'esempio precedente è ipotizzabile che, se l'utente non ha ancora iniziato il pranzo all'una, allora il sistema dovrebbe inviare una prima segnalazione, per esempio tramite indicazioni visive o luminose, permettendo così compiere l'azione entro tempi accettabili. Nel caso in cui l'azione alle 13:30 non si sia ancora verificata allora la segnalazione dovrebbe avere un peso sensoriale maggiore e si potrebbe ipotizzare di integrare l'informazione visiva con una acustica, per aumentare le possibilità che l'utente percepisca il messaggio. In

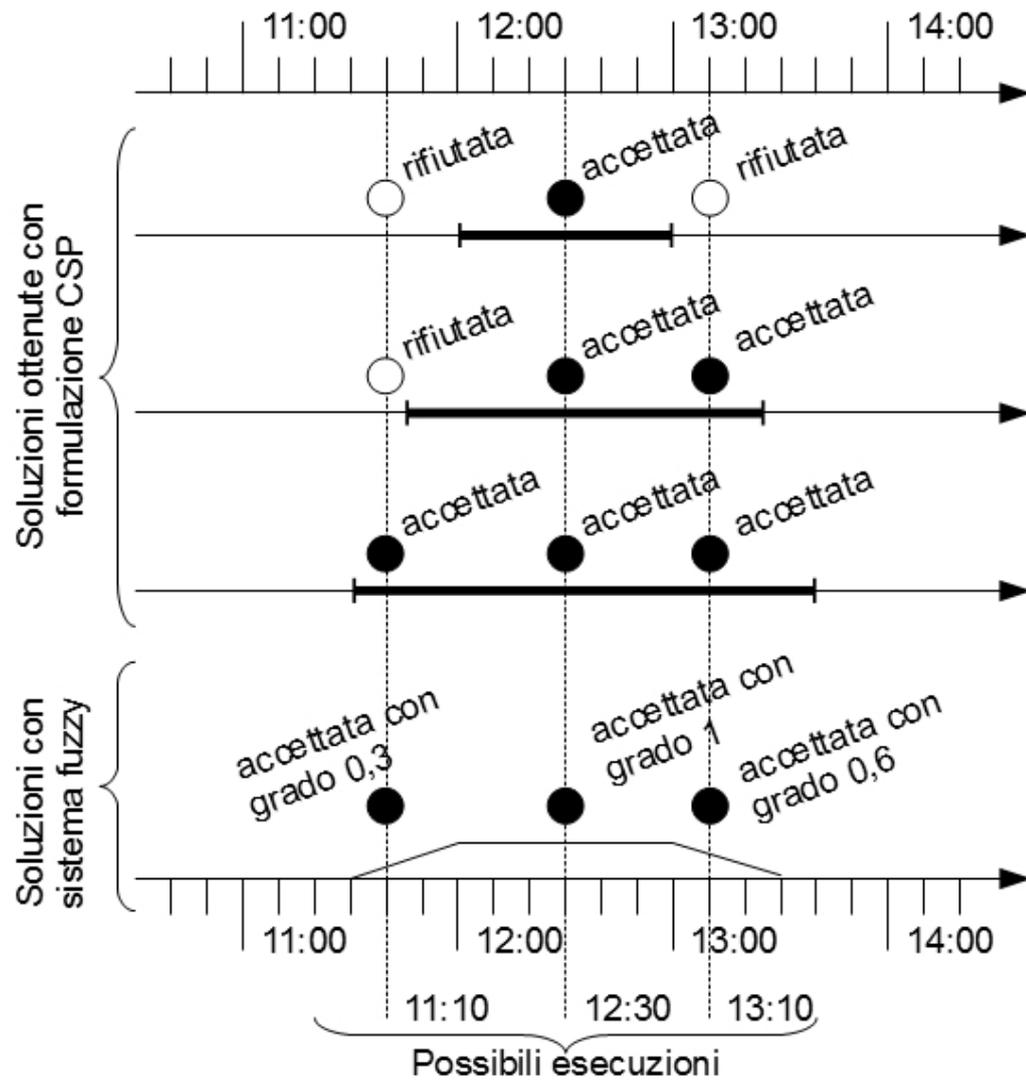


Figura 5.1: Possibili reazioni di sistemi basati su CSP (i primi tre) o ragionamento fuzzy di fronte a possibili inizi del pranzo

modo analogo è possibile operare nei casi in cui l'azione di inizio del pranzo avvenga prima delle 11:30 o, comunque, prima di mezzogiorno.

5.2 Ipotesi di sistema a tre livelli

Un sistema di supervisione fuzzy, per risultare non invasivo verso l'utente e comunque funzionante, dovrebbe essere progettato come componente di un sistema più generale di assistenza anziano.

Analizzando quanto presente in letteratura è possibile ipotizzare il seguente scenario: l'utente è immerso in un ambiente domestico dotato di sensori che convogliano le loro rilevazioni ad un sistema che elabora tali informazioni e genera, se necessario, messaggi verso l'utente del sistema o quanti se ne prendono cura.

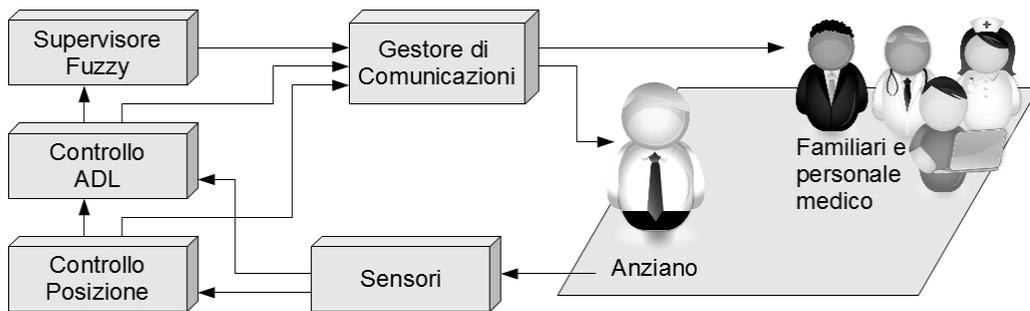


Figura 5.2: Esempio di sistema a tre livelli

Il sistema può essere modellato tramite un'architettura verticale costituita da tre livelli con funzionalità diverse, e riportate in figura 5.2:

1. un primo strato dovrebbe fornire indicazioni sulla posizione e sullo stato dell'utente. La posizione, come proposto in letteratura, può essere rilevata in un ambiente domestico tramite sensori di pressione posti sul pavimento o videocamere, mentre per gli spostamenti all'esterno è possibile utilizzare un localizzatore GPS, attualmente integrato in tutti gli smartphone presenti in commercio. Le informazioni sulla posizione possono essere utilizzate per decidere dove inviare le informazioni all'utente o dove indirizzare quanti devono prestare soccorso; è inutile, infatti, generare una segnalazione visiva in cucina se l'utente si trova in camera da letto o far arrivare i familiari a casa se l'utente ha avuto un mancamento al parco. Le indicazioni sullo stato dell'utente possono essere generate, nel caso della postura (seduta, sdraiata, in piedi...)

tramite accelerometri o videocamere, mentre altri strumenti a contatto con la pelle possono valutare altri parametri, come il battito cardiaco. Queste indicazioni servono per fornire una rappresentazione il più possibile completa sullo stato fisico del soggetto e generare allarmi, se vengono rilevati fenomeni come le cadute;

2. un secondo strato dovrebbe fornire informazioni sulle azioni che vengono compiute dall'utente, quali dormire, mangiare, prendere i medicinali, ecc... Queste informazioni possono risultare utili all'utente nel caso di temporanei problemi di memoria (per esempio rispondendo a domande del tipo "ho preso la pastiglia X oggi?") e possono essere utilizzate a posteriore dal medico o dai familiari per valutare lo stato mentale e fisico dell'anziano. Per esempio, se un utente riduce gradualmente i tempi che utilizza ogni giorno per la sua camminata è possibile ipotizzare che si stanchi sempre più facilmente, mentre la ripetizione o il salto dei pasti o delle medicazioni può indicare un decadimento delle facoltà mentali;
3. il terzo strato dovrebbe essere dedicato al controllore fuzzy. Utilizzando le informazioni dagli strati sottostanti ed integrando il sistema con vincoli temporali è possibile assistere l'utente andando a confrontare il comportamento tenuto con i vincoli imposti ed eventualmente suggerendo come modificare le azioni per renderle più vicine al comportamento atteso.

Ipotizzando questo modello a tre strati è stato costruito un prototipo di simulatore di sorvegliante fuzzy.

5.3 Algoritmo per il controllo

L'attuale versione dell'algoritmo di controllo si ispira a quello proposto in [Cesta et al, 2011] e prevede l'ipotesi semplificativa che le letture dei sensori siano corrette e che ogni azione avvenga al più una volta al giorno. L'algoritmo si appoggia sui seguenti oggetti:

- la rete contenente i vincoli originali ricavati da un file di input (original);
- la rete che rappresenta lo stato attuale (current);
- la rete che rappresenta il comportamento atteso o desiderato (desired);
- una rete di supporto (support);

- due code, la prima che tenga traccia dei vincoli originali attualmente verificati in modo completo (okQueue) e la seconda che memorizzi i vincoli attualmente verificati ma con grado compreso fra 0 ed 1 (maybeQueue).

La rete current ha lo scopo di memorizzare le informazioni su quanto avviene nell'ambiente reale e viene generata a partire da quella originale imponendo le seguenti relazioni:

- $\{b, s\}$ per tutti i vincoli fra l'origine dei tempi ed ogni intervallo;
- $\{b, eq\}$ per tutti i vincoli fra l'origine dei tempi ed ogni altro punto;
- $\{s\}$ per il vincolo fra ogni intervallo ed il rispettivo punto di inizio evento;
- $\{f\}$ per il vincolo fra ogni intervallo ed il rispettivo punto di fine evento;
- $\{b\}$ per il vincolo fra ogni punto di inizio evento ed il corrispondente punto di fine;
- ogni altro vincolo viene posto ad identità.

Ogni volta che lo stato corrente deve essere aggiornato servono due passaggi: innanzitutto è necessario aggiornare le relazioni origine dei tempi – inizio/fine eventi, secondo le rilevazioni dei sensori; è possibile, quindi, aggiornare tutte le altre relazioni presenti all'interno della rete invocando l'algoritmo di path-consistency sulla rete aggiornata.

Supponendo di dover compiere tale operazione ad un generico istante t , allora le modifiche dei vincoli avvengono secondo le seguenti indicazioni:

- se all'istante t l'azione non ha ancora avuto inizio, allora l'inizio e la fine dell'evento avverranno ad un istante $t' > t$; quindi è possibile imporre la relazione $\{(t, t, +\infty, +\infty)\}$ fra l'origine dei tempi ed i punti di inizio e fine evento;
- se l'evento ha iniziato a verificarsi all'istante t allora la relazione fra origine dei tempi e punto di inizio evento non è più incerta, ma determinata e fissata all'istante t , quindi la relazione fra i due viene posta a $\{(t, t, t, t)\}$;
- fintanto che l'azione è in corso all'istante t , allora si può presumere che l'evento terminerà in un istante $t' > t$ e, quindi, la relazione fra origine dei tempi e punto di fine evento potrà essere posta a $\{(t, t, +\infty, +\infty)\}$;

- se, infine, l'azione è terminata nell'istante t , allora la relazione fra origine dei tempi e punto di fine evento non è più incerta, ma determinata ed è possibile porre la relazione a $\{[t, t, t, t]\}$.

Una volta aggiornate le relazioni di inizio e fine evento tramite l'algoritmo di path-consistency vengono calcolate anche tutte le altre relazioni e si può, per esempio, determinare che la durata di un evento ha la forma $\{(0, 0, +\infty, +\infty)\}$ se deve ancora iniziare, $\{(t'', t'', +\infty, +\infty)\}$ se l'azione è in corso da t'' istanti temporali o $\{[t''', t''', t''', t''']\}$ se l'azione si è conclusa in t''' istanti temporali; discorso analogo può essere applicato per le relazioni fra i vari eventi.

Prima di descrivere il comportamento del supervisore devono essere specificati almeno altri due fattori: conoscere come modellare il comportamento atteso dell'utente e come valutare eventuali discrepanze tra lo sviluppo reale degli eventi e quello desiderato.

Per quanto concerne la rete "desired", che riassume il comportamento atteso, all'inizio dell'elaborazione (quando nessuna azione è stata compiuta dall'utente) è sufficiente inicializzarla come una copia della rete "original" sulla quale è stato applicato l'algoritmo di path-consistency. Qualora l'anziano inizi a produrre degli eventi, più o meno conformi al comportamento desiderato, tale rete deve essere aggiornata tenendo conto di tali modifiche, eliminando i vincoli che sono stati violati e modificando le relazioni in modo da massimizzare il rispetto degli obiettivi imposti.

Riprendendo l'esempio precedente, si supponga che il pranzo debba iniziare fra mezzogiorno e l'una e che la sua durata sia solitamente di 1 ora. All'inizio la rete desired prevederà, fra le altre, le seguenti relazioni:

- origin of time $\{ (11:30, 12:00, 13:00, 13:30) \}$ lunch_s;
- origin of time $\{ (12:15, 13:00, 14:00, 14:45) \}$ lunch_f;
- lunch_s $(0:45, 1:00, 1:00, 1:15)$ lunch_f.

Ipotizzando che il pranzo inizi alle 12:15 allora tutti i vincoli vengono mantenuti e le relazioni vengono riscritte come:

- origin of time $\{ [12:15, 12:15, 12:15, 12:15] \}$ lunch_s;
- origin of time $\{ (13:00, 13:15, 13:15, 13:30) \}$ lunch_f;
- lunch_s $\{(0:45, 1:00, 1:00, 1:15)\}$ lunch_f.

Nel caso in cui il pranzo inizi in anticipo, per esempio alle 11:15, allora il vincolo sull'inizio del pranzo dovrebbe essere escluso perché non è stato rispettato. Per massimizzare il numero di obiettivi imposti, la relazione di durata dovrebbe essere mantenuta generando, quindi, il seguente stato:

- origin of time { [11:15, 11:15, 11:15, 11:15] } lunch_s;
- origin of time { (12:00, 12:15, 12:15, 12:30) } lunch_f;
- lunch_s {(0:45, 1:00, 1:00, 1:15) } lunch_f.

Il secondo fattore da verificare è relativo alle discrepanze, in run-time, fra il comportamento reale e quello desiderato. Ipotizzando che una qualsiasi relazione metrica (sull'inizio o la fine di un evento o sulla sua durata) all'interno della rete desired sia nella forma (a, b, c, d) e ricordando che le relazioni nella rete corrente hanno la forma $(t, t, +\infty, +\infty)$ o $[t', t', t', t']$ allora si possono osservare i seguenti fatti:

- se $t < c$ allora significa che vi è ancora la possibilità che la relazione sia rispettata con grado massimo;
- se $t \in [c, d]$ allora l'obiettivo può ancora essere rispettato, ma non con grado massimo;
- se $t > d$, allora il vincolo è sicuramente non atteso;
- se $t' \in (b, c)$ allora il vincolo è stato verificato con il suo grado massimo;
- se $t' \in (a, b) \vee t' \in (c, d)$ allora il vincolo si può considerare verificato anche se non segue il comportamento ottimo;
- se $t' < a \wedge t' > d$ allora il vincolo non è stato soddisfatto.

Osservando tutti e sei i casi sopra riportati è possibile ipotizzare un comportamento più generale che può essere applicato per il confronto fra la rete current e desired. Dato il vincolo R_{ij}^C , fra gli oggetti temporali i e j della rete current, e R_{ij}^D , corrispondente vincolo nella rete desired, allora il grado di rispetto fra il comportamento reale e quello atteso si può ricavare dal grado massimo del vincolo che si ottiene dall'operazione di intersezione $R_{ij}^C \otimes R_{ij}^D$. Se tale grado è pari ad 1 allora il comportamento reale è o può essere conforme al comportamento atteso; se il grado è compreso in $(0, 1)$ allora il vincolo è rispettato anche se non secondo il grado massimo; se, infine, il grado è 0 allora il vincolo non è stato rispettato.

Tenendo presente tali aspetti è, ora, possibile indicare l'algoritmo di controllo tramite lo pseudo-codice seguente:

```

while true do
  updateCurrent();
  if(differences()) then
    desired ← current
    while(getArc(Rij))
      support ← desired ∪ Rij
      support.pathConsistency()
      Rij ← support.getRelation(i, j)
      if(Rij.getMaxDegree() > 0)
        desired ← support
        if(Rij.getMaxDegree() < 1)
          message(Rij, average)
          maybeQueue.enqueue(i, j)
        else
          okQueue.enqueue(i, j)
        end if
      end if
    end while
  end if
end while

```

Il suo comportamento è quello atteso, ossia: ad ogni iterazione viene aggiornato lo stato corrente del sistema e, se non vengono rilevate discrepanze fra comportamento reale e quello atteso, non occorre procedere con ulteriori operazioni. Nel caso in cui vi sia almeno un vincolo non pienamente soddisfatto, allora è necessario procedere all'esame di tutti i vincoli imposti per suggerire all'utente eventuali comportamenti utili per restare il più possibile coerenti con quanto ci si è prefissati.

Per ogni vincolo ancora utilizzato, viene valutata la sua bontà rispetto allo stato corrente e a tutti i vincoli finora approvati; nel caso in cui tale relazione sia violata, allora viene scartata dopo aver segnalato tale grave violazione. Nel caso in cui, invece, il vincolo sia entro parametri accettabili, allora viene confermato e lasciato nel comportamento atteso, anche se un suo passaggio dal soddisfacimento massimo ad uno intermedio genera una segnalazione verso l'utente allo scopo di avvisarlo di eventuali rischi imminenti.

5.4 Prototipo

Il prototipo generato presenta come nucleo centrale l'algoritmo di supervisione descritto sopra, mentre le restanti parti costituiscono l'interfaccia grafica messa a disposizione dell'utente per simulare il funzionamento dello stesso. Tale interfaccia presenta quattro componenti fondamentali (orologio di sistema, sensori, rappresentazione grafica e messaggistica verso l'utente) alcuni dei quali sott'intendono problematiche che vengono indicate di seguito.

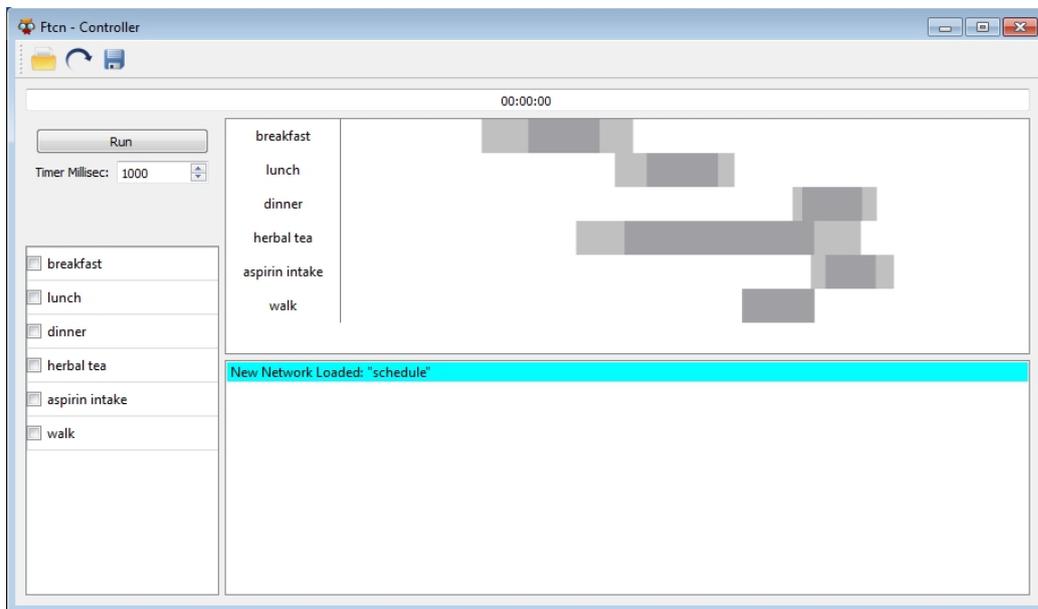


Figura 5.3: Interfaccia del simulatore per la supervisione del comportamento di soggetti anziani

5.4.1 Orologio di sistema

Una prima importante scelta per la realizzazione di un controllore di vincoli fuzzy è come modellare il ciclo:

```
while true do
  [...]
end while
```

utilizzato nello pseudo-codice dell'algoritmo di controllo.

Sono ipotizzabili almeno due differenti approcci al problema: da un lato si può pensare di invocare il controllore ogni volta che un evento si verifica

o dovrebbe verificarsi (event driven), dall'altro si può ipotizzare di invocare periodicamente il controllore secondo un lasso di tempo prefissato (clock driven).

Il vantaggio del primo metodo è che il controllore viene invocato solo quando si manifestano gli eventi, ma di contro se questi sono troppo frequenti si rischia di invocare troppo spesso il controllore, appesantendo così il carico computazionale. Il secondo metodo prevede l'invocazione del controllore a prescindere dalle rilevazioni dei sensori e questo provoca computazioni non necessarie, anche se la verifica di scostamento della rete corrente da quella desiderata permette di ridurre le operazioni eseguite. Aspetti vantaggiosi del metodo sono la possibilità di garantire il funzionamento in real-time (tarando adeguatamente la finestra di clock) e la possibilità di verificare le letture dei sensori (andando ad esaminare quali comportamenti sono persistenti e quali occasionali). Quest'ultimo metodo è stato implementato nel simulatore, con un orologio dell'applicazione che viene inizializzato secondo il valore dell'origine dei tempi ed incrementato di un quanto temporale ad ogni clock. Essendo una simulazione del processo è possibile velocizzare i clock rispetto al valore reale, nonché bloccarla e riprenderla a seconda delle necessità dell'utilizzatore.

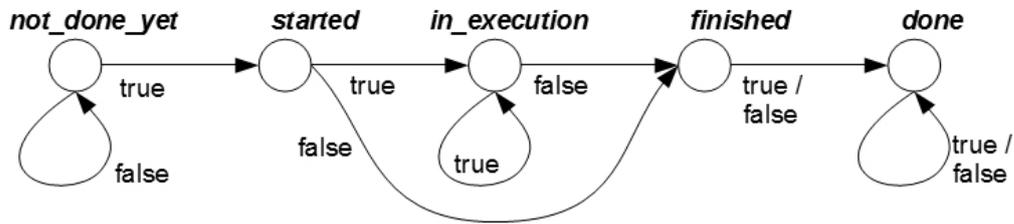
5.4.2 Sensori e stato degli eventi

Un secondo componente è il simulatore dello stato degli eventi, attualmente implementato come una lista di checkbox, una per evento da osservare.

In base all'evoluzione del valore letto nei componenti è possibile dedurre lo stato dell'evento nel sistema. Ricordando le ipotesi semplificative secondo le quali un evento avviene nel sistema al più una volta e le rilevazioni avvengono ad ogni clock del sistema, è possibile prevedere cinque stati attraverso i quali evolve l'evento. Essi sono:

- `not_done_yet`, se l'evento non è ancora occorso;
- `started`, se l'evento ha avuto inizio nel periodo precedente al clock;
- `in_execution`, se l'evento era in corso durante l'ultimo periodo;
- `finished`, se l'evento si è concluso nel periodo appena trascorso;
- `done`, se l'evento si è concluso da più di un periodo.

L'evoluzione può essere rappresentata tramite il seguente automa a stati finiti:

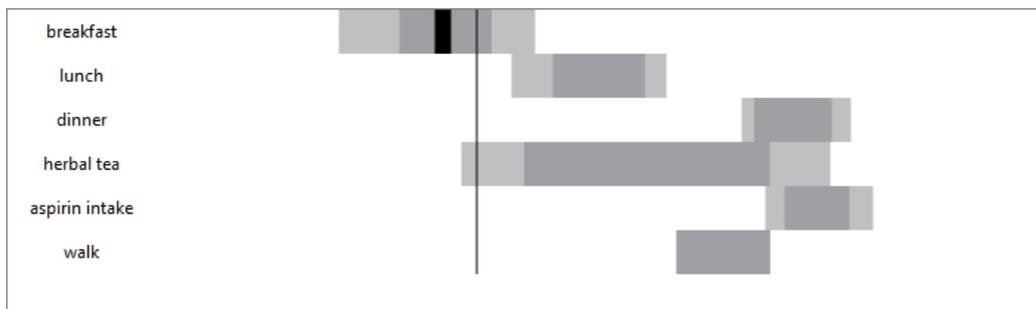


5.4.3 Rappresentazione degli eventi

Il modulo di rappresentazione degli eventi ha il solo scopo di fornire una prima stima delle informazioni contenute nelle reti, che rappresentano l'evoluzione corrente del sistema e quella desiderata.

Una linea verticale rappresenta l'istante attuale del sistema. Per semplicità è stata prevista una finestra temporale di 24 ore per simulare una giornata tipo dell'utente. Allo scopo di generalizzare l'ambito di utilizzo, al termine della finestra temporale di simulazione, il programma non si arresta, ma semplicemente raddoppia l'intervallo di tempo rappresentato. Questo perché il sistema è nato con lo scopo di simulare un controllore di un soggetto anziano, ma, visto che tutte le informazioni necessarie vengono ricavate dalla rete temporale ricevuta in input, è ipotizzabile che il sistema funzioni su qualsiasi scenario modellato.

Per ogni evento viene rappresentata l'attuale esecuzione tramite un rettangolo nero, mentre con tonalità diverse di grigio vengono rappresentati gli intervalli nei quali lo svolgersi dell'evento è visto come desiderabile (a grado massimo e di colore grigio scuro) o comunque accettabile (per le zone a grado fra 0 e 1 in grigio chiaro).



5.4.4 Interazione con l'utente

Lo scopo dell'ultimo modulo è simulare i messaggi da inviare all'utente in caso di comportamenti non coerenti con quelli attesi. I messaggi vengono scritti

su sfondi colorati diversi per distinguere il tipo di informazioni che vengono trasmesse. Attualmente sono utilizzate tre diverse tipologie di messaggi:

- in azzurro le comunicazioni tecniche come la descrizione della rete in uso;
- in giallo i messaggi a media priorità, ossia quelli relativi a vincoli che non sono violati ma con grado inferiore ad 1;
- in rosso le informazioni a priorità massima, ossia quelle relative ai vincoli forniti dall'utente ma violati nel comportamento attuale.

Come già accennato in precedenza è ipotizzabile che in un sistema reale le comunicazioni gialle siano eseguite, per esempio, tramite messaggio scritto su uno schermo, mentre quelle a priorità più alta debbano essere affiancate da segnali audio. Un aspetto critico in questo genere di sistemi è collegato alla modalità con cui le violazioni dei vincoli vengono verbalizzate, che devono permettere agli anziani di comprendere le ragioni che provocano tali avvertimenti.

Tale traduzione viene attualmente implementata nella forma:

You should have <primo evento> <relazioni> <secondo evento>

se il vincolo coinvolge due intervalli. Per esempio se il vincolo violato prevede che si prenda l'aspirina dopo la cena, con corrispondente vincolo nella rete del tipo aspirin intake {a} dinner, allora la frase generata risulta: *You should have aspirin intake after dinner*. Varianti di tale forma vengono utilizzate anche per le relazioni punto – intervallo ed intervallo – punto. Nel caso di vincoli punto – punto devono essere valutati i vertici coinvolti nella relazione, per generare messaggi facilmente comprensibili. In questi casi i problemi sono legati ad eventi occorsi troppo presto o che devono essere compiuti; se sono occorsi troppo presto, si genereranno delle annotazioni per un uso futuro, mentre nel caso opposto saranno generate delle esortazioni affinché l'evento si verifichi il prima possibile.

Per esempio se il pranzo deve iniziare fra mezzogiorno e l'una con un margine di 30 minuti e l'azione avviene alle 11:45 allora verrà generato il messaggio: *Next time you should start lunch later* visto che nessun comportamento dell'utente può modificare un'azione già compiuta; d'altra parte se alle 13:00 l'utente non ha ancora pranzato, allora verrà prodotta la segnalazione *You should start lunch as soon as possible* per indurre l'utente a compiere tale azione il prima possibile.

In modo analogo se la durata del pasto dovrebbe durare circa un'ora, ma dopo 60 minuti l'azione non si è ancora conclusa, allora il messaggio che verrà

prodotto avrà la forma *You should use less time for lunch*, nel caso opposto in cui il pranzo duri 30 minuti, la segnalazione avrà la forma *Next time you should use more time for lunch*.

Astraendo dal simulatore attualmente realizzato è doveroso ricordare che il problema dell'interazione con l'utente è uno degli aspetti salienti dei sistemi di assistenza anziani che possono determinare il successo o il fallimento di un prodotto commerciale indipendentemente dalle sue caratteristiche tecniche.

Come già accennato in letteratura, fra le caratteristiche essenziali indicate dagli utenti vengono segnalate la necessità di intuire che il sistema è in funzione, anche nel caso in cui il comportamento dell'utente è eseguito entro i parametri ottimali e quindi nessun feedback viene prodotto. È necessario, inoltre, che i messaggi siano posti in forma tale da non sembrare delle imposizioni per l'utente, ma piuttosto dei suggerimenti per meglio affrontare le varie situazioni. Infine è preferibile che le interfacce prodotte contengano il minor numero possibile di elementi, per facilitarne la lettura e generare meno confusione possibile.

5.5 Osservazioni e proposte

In fase di testing del simulatore sono emersi alcuni elementi di cui sarà utile tener conto in un futuro sviluppo del progetto. Un fattore importante è la determinazione del quanto temporale utilizzato nella rete. Tale valore, utile per trasformare le espressioni dal formato ore – minuti – secondi in valori reali, assume un significato importante nel supervisore fuzzy, visto che viene utilizzato come intervallo fra due clock del sistema. Per una buona taratura di tale parametro alcuni fra gli elementi utili da considerare sono i seguenti:

- il quanto temporale dovrebbe essere impostato ad un valore al più uguale alla minima fra le massime durate degli eventi. Valori troppo elevati comportano un calcolo della durata, vista appunto come differenza fra fine ed inizio dell'evento (che sono multipli del quanto stesso), che portano a non rispettare i vincoli imposti, secondo il controllore, e ciò può generare incomprensioni all'utente che potenzialmente ha eseguito l'azione nel tempo desiderato;
- il quanto temporale dovrebbe essere tale da permettere di distinguere eventi separati. Per comprendere meglio tale affermazione prendiamo ad esempio il vincolo pills intake is after lunch che comporta la relazione pills intake_s a lunch_f. Se la finestra tra due controlli è troppo lunga si rischia di vedere gli eventi pills intake_s e lunch_f come coincidenti, quando magari l'utente li ha compiuti separatamente. Infatti

se ipotizziamo una finestra di 10 minuti, la fine del pranzo al primo minuto e l'inizio del prendere le pillole al nono, allora per un controllore umano il vincolo è rispettato, mentre per il controllore fuzzy il vincolo è violato perché sono entrambi posti al valore di clock successivo;

- il quanto deve essere lungo abbastanza da permettere di completare un run dell'algoritmo di controllo. Essendo previste varie operazioni di path-consistency al caso pessimo, l'utilizzo di un quanto non sufficientemente lungo porta il sistema a richiamare l'algoritmo di controllo quando quest'ultimo non ha ancora concluso la propria esecuzione. Nei sistemi real-time la cosa viene solitamente gestita o interrompendo l'attuale processo per concluderlo solo quando il processore è libero, o ritardando il nuovo processo. Entrambi i casi in questo specifico esempio non sembrano accettabili.

Un ultimo aspetto che deve essere considerato è quello sulle ipotesi semplificative applicate al simulatore, ovvero che le letture dei sensori sono sempre esatte e che un'evento avviene al più una sola volta. La prima ipotesi, per quanto sia utile in fase di progetto, risulta dannosa in un sistema reale; la seconda è in parte legata alla prima, in quanto una lettura errata della realtà potrebbe portare il sistema ad ipotizzare l'esecuzione di un'azione da parte dell'utente quando questo la realizzerà solo successivamente: in questo caso il sistema si orienta in base alla falsa lettura e genera messaggi che non tengono conto del reale andamento dei fatti. Tale scenario, nel caso di eventi fortemente legati fra loro, porterebbe il controllore ad essere inutile per l'utente nella giornata in corso. Un altro aspetto rilevante della ripetizione delle azioni è legato alla possibilità dell'utente di rispondere ad alcune segnalazioni mutando il suo comportamento. Si ipotizzi la presenza di un vincolo che impone all'anziano di pranzare dopo l'una, ma che il soggetto se ne dimentichi ed inizi a mangiare verso mezzogiorno. Attualmente il sistema genera un messaggio di avvertimento per il futuro ma, ipotizzando una segnalazione del tipo *You should have lunch after 13:00 today* l'utente potrebbe decidere di posticipare il pranzo all'ora stabilita. Questa soluzione, in prima approssimazione, potrebbe essere applicata al sistema inserendo non uno ma due trapezi della forma $\{[t', t', t', t'], [t'', t'', t'', t'']\}$ nella relazione fra origine dei tempi e inizio del pranzo, anche se è necessario approfondire come questa modifica vada ad interagire con i vari vincoli imposti nel sistema. Si può notare, inoltre, come tale modifica apparentemente facile in un sistema di ragionamento fuzzy risulti impossibile nella normale formulazione del problema tramite TCSP, dove ad ogni variabile viene assegnato un solo valore del dominio.

Capitolo 6

Simulazione di scenario reale

Quanto detto fino a questo punto è stato utilizzato per simulare uno scenario con lo scopo di verificare le capacità del sistema di ragionamento temporale fuzzy di elaborare vincoli temporali e di utilizzarli per controllarne l'esecuzione.

6.1 Descrizione del Problema

Si supponga di dover controllare una giornata tipo di un soggetto anziano con un grado di decadimento fisico e mentale tale da permettergli di vivere in modo autonomo nella propria abitazione. Si presuma, inoltre, che la sua dimora sia dotata di sensori e componenti software che permettano di comprendere con chiarezza ciò che viene compiuto dal soggetto.

Fra le attività che l'utente deve compiere nella sua routine giornaliera ve ne sono sei, legate agli aspetti di alimentazione e di salute, che presentano dei vincoli temporali da rispettare perché prescritti dal medico curante. Tali attività sono:

- colazione: non dovrebbe iniziare prima delle 8:00 e la sua durata media è di circa 30 minuti;
- pranzo: non dovrebbe iniziare prima delle 13:00 e, comunque, fra le 4 e le 6 ore dalla fine della colazione. La sua durata è di circa 1 ora;
- cena: dovrebbe iniziare fra le 19:30 e le 21:00 e, comunque fra le 5 e le 6 ore dalla fine del pranzo. La sua durata media è di circa 1 ora;
- infuso d'erbe: dovrebbe essere preso fra le 12:00 e le 20:00 e nel complesso è un'azione che dura circa 20 minuti;

- aspirina: dovrebbe essere presa dopo la cena e ad al più 30 minuti dalla sua fine. È un'attività di circa 5 minuti.
- camminata: il soggetto deve camminare per circa 1 ora al giorno, fra le 17:00 e le 20:00.

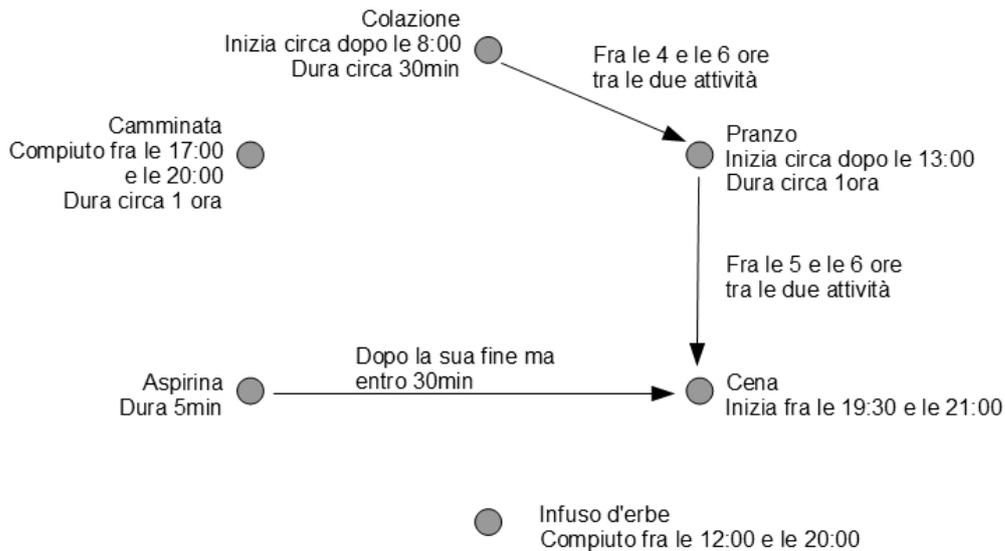


Figura 6.1: Esempio di rappresentazione tramite grafo dei vincoli dell'utente

6.2 Generazione della rete

Lo scenario proposto può essere modellato su una rete costituita da 6 eventi completi denominati: “breakfast”, “lunch”, “dinner”, “herbal tea”, “aspirin intake” e “walk”; sono imposti, inoltre, l'origine dei tempi alla mezzanotte del giorno, un quanto temporale di 5 minuti e un grado di incertezza del 25%.

6.2.1 Vincoli in linguaggio utente

Ipotizzando questi eventi, i vincoli sopra riportati possono essere così tradotti:

- l'espressione “la colazione non dovrebbe iniziare prima delle 8:00 e la sua durata media è di circa 30 minuti” può essere tradotta tramite le due relazioni

-
- breakfast begins after about 8:00
 - breakfast length is on about 30min
 - i vincoli sul pranzo che “non dovrebbe iniziare prima delle 13:00 e, comunque, fra le 4 e le 6 ore dalla fine della colazione; la sua durata è di circa 1 ora” possono essere riportati attraverso tre relazioni
 - lunch begins after about 13:00
 - lunch begins after about 4h but before about 6h compared to the end of breakfast
 - lunch takes on about 1h
 - l’espressione “la cena dovrebbe iniziare fra le 19:30 e le 21:00, e comunque fra le 5 e le 6 ore dalla fine del pranzo; la sua durata media è di circa 1 ora” può essere formalizzata come
 - dinner begins after about 19:30 but before about 21:00
 - dinner begins after about 5h but before about 6h compared to the end of lunch
 - dinner length is on about 1h
 - i vincoli sull’infuso d’erbe che “dovrebbe essere preso fra le 12:00 e le 20:00” e che “dura circa 20 minuti” si possono modellare tramite le relazioni:
 - herbal tea begins after about 12:00 but before about 20:00
 - herbal tea ends after about 12:00 but before about 20:00
 - herbal tea length is on about 20min
 - il vincolo “l’aspirina dovrebbe essere presa dopo la cena e ad al più 30 minuti dalla sua fine; è un’attività di circa 5 minuti” può essere tradotto come
 - aspirin intake begins after but before about 30min compared to the end of dinner
 - aspirin intake length is on about 5min
 - per quanto riguarda, infine, la camminata che il soggetto “deve fare per circa 1 ora al giorno, fra le 17:00 e le 20:00”; questa può essere modellata nel seguente modo

- walk takes on about 1h
- walk begins after 17:00 but before 20:00
- walk ends after 17:00 but before 20:00
- walk is before or meets dinner

dove l'ultimo vincolo è imposto per rendere palese al sistema che l'azione di camminare non può essere compiuta in contemporanea con la cena.

6.2.2 Traduzione vincoli dal linguaggio utente

I vincoli espressi, una volta elaborati dal modulo di interpretazione del linguaggio umano, risultano:

- “breakfast begins after about 8:00” →
 $\text{origin of time } \{(72, 96, +\infty, +\infty)\} \text{ breakfast_s}$
- “breakfast length is on about 30min” →
 $\text{breakfast_s } \{(4.5, 6, 6, 7.25)\} \text{ breakfast_f}$
- “lunch begins after about 13:00” →
 $\text{origin of time } \{(117, 156, +\infty, +\infty)\} \text{ lunch_s}$
- “lunch begins after about 4h but before about 6h compared to the end of breakfast” →
 $\text{lunch_s } \{(-78, -72, -48, -42)\} \text{ breakfast_f}$
- “lunch takes on about 1h” →
 $\text{lunch_s } \{(9, 12, 12, 15)\} \text{ lunch_f}$
- “dinner begins after about 19:30 but before about 21:00” →
 $\text{origin of time } \{(229.5, 234, 252, 256.5)\} \text{ dinner_s}$
- “dinner begins after about 5h but before about 6h compared to the end of lunch” →
 $\text{dinner_s } \{(-75, -72, -60, -57)\} \text{ lunch_f}$

- “dinner length is on about 1h” →

$$\text{dinner_s} \{(9, 12, 12, 15)\} \text{dinner_f}$$
- “herbal tea begins after about 12:00 but before about 20:00” →

$$\text{origin of time} \{(120, 144, 240, 264)\} \text{herbal tea_s}$$
- “herbal tea ends after about 12:00 but before about 20:00” →

$$\text{origin of time} \{(120, 144, 240, 264)\} \text{herbal tea_f}$$
- “herbal tea length is on about 20min” →

$$\text{herbal tea_s} \{(3, 4, 4, 5)\} \text{herbal tea_f}$$
- “aspirin intake begins after but before about 30min compared to the end of dinner” →

$$\text{aspirin intake_s} \{(-7.5, -6, 0, 0)\} \text{dinner_f}$$
- “aspirin intake length is on about 5min” →

$$\text{aspirin intake_s} \{(0.75, 1, 1, 1.25)\} \text{aspirin intake_f}$$
- “walk takes on about 1h” →

$$\text{walk_s} \{(9, 12, 12, 15)\} \text{walk_f}$$
- “walk begins after 17:00 but before 20:00” →

$$\text{origin of time} \{[204, 204, 240, 240]\} \text{walk_s}$$
- “walk ends after 17:00 but before 20:00” →

$$\text{origin of time} \{[204, 204, 240, 240]\} \text{walk_f}$$
- “walk is before or meets dinner” →

$$\text{walk} \{b, m\} \text{dinner}$$

Una rappresentazione del programma che l’utente deve seguire è riportata in figura 6.2.

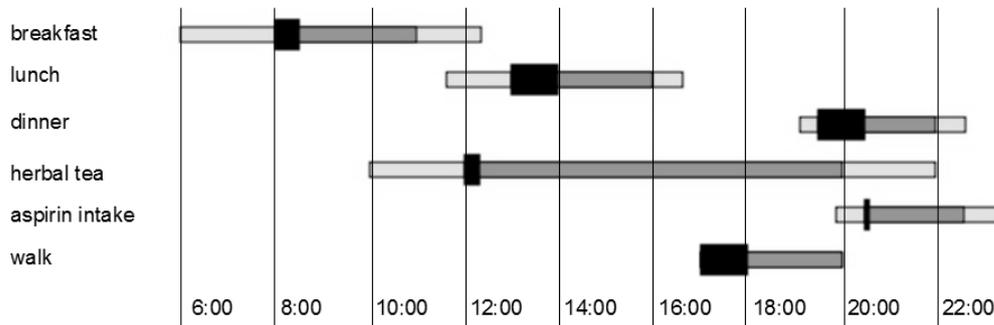


Figura 6.2: Esempio di programma per soggetto anziano

6.2.3 File dati

Di seguito viene riportato un estratto del file in cui sono salvate le informazioni sulla rete appena descritta. Per non appesantire la lettura, vengono riportate solo le parti relative all'attività della colazione.

```
<?xml version="1.0" encoding="UTF-8" ?>
<fcnetwork descr="schedule"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ftcn.xsd">
  <origin>00:00:00 </origin>
  <quantum>00:05:00 </quantum>
  <uncertainty>0.25</uncertainty>
  <vertices count="19">
    <vertex type="point" id="0" descr="origin of time">
    </vertex>
    <vertex type="interval" id="1" descr="breakfast">
    </vertex>[...]
    <vertex type="point" id="7" descr="breakfast_s">
    </vertex>[...]
    <vertex type="point" id="13" descr="breakfast_f">
    </vertex>[...]
  </vertices>
  <constraints count="35">
    <pifconstr vertex1="0" vertex2="1">
      <b>1</b>
      <s>1</s>
    </pifconstr>
```

```

[...]  

<mfconstrs count="1" vertex1="0" vertex2="7">  

  <mfconstr a="72" b="96" c="1" d="1"  

    left="opened" right="unbounded">  

    <alpha>1</alpha>  

  </mfconstr>  

</mfconstrs>  

[...]  

<ipfconstr vertex1="1" vertex2="7">  

  <si>1</si>  

</ipfconstr>  

<ipfconstr vertex1="1" vertex2="13">  

  <fi>1</fi>  

</ipfconstr>  

[...]  

<mfconstrs count="1" vertex1="7" vertex2="13">  

  <mfconstr a="4.5" b="6" c="6" d="7.5"  

    left="opened" right="opened">  

    <alpha>1</alpha>  

  </mfconstr>  

</mfconstrs>  

[...]  

</constraints>  

</fcnetwork>

```

6.3 Simulazione del comportamento dell'utente

Viene riportata una delle possibili esecuzioni del programma dell'utente fin qui elaborate allo scopo di verificare il comportamento del controllore di fronte ai casi in cui alcuni dei vincoli non vengono rispettati.

- 9:05 - 9:10 l'utente inizia a fare colazione;
- 9:35 - 9:40 completata la colazione, visto che tale azione rispetta tutti i vincoli imposti il controllore non produce nessuna segnalazione;
- 13:15 - 13:20 il soggetto inizia il pranzo, ma essendo la distanza fra colazione e pranzo inferiore alle quattro ore come prescritto, allora il controllore invia il messaggio (a priorità media) "Times between lunch and breakfast could

not match with the prescription! Next time you should have more time between them.”

- 14:15 - 14:20 l'utente conclude il proprio pranzo in perfetto orario, quindi non sono necessari ulteriori messaggi;
- 16:00 - 16:05 l'anziano comincia a prendere il suo infuso d'erbe;
- 16:15 - 16:20 l'azione di prendere l'infuso alle erbe si conclude con 5 minuti di anticipo rispetto al programma stabilito, per questa ragione il sistema suggerisce “Next time you should use more time for herbal tea”. La segnalazione ha priorità massima perché rende il vincolo inconsistente essendo il comportamento desiderato modellato come $\{ (3, 4, 4, 5) \}$ mentre il valore corrente è $\{ [3, 3, 3, 3] \}$;
- 17:40 - 17:45 l'utente esce per la sua camminata;
- 17:50 - 17:55 il soggetto non ha ancora concluso la sua passeggiata e il sistema lo informa (con messaggio a priorità media) con “You should use less time for walk”;
- 19:00 - 19:05 il messaggio “You should use less time for walk” viene iterato a priorità massima perché l'utente non ha ancora completato tale attività;
- 19:15 - 19:20 l'anziano rientra a casa e completa l'attività della passeggiata;
- 20:30 - 20:35 al soggetto viene notificato che il tempo consigliato fra il pranzo e la cena sta per concludersi tramite messaggio a priorità media;
- 20:40 - 20:45 il messaggio “Times between dinner and lunch could not match with the prescription! You should have less time between them.” viene iterato ma a priorità massima;
- 20:45 - 20:50 il soggetto inizia a cenare;
- 21:45 - 21:50 l'anziano termina di cenare entro il tempo stabilito;
- 22:00 - 22:05 l'utente inizia a prendere la pastiglia di aspirina;
- 22:05 - 22:10 il soggetto termina anche la sua ultima attività giornaliera secondo i vincoli proposti.

Tramite questo esempio è possibile osservare come il sistema possa suggerire all'utente il comportamento desiderato da adottare, come nei casi della

passaggiata prolungata o del ritardo nella consumazione della cena, o suggerisce come modificare in futuro tale comportamento perché risponda ai vincoli imposti, come nei casi del pranzo troppo anticipato rispetto alla colazione o alla durata troppo breve per consumare l'infuso di erbe.

In figura 6.3 viene riportato lo schema delle attività simulate in questo esempio, così come indicato nel simulatore.

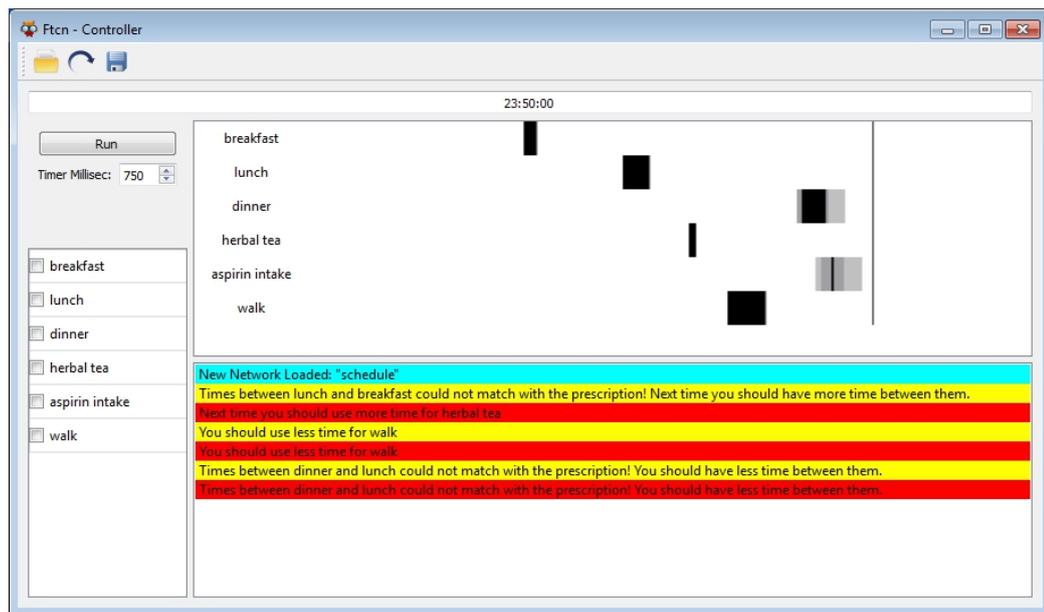


Figura 6.3: Esempio di simulazione delle attività di un soggetto anziano

Capitolo 7

Conclusioni

Dopo aver analizzato la tematica dell'assistenza agli anziani, e considerato le necessità e le problematiche che sono state rilevate in tale filone di studi, è emersa la possibilità di utilizzare il sistema di ragionamento temporale fuzzy per indirizzare il comportamento di quegli anziani che presentano ancora un buon grado di indipendenza.

In particolare la potenza espressiva messa a disposizione dal sistema permette all'anziano, o a quanti se ne prendono cura, di inserire attraverso semplici comandi dei parametri che corrispondano alle necessità che l'applicazione è chiamata a supervisionare. Il sistema è, inoltre, sufficientemente complesso da poter generare una risposta al comportamento dell'utente che rispecchi i gradi di urgenza impliciti nelle varie situazioni.

Lo stesso stato dell'arte del campo delle tecnologie assistenziali negli ultimi anni è migliorato, e questo ci permette di ipotizzare un possibile utilizzo pratico del lavoro fin qui svolto nel medio periodo; tale affermazione è motivata dalla necessità di consolidare le tecnologie sui cui risultati si deve appoggiare un sistema di supervisione fuzzy.

È presente, allo stesso tempo, un buon margine di miglioramento del sistema fin qui descritto al fine di poter gestire false letture dei sensori e permettere la ripetizione di comportamenti più volte all'interno della stessa giornata, soprattutto se compiute in risposta a suggerimenti sollevati dal sistema stesso.

Appendice A

Tabelle delle relazioni qualitative

Di seguito sono riportate le tabelle per eseguire l'operazione di composizione fra relazioni qualitative.

	II	IP	PI	PP
II	T_{IA}	T_1	\emptyset	\emptyset
IP	\emptyset	\emptyset	T_2	T_3
PI	T_4	T_5	\emptyset	\emptyset
PP	\emptyset	\emptyset	T_6	T_{PA}

Tabella A.1: Tabella complessiva per le composizioni

T_{PA}	b	eq	a
b	b	b	?
eq	b	eq	a
a	?	a	a

Tabella A.2: Tabella per l'operazione di composizione $PP \circ PP$

T_6	b	a	d	s	f
b	b	?	b s d	b	b s d
eq	b	a	d	s	f
a	?	a	d f a	d f a	a

Tabella A.3: Tabella per l'operazione di composizione per $PP \circ PI$

T_5	bi	ai	di	si	fi
b	?	b	b	b	b
a	a	?	a	a	a
d	a	b	?	a	b
s	a	b	b	eq	b
f	a	b	a	a	?

Tabella A.4: Tabella per l'operazione di composizione $PI \circ IP$

T_4	b	a	d	di	o	oi	m	mi	s	si	f	fi	eq
b	b	?	b s d	b	b	b d s	b	b d s	b	b	b d s	b	b
a	?	a	a d f	a	a d f	a	a d f	a	a d f	a	a	a	a
d	b	a	d	a	d	a	s	a	d	a	f	f	f
s	b	a	d	b	b	d	b	f	s	s	d	b	s
f	b	a	d	a	d	a	s	a	d	a	f	f	f

Tabella A.5: Tabella per l'operazione di composizione per $PI \circ II$

T_3	b	eq	a
bi	?	bi	bi
ai	ai	ai	?
di	di fi ai	di	di si bi
si	di fi ai	si	bi
fi	ai	fi	bi si di

Tabella A.6: Tabella per l'operazione di composizione $IP \circ PP$

T_2	b	a	d	s	f
bi	?	a	a mi oi d f	a mi oi d f	a
ai	b	?	b m o d s	b	b m o d f
di	b m o di fi	a mi oi si di	o oi eq s d f si di fi	o di fi	oi di si
si	b m o di fi	a	oi d f	s si eq	mi
fi	b	a mi oi si di	o s d	m	f fi eq

Tabella A.7: Tabella per l'operazione di composizione per $IP \circ PI$

T_1	b	a	d	s	f
b	b	?	b	b	b
a	?	a	a	a	a
d	b s d	d f a	d	d	d
di	b	a	?	b	a
o	b	d f a	b s d	b	d
oi	b s d	a	a	f	a
m	b	d f a	b	b	s
mi	b s d	a	a	f	a
s	b	d f a	d	s	d
si	b	d f a	d	s	d
f	b s d	a	d	d	f
fi	b	a	b d s	b	f
eq	b	a	d	s	f

Tabella A.8: Tabella per l'operazione di composizione $II \circ IP$

T_{IA}	b	a	d	di	o	oi	m	mi	s	si	f	fi	eq
b	b	?	b o m d s	b	b	b o m d s	b	b o m d s	b	b	b o m d s	b	b
a	?	a	a oi mi d f	a	a oi mi d f	a	a oi mi d f	a	a oi mi d f	a	a	a	a
d	b	a	d	?	b o m d s	a oi mi d f	b	a	d	a oi mi d f	d	b o m d s	d
di	b o m di fi	a oi di mi si	o oi s si d di f fi eq	di	o di fi	oi di si	o di fi	oi di si	o di fi	di	di si oi	di	di
o	b	a oi di mi si	o d s	b o m di fi	b o m	o oi s si d di f fi eq	b	oi di si	o	di fi o	d s o	b o m	o
oi	b o m di fi	a	oi d f	a oi mi di si	o oi s si d di f fi eq	a oi mi	o di fi	a	oi d f	oi a mi	oi	oi di si	oi
m	b	a oi mi di si	o d s	b	b	o d s	b	f fi eq	m	m	o d s	b	m
mi	b o m di fi	a	oi d f	a	oi d f	a	s si eq	a	d f oi	a	mi	mi	mi
s	b	a	d	b o m di fi	b o m	oi d f	b	mi	s	s si eq	d	b m o	s
si	b o m di fi	a	oi d f	di	o di fi	oi	o di fi	mi	s si eq	si	oi	di	si
f	b	a	d	a oi mi di si	o d s	a oi mi	m	a	d	a oi mi	f	f fi eq	f
fi	b	a oi mi di si	o d s	di	o	oi di si	m	si oi di	o	di	f fi eq	fi	fi
eq	b	a	d	di	o	oi	m	mi	s	si	f	fi	eq

Tabella A.9: Tabella per l'operazione di composizione per $II \circ II$

Appendice B

Activity of Daily Living

Metodo Katz e Akporn

Activities Points(1 or 0)	Independence (1 Point) No supervision, direction or personal assistance.	Dependence (0 Points) With supervision, direction, personal assistance or total care.
Bathing	Bathes self completely or needs help in bathing only a single part of the body, such as the back, genital area or disabled extremity.	Need help with bathing more than one part of the body, getting in or out of the tub or the shower. Requires total bathing.
Dressing	Get clothes from closets and drawers and puts on clothes and outer garments complete with fasteners. May have help tying shoes.	Needs help with dressing self or needs to be completely dressed.
Transferring	Moves in and out of bed or chair unassisted. Mechanical transfer aids are acceptable.	Needs help in moving from bed to chair or requires a complete transfer.
Toileting	Goes to toilet, gets on and off, arranges clothes, cleans genitals area without help.	Needs help transferring to the toilet, cleaning self or uses bedpan or commode.
Continence	Exercises complete self control over urination and defecation.	Is partially or totally incontinent of bowel or bladder.
Feeding	Gets food from plate into mouth without help. Preparation of food may be done by another person.	Needs partial or total help with feeding or requires parenteral feeding

Metodo Lawton e Brody

HouseKeeping		Mode of Transportation	
1. Maintains house alone or with occasional assistance.	1	1. Travels independently on public transportation or drives own car.	1
2. Performs light daily tasks such as dish washing, bed making.	1	2. Arranges own travel via taxi, but does not otherwise use public transportation.	1
3. Performs light daily tasks but cannot maintain acceptable level or cleanliness.	1	3. Travels on public transportation when accompanied by another.	1
4. Needs help with all home maintenance tasks.	1	4. Travel limited to taxi or automobile with assistance of another.	0
5. Does not participate in any housekeeping tasks.	0	5. Does not travel at all.	0
Telephone		Shopping	
1. Operates telephone on own initiative-looks up and dials numbers, etc.	1	1. Takes care of all shopping needs independently.	1
2. Dials a few well-known numbers.	1	2. Shops independently for small purchases.	0
3. Answers telephone but does not dial.	1	3. Needs to be accompanied on any shopping trip.	0
4. Does not use telephone at all.	0	4. Completely unable to shop.	0
Food Preparation		Ability to handle finances	
1. Plans, prepares and serves adequate meals independently.	1	1. Manages financial matters independently, collects and keeps track of income.	1
2. Prepares adequate meals if supplied with ingredients.	0	2. Manages day-to-day purchases, but needs help with banking, major purchases, etc.	1
3. Heats, serves and prepares meals, or prepares meals, or prepares meals but does not maintain adequate diet.	1	3. Incapable of handling money.	0
4. Needs to have meals prepared and served.	0		
Medications		Laundry	
1. Us responsible for taking medication in correct dosages at correct time.	1	1. Does personal laundry completely.	1
2. Takes responsibility if medication is prepared in advance in separate dosage.	0	2. Launders small items.	1
3. Is not capable of dispensing own medication.	0	3. All laundry must be done by others.	0

Bibliografia

- [Allen, 1983] Allen, J. 1983, *Maintaining Knowledge about temporal intervals*, Communications of the ACM, vol. 26, no. 11, pp. 832-843.
- [Ammouri, Bilodeau, 2008] Ammouri, S. & Bilodeau, G.A. 2008, *Face and hands detection and tracking applied to the monitoring of medication intake*, Proceedings of the 5th Canadian Conference on Computer and Robot Vision, CRV 2008, pp. 147.
- [Badaloni et al, 2003] S. Badaloni, M. Falda & M. Giacomini. 2003, *Integrating qualitative and quantitative constraints in fuzzy temporal networks*. In F.D. Anger et al (Eds.), Proc. of the International Workshop on Spatial and Temporal Reasoning, IJCAI03, pp. 105-112, Acapulco, Mexico.
- [Badaloni et al, 2004a] S. Badaloni, M. Falda & M. Giacomini. 2004, *Integrating quantitative and qualitative fuzzy temporal constraints*. In AI Communications, Special issue on Spatial and Temporal Reasoning, vol. 17 no. 4, pp. 187-200, IOS Press.
- [Badaloni et al, 2004b] S. Badaloni, M. Falda & M. Giacomini. 2004, *An hybrid fuzzy temporal constraint approach: a case study*. Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2004, pp. 1585-1592, Perugia, Italy.
- [Badaloni et al, 2007] S. Badaloni, M. Falda & M. Giacomini. 2007, *Solving temporal over-constrained problems using fuzzy techniques*. Journal of Intelligent and fuzzy systems, vol. 18, no. 3, pp. 255-265, IOS Press, 2007.
- [Badaloni, Falda, 2010] S. Badaloni, M. Falda. 2010, *Temporal-based medical diagnoses using a Fuzzy Temporal Constraint System*. Journal of Intelligent manufacturing, vol. 21, pp. 145-153, 2010.

- [Bahadori et al, 2005] Bahadori, S., Cesta, A., Iocchi, L., Leone, R., Nardi, D., Pecora, F., Rasconi, R. & Scozzafava, L. 2005, *Towards Ambient Intelligence for the Domestic Care of the Elderly*. In P. Remagnino, G.L. Foresti, and T.Ellis (Eds), "Ambient Intelligence: A Novel Paradigm", pp. 15-38, Springer.
- [Berenguer et al, 2008] Berenguer, M., Giordani, M., Giraud-By, F. & Noury, N. 2008, *Automatic detection of activities of daily living from detecting and classifying electrical events on the residential power line*, 2008 10th IEEE Intl. Conf. on e-Health Networking, Applications and Service, HEALTHCOM 2008, pp. 29.
- [Bilodeau, Ammouri, 2009] Bilodeau, G.A. & Ammouri, S. 2009, *Monitoring of Medication Intake Using a Camera System*, Journal of medical systems, , pp. 1-13.
- [Bourke et al, 2007] Bourke, A.K., O'Brien, J.V. & Lyons, G.M. 2007, *Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm*, Gait and Posture, vol. 26, no. 2, pp. 194-199.
- [Bourke, Lyons, 2008] Bourke, A.K. & Lyons, G.M. 2008, *A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor*, Medical Engineering and Physics, vol. 30, no. 1, pp. 84-90.
- [Cesta et al, 2010] Cesta, A., Coradeschi, S., Cortellessa, G., Gonzalez, J., Tiberio, L. & Von Rump, S. 2010, *Enabling Social Interaction Through Embodiment in ExCITE*, in ForItAAL. Second Italian Forum on Ambient Assisted Living.
- [Cesta et al, 2011] Cesta, A., Cortellessa, G., Rasconi, R., Pecora, F., Scopelliti, M. & Tiberio, L. 2011, *Monitoring Older People with the RoboCare Domestic Environment: Interaction Synthesis and User Evaluation*. In Computational Intelligence, vol. 27, no. 1, pp. 60-82.
- [Chen et al, 2010] Chen, Y.T., Lin, Y.C. & Fang, W.H. 2010, *A hybrid human fall detection scheme*, Proceedings - International Conference on Image Processing, ICIP, pp. 3485.
- [Dai et al, 2010] Dai, J., Bai, X., Yang, Z., Shen, Z. & Xuan, D. 2010, *PerFallD: A pervasive fall detection system using mobile phones*, 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2010, pp. 292.

- [Dinh, Struck, 2009] Dinh, C. & Struck, M. 2010, *A new real-time fall detection approach using fuzzy logic and a neural network*, Proceedings of the 6th International Workshop on Wearable, Micro, and Nano Technologies for Personalized Health: Facing Future Healthcare Needs, pHealth 2009, pp. 57.
- [Fleury et al, 2009] Fleury, A., Vacher, M. & Noury, N. 2010, *SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results*, IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 2, pp. 274-283.
- [Fleury et al, 2010] Fleury, A., Noury, N. & Vacher, M. 2010, *Introducing knowledge in the process of supervised classification of activities of Daily Living in Health Smart Homes*, 12th IEEE International Conference on e-Health Networking, Application and Services, Healthcom 2010.
- [Foroughi et al, 2008] Foroughi, H., Rezvanian, A. & Pazirae, A. 2008, *Robust fall detection using human shape and multi-class support vector machine*, Proceedings - 6th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2008, pp. 413.
- [Giuliani et al, 2005] Giuliani, M.V., Scopelliti, M. & Fornara, F. 2005, *Elderly people at home: Technological help in everyday activities*, Proceedings - IEEE International Workshop on Robot and Human Interactive Communication, pp. 365.
- [Hirsch et al, 2000] Hirsch, T., Forlizzi, J., Hyder, E., Goetz, J., Stroback, J. & Kurtz, C. 2000, *The ELDer project: Social, emotional, and environmental factors in the design of eldercare technologies*, Proceedings of the Conference on Universal Usability, pp. 72.
- [Helal, Chen, 2009] Helal, S. & Chen, C. 2009, *The Gator Tech Smart House: Enabling technologies and lessons learned*, ICREATE '09 - International Convention on Rehabilitation Engineering and Assistive Technology.
- [Hodges et al, 2006] Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N. & Wood, K. 2006, *SenseCam: A Retrospective Memory Aid*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) vol. 4206, pp. 177-193.
- [Hoey et al, 2010] Hoey, J., Poupard, P., Bertoldi, A.v., Craig, T., Boutilier, C. & Mihailidis, A. 2010, *Automated handwashing assistance for persons*

with dementia using video and a partially observable Markov decision process, Computer Vision and Image Understanding, vol. 114, no. 5, pp. 503-519.

[Huang et al, 2009] Huang, C.N., Chiang, C.Y., Chang, J.S., Chou, Y.C., Hong, Y.X., Hsu, S.J., Chu, W.C. & Chan, C.T. 2009, *Location-aware fall detection system for medical care quality improvement*, 3rd International Conference on Multimedia and Ubiquitous Engineering, MUE 2009, pp. 477.

[Klenk et al, 2011] Klenk, J., Becker, C., Lieken, F., Nicolai, S., Maetzler, W., Alt, W., Zijlstra, W., Hausdorff, J.M., van Lummel, R.C., Chiari, L. & Lindemann, U. 2011, *Comparison of acceleration signals of simulated and real-world backward falls*, Medical Engineering and Physics, vol. 33, no. 3, pp. 368-373.

[Li et al, 2009] Li, Q., Stankovic, J.A., Hanson, M.A., Barth, A.T., Lach, J. & Zhou, G. 2009, *Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information*, Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009, pp. 138.

[Liu et al, 2010] Liu, C.L., Lee, C.H. & Lin, P.M. 2010, *A fall detection system using k-nearest neighbor classifier*, Expert Systems with Applications, vol. 37, no. 10, pp. 7174-7181.

[Lord, Colvin, 1991] C.J. Lord, C.J. & Colvin, D.P. 1991, *Falls in the elderly: Detection and assessment*, Proceedings of the Annual Conference on Engineering in Medicine and Biology, pp. 1938.

[Mann et al, 1999] Mann, W.C., Ottenbacher, K.J., Fraas, L., Tomita, M. & Granger, C.V. 1999, *Effectiveness of assistive technology and environmental interventions in maintaining independence and reducing home care costs for the frail elderly a randomized controlled trial*, Archives of Family Medicine, vol. 8, no. 3, pp. 210-217.

[Meiri, 1995] Meiri, I. 1996, *Combining qualitative and quantitative constraints in temporal reasoning*, Artificial Intelligence, vol. 87, no. 1-2, pp. 343-385.

[Miskelly, 2001] Miskelly, F.G. 2001, *Assistive technology in elderly care*, Age and Ageing, vol. 30, no. 6, pp. 455-458.

- [Niazmand et al, 2010] Niazmand, K., Jehle, C., D'Angelo, L.T. & Lueth, T.C. 2010, *A new washable low-cost garment for everyday fall detection*, 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10, pp. 6377.
- [Noury et al, 2007] Noury, N., Fleury, A., Rumeau, P., Bourke, A.K., Laignin, G.O., Rialle, V. & Lundy, J.E. 2007, *Fall detection - Principles and methods*, Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, pp. 1663.
- [Noury et al, 2008] Noury, N., Galay, A., Pasquier, J. & Ballussaud, M. 2008, *Preliminary investigation into the use of autonomous fall detectors*, Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - Personalized Healthcare through Technology, pp. 2828.
- [Noury et al, 2010] Noury, N., Quach, K.A., Berenguer, M., Teyssier, H., Bouzid, M.J., Goldstein, L. & Giordani, M. 2010, *Ubiquitous but non invasive evaluation of the activity of a person from a unique index built on the electrical activities on the residential power line*, 12th IEEE International Conference on e-Health Networking, Application and Services, Healthcom 2010.
- [Noury et al, 2011] Noury, N., Berenguer, M., Teyssier, H., Bouzid, M.J. & Giordani, M. 2011, *Building an index of activity of inhabitants from their activity on the residential electrical power line*, IEEE Transactions on Information Technology in Biomedicine, vol. 15, no. 5, pp. 758-766.
- [O'Brien, Mac Ruairi, 2009] O'Brien, A. & Ruairi, R.M. 2009, *Survey of assistive technology devices and applications for aging in place*, 2nd International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies, and Services - CENTRIC 2009, pp. 7.
- [Philipose et al, 2004] Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Fox, D., Kautz, H. & Hahnel, D. 2004, *Inferring activities from interactions with objects*, IEEE Pervasive Computing, vol. 3, no. 4, pp. 50-57.
- [Pineau et al, 2003] Pineau, J., Montemerlo, M., Pollack, M., Roy, N. & Thrun, S. 2003, *Towards robotic assistants in nursing homes: Challenges and results*, Robotics and Autonomous Systems, vol. 42, no. 3-4, pp. 271-281.

- [Pollack, 2005] Pollack, M.E. 2005, *Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment*, AI Magazine, vol. 26, no. 2, pp. 9-24.
- [Rougier et al, 2006] Rougier, C., Meunier, J., St-Arnaud, A. & Rousseau, J. 2006, *Monocular 3D head tracking to detect falls of elderly people*, Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, pp. 6384.
- [Rougier et al, 2011] Rougier, C., Meunier, J., St-Arnaud, A. & Rousseau, J. 2011, *Robust video surveillance for fall detection based on human shape deformation*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 5, pp. 611-622.
- [Shieh, Huang, 2009] Shieh, W.Y. & Huang, J.C. 2009, *Speedup the multi-camera video-surveillance system for elder falling detection*, Proceedings - 2009 International Conference on Embedded Software and Systems, ICESS 2009, pp. 350.
- [Stanford, 2002] Stanford, V. 2002, *Using pervasive computing to deliver elder care*, IEEE Distributed Systems Online, vol. 3, no. 3.
- [Snoek et al, 2007] Snoek, J., Hoey, J., Stewart, L. & Zemel, R.S. 2006, *Automated detection of unusual events on stairs*, Third Canadian Conference on Computer and Robot Vision, CRV 2006.
- [Tzeng et al, 2010] Tzeng, H.W., Chen, M.Y. & Chen, J.Y. 2010, *Design of fall detection system with floor pressure and infrared image*, 2010 International Conference on System Science and Engineering, ICSSE 2010, pp. 131.
- [Vilain et al, 1989] Vilain, M., Kautz, H. & Van Beek, P. 1989, *Constraint propagation algorithms for temporal reasoning: a revised report*, in Reading in Qualitative Reasoning about Physical Systems, J. de Kleer and D.S. Weld, eds, San Mateo, CA.
- [Vinjumur et al,] Vinjumur, J.K., Becker, E., Ferdous, S., Galatas, G. & Makedon, F. 2010, *Web Based medicine intake tracking application*, ACM International Conference Proceeding Series.
- [Wang et al, 2008] Wang, C.C., Chiang, C.Y., Lin, P.Y., Chou, Y.C., Kuo, I.T., Huang, C.N. & Chan, C.T. 2008, *Development of a fall detecting system for the elderly residents*, 2nd International Conference on Bioinformatics and Biomedical Engineering, iCBBE 2008, pp. 1359.

-
- [Yanco, 2001] Yanco, H.A. 2001, *Development and Testing of a Robotic Wheelchair System for Outdoor Navigation*, in Proceedings of the 2001 Conference of the Rehabilitation Engineering and Assistive Technology Society of North America.
- [Yu, 2008] Yu, X. 2008, *Approaches and principles of fall detection for elderly and patient*, 2008 10th IEEE Intl. Conf. on e-Health Networking, Applications and Service, HEALTHCOM 2008, pp. 42.
- [Zhang et al, 2006] Zhang. T., Wang, J., Liu. P. & Hou, J. 2006. *Fall detection by embedding an accelerometer in cellphone and using KFD algorithm* International Journal of Computer Science and Network Security, vol. 6, no.10, pp. 277.