



ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

SECONDA FACOLTÀ DI INGEGNERIA

SEDE DI CESENA

Corso di Laurea in Ingegneria Informatica
(codice 0206)

ELABORATO FINALE

**INTEROPERABILITÀ
TRA FRAMMENTI E.H.R. E
COORDINAMENTO SEMANTICO**

Candidato:
Simone Norcini

Relatore:
Chiar.mo Prof. Ing. ANDREA OMICINI

Anno Accademico 2011/2012
Sessione: I

Indice

1. Introduzione	pag. 4
1.1 Motivazioni per la ricerca	pag. 4
1.2 Frammenti E.H.R.	pag. 4
1.3 Interoperabilità nei modelli data-centrici	pag. 5
2. Evoluzione dei modelli data-centrici	pag. 7
2.1 Linda e le sue implementazioni	pag. 7
2.2 Omogeneità tra informazione e controllo	pag. 9
2.3 Verso la programmabilità del comportamento	pag. 10
2.4 Approcci object-oriented	pag. 11
2.5 Limiti degli spazi di tuple	pag. 12
2.6 Centri di tuple	pag. 14

3. Centri di tuple semantici	pag. 17
3.1 Esigenza dell'estensione semantica	pag. 17
3.2 Semantica per il Web	pag. 18
3.3 Semantica e sistemi ad informazione intensiva	pag. 18
3.4 Requisiti e componenti per l'estensione semantica	pag. 19
4. Semantic Health Coordination	pag. 23
4.1 Interoperabilità	pag. 23
4.2 Standard pre-semantica	pag. 24
4.3 Standard e iniziative post-semantica	pag. 24
4.4 Obiettivi di SemHealthCoord	pag. 26
5. Conclusione	pag. 27
6. Bibliografia	pag. 28

Capitolo 1

Introduzione

1.1 Motivazioni per la ricerca

L'integrazione delle nuove tecnologie all'interno di un sistema sanitario pone delle sfide che la ricerca sta affrontando da molteplici punti di vista. Per comprendere la necessità dello sviluppo di sistemi e-Health, dove l'informatizzazione è pervasiva, basti pensare alla triplice necessità *i)* di ridurre la possibilità di errore umano nella trasmissione dell'informazione in un sistema critico dove i fallimenti possono provocare incidenti con perdita di vite umane, *ii)* alla riduzione dei tempi /semplificazione delle modalità di accesso all'informazione, infatti anche i tempi sono un fattore critico, e infine, *iii)* alle pressioni attuali per ridurre i costi del servizio [1].

In un sistema non informatizzato quindi capita che [4][5][6] : *i)* insorgano errori nell'interpretazione ortografica delle cartelle cliniche oppure che alcune cartelle vadano smarrite; *ii)* le informazioni mediche vengono trasmesse da operatori umani per via telefonica e facendo reperire fisicamente dati sanitari cartacei da altri luoghi.

1.2 Frammenti E.H.R.

Alla base di un sistema e-Health vi è l'informazione sanitaria ovvero la conoscenza riportata sulle cartelle cliniche che deve però essere immagazzinata in formato digitale. Un frammento E.H.R. (Electronic Health Record) [11] ha perciò il compito di imbrigliare una parte di conoscenza medica relativa ad un paziente e costituisce perciò un "mattone" in un sistema knowledge-intensive.

Si noti innanzitutto come un frammento E.H.R. non corrisponda affatto al concetto di cartella clinica. Si pensi invece quest'ultima come ad una vista o prospettiva che, in un sistema e-Health, può essere richiamata in maniera personalizzata volta per volta da uno specialista e costruita con un criterio di aggregazione di frammenti. Per via della natura intrinseca di un sistema sanitario i frammenti E.H.R., così come i dati clinici, vengono collezionati da diverse istituzioni cooperanti tra loro ma delocalizzate spazialmente sul territorio.

La sfida informatica consiste nell'individuare un opportuno modello data-centrico che non sia solo in grado di conservare i frammenti in maniera coerente ma che fornisca anche la possibilità, in seguito ad un'attività di ragionamento sui dati, di generare nuova informazione come ad esempio sondaggi, statistiche di sistema e ricerche.

1.3 Interoperabilità nei modelli data-centrici

Interoperabilità, nel nostro caso di studio, non indica solamente le modalità in cui alcuni servizi possono essere rilasciati all'interno di un singolo sistema sanitario ma, in un senso più ampio, ha a che vedere con le migliori pratiche per la rappresentazione dell'informazione al fine di [1] rendere i dati sanitari accessibili anche al di fuori dei confini nazionali.

Per affrontare la questione dell'interoperabilità è quindi fondamentale la progettazione di un middleware, ovvero uno strato software mediatore, in grado non solo di coordinare i frammenti E.H.R. stessi ma anche di gestire le problematiche relative alla distribuzione, l'apertura [3][10] e la sicurezza del sistema in maniera coerente. La questione della distribuzione si sviluppa in tre dimensioni ortogonali: *i*) distribuzione del controllo, ovvero il modo in cui gli attori indipendenti che caratterizzano il sistema agiscono sul sistema stesso, *ii*) distribuzione spaziale, entità e dispositivi sono fisicamente distribuiti e connessi in una rete locale (LAN) o globale (WAN, Internet) *iii*) distribuzione temporale, non è richiesto che i componenti del sistema siano attivi e disponibili all'interazione allo stesso momento.

Sul fronte dell'apertura invece, il middleware deve fronteggiare eterogeneità e dinamismo dei componenti del sistema. Infatti in un sistema computazionale complesso vi sono componenti dissimili in termini di architettura e tecnologia che vengono aggiornati, aggiunti o rimossi in maniera dinamica mentre il sistema rimane in esecuzione. Nonostante siano stati compiuti molti sforzi per far fronte a questi requisiti le soluzioni proposte dalla letteratura sembrano non essere sufficientemente adatte a risolvere contemporaneamente tutte le problematiche. In questa tesi si cercherà di fornire un punto di vista sulle tecniche semantiche intese come estensione “naturale” delle architetture data-centriche.

Si cercherà di fare ciò mantenendo una prospettiva evolutivista che ripercorra i modelli architetturali storici basati su Linda. Ne verranno quindi sottolineati i vincoli che hanno reso necessario un graduale cambiamento di paradigma fino ad arrivare alle nuove possibilità che un approccio semantico può portare grazie all'impiego delle ontologie in termini di capacità espressiva e comunicazione generativa. Infine verrà approfondita la tematica dei servizi semantici in un ambiente e-Health dove è fondamentale poter intervenire con correzioni a runtime, senza la possibilità di fermare il sistema.

Capitolo 2

Evoluzione dei modelli data-centrici

2.1 Linda e le sue implementazioni

I principali modelli di coordinamento del tipo data-centrico si sono evoluti facendo riferimento al concetto di spazio di dati condiviso [12] ovvero una struttura comune che permette al suo contenuto di essere raggiunto mediante opportune strategie di indirizzamento. Tutti gli attori coinvolti nella computazione possono comunicare tra di loro solo indirettamente attraverso questo unico tramite. Gli attori possono così caricare e scaricare informazioni sia rimuovendo dati dallo spazio condiviso che, semplicemente, rilevandone una copia. Poiché la comunicazione tra i processi è resa possibile solo mediante lo spazio di dati condiviso e poiché la sopravvivenza dei dati non è legata alla sorte dei processi, è possibile, con l'utilizzo di questa astrazione, ottenere il disaccoppiamento spazio-temporale all'interno del sistema. Infatti, alcuni processi possono creare dati o addirittura terminare la propria esecuzione mentre altri processi li raccolgono in maniera del tutto asincrona. Non è richiesto che il produttore sia consapevole dell'esistenza del consumatore né dello stato dell'informazione che ha creato. Un processo potrebbe persino ricoprire sia il ruolo di produttore che quello di consumatore.

Nella famiglia dei modelli data-centrici le varie implementazioni differiscono le une dalle altre relativamente a parametri quali il tipo di struttura dati, le tecniche di indirizzamento o i meccanismi di accesso ai dati. Storicamente il primo archetipico appartenente alla famiglia dei linguaggi di coordinamento fu Linda, la quale fornisce una maniera semplice ed elegante di separare la computazione dalle questioni relative alla comunicazione. Inoltre Linda è basata sul paradigma di comunicazione generativa [13]. Infatti, se due processi sono interessati allo scambio di dati, allora il mittente crea un nuovo oggetto chiamato tupla e lo carica sullo spazio di dati condiviso, che nel modello Linda è chiamato spazio di tuple. Da qui il destinatario lo può ottenere. Dal momento che lo spazio di tuple è effettivamente uno spazio di dati

condiviso come definito per i modelli data-centrici, grazie a Linda è possibile disaccoppiare i processi sia spazialmente che temporalmente. Infatti i processi partecipanti non hanno la necessità di conoscere le reciproche identità né devono essere necessariamente creati allo stesso momento. Inoltre le tuple non sono necessariamente oggetti passivi, infatti possono anche contenere processi o tuple attive che, dopo il completamento della loro esecuzione, diventano tuple passive. Tuttavia Linda non è un linguaggio di coordinamento del tutto completo quanto piuttosto un piccolo insieme di semplici primitive di coordinamento:

- **out()**: l'esecuzione di questa primitiva provoca l'inserimento di una tupla nello spazio di tuple. Il processo che la invoca continua immediatamente.
- **in()**: quando questo comando è eseguito, se nello spazio di tuple esiste una tupla con il nome specificato, quella tupla viene scaricata e rimossa. Se nello spazio di tuple non viene trovata alcuna tupla corrispondente, la procedura in() si sospende in attesa dell'arrivo di una tupla conforme.
- **rd()**: questa procedura è identica alla in() tranne per il fatto che, quando una tupla corrispondente viene trovata, una volta consegnata al richiedente non viene però rimossa dallo spazio di tuple. Quindi anche la read() è una procedura potenzialmente bloccante.

Come visto, la primitiva out() è non bloccante, mentre le primitive rd() e in() sono operazioni bloccanti che sospenderanno l'esecuzione fino al momento in cui viene trovata nello spazio di tuple la tupla desiderata. Nel corso degli anni, nel tentativo di superare le limitazioni imposte dalle operazioni bloccanti, che di fatto precludono temporaneamente l'accesso allo spazio di tuple da parte di qualunque altro attore, sono state introdotte delle varianti alle primitive base. Infatti, rdp() e inp() sono varianti non bloccanti di rd() e in(), cioè invece di bloccare il sistema, quando la tupla desiderata non viene trovata, restituiscono il valore booleano "falso".

La comunicazione generativa [13] e l'accesso associativo rendono i modelli di coordinamento che adottano le tuple particolarmente interessanti per la progettazione di sistemi distribuiti, aperti e basati sulle informazioni [14]. In particolare la comunicazione generativa promuove il disaccoppiamento della comunicazione tra i componenti del sistema: i produttori di tuple e i consumatori possono interagire senza avere la reciproca conoscenza a priori, ottenendo così il disaccoppiamento spazio-temporale e il disaccoppiamento dei nomi degli attori, che generalmente occorrerebbero per istanziare la comunicazione. L'accesso associativo alle tuple, insieme alla sincronizzazione basata sulla disponibilità della tupla nello

spazio di tuple, promuove la coordinazione in uno scenario basato sulla conoscenza dove i componenti del sistema sono coordinati facendo riferimento al contenuto informativo delle tuple.

Riassumendo, le seguenti caratteristiche di Linda sono da considerarsi importanti per la progettazione di agenti negli scenari distribuiti [17]:

- disaccoppiamento dei processi interagenti sia dal punto di vista del riferimento che temporalmente. Il produttore di una tupla e il consumatore della stessa non hanno bisogno di conoscere i rispettivi indirizzi o di coesistere in maniera concorrente.
- Linda permette l'indirizzamento associativo. Ciò significa che i dati sono reperiti in base al tipo richiesto piuttosto che al dato specifico che si sta cercando.
- Linda supporta l'asincronia e la concorrenza in maniera intrinseca così come le eredita dall'astrazione dello spazio di tuple.
- Separa l'implementazione della coordinazione dalle caratteristiche della piattaforma ospitante o del linguaggio di programmazione. Linda è perciò indipendente dalle tecnologie sottostanti utilizzate.

Un'altra caratteristica che rende i modelli derivati da Linda particolarmente adatti agli scenari distribuiti è l'indipendenza del modello dalle tecnologie sottostanti. Infatti, dopo la formulazione originale di Linda sono state sviluppate e proposte dalla letteratura un certo numero di implementazioni, ognuna delle quali portava con sé una diversa visione delle modalità con cui si sarebbero dovuti evolvere i sistemi data-centrici.

2.2 Omogeneità tra informazione e controllo

L'esigenza di omogeneizzare la conoscenza contenuta in un sistema con le strutture di controllo del sistema stesso portò alla realizzazione di un'estensione diretta del modello Linda. Linda Bauhaus [26] si distingue dall'originale per la capacità di implementare spazi di tuple multipli sottoforma di svariati multiset (msets). In quest'implementazione non c'è una vera e propria differenziazione tra le singole

tuple e gli spazi di tuple. Un'altra caratteristica è l'assenza dei template, altrimenti utilizzati per specificare insiemi di caratteristiche da ricercare nelle tuple. Inoltre non vi è diversità tra tuple attive, che contengono processi, e tuple passive contenenti semplici informazioni o processi terminati. Invece di aggiungere, leggere o rimuovere tuple da un singolo mezzo condiviso piatto, le primitive di Linda Bauhaus consentono di aggiungere, leggere o rimuovere dei multiset da un altro multiset. Di conseguenza il criterio di confronto associativo dipendente dalla posizione nelle tuple di Linda è qui rimpiazzato da un insieme d'inclusione non ordinato. Per esempio consideriamo l'esistenza del multiset $\{a \{x Q\} \{\{z\}\} P\}$, dove gli elementi in lettere maiuscole denotano processi e gli elementi in lettere minuscole semplici tuple. Se il processo P esegue la primitiva "out $\{x>R\}$ " il multiset diventerà $\{a \{x Q R\} \{\{z\}\} P\}$. Inoltre, se il multiset ha la forma di $\{a \{x y\} P\}$ e viene eseguito l'insert "m:= in $\{x\}$ " allora m diviene assegnato alla struttura $\{x y\}$.

Il linguaggio di Linda Bauhaus introduce inoltre la nuova primitiva "move", del tutto inesistente nel modello base, la quale permette alle tuple di essere spostate linearmente tra i livelli di un multiset, concedendo così la possibilità di organizzare i dati in gerarchie.

2.3 Verso la programmabilità del comportamento

L'omogeneità raggiunta con Linda Bauhaus era stata ottenuta al prezzo della modifica del modello base e delle sue primitive. Comportava inoltre la perdita della chiara separazione tra contenuto informativo e comportamento del sistema. Un approccio differente in grado di mantenere quest'importante separazione è Linda Law-Governed [27], la quale non modifica le primitive base ma vi sovrappone un insieme di leggi alle quali devono aderire tutti i processi desiderosi di partecipare a qualche scambio di dati attraverso lo spazio di tuple. In questo modello esiste un controllore per ogni processo partecipante al sistema e ogni controllore dispone di una copia della legge imposta sullo spazio di dati condiviso. Il compito di un controllore è di intercettare ogni tentativo di comunicazione tra i processi che controlla ed il resto del mondo. Se il tentativo di comunicazione non è conforme alla

legge, viene impedito. Un sistema Linda Law-Governed è completamente determinato da una quintupla contenente i seguenti elementi :

- C: lo spazio di tuple.
- P: un insieme di processi.
- CS: un insieme di stati di controllo per i processi. Ve n'è uno per ogni processo pertanto la cardinalità di CS è identica a quella di P.
- L: la legge che governa il sistema.
- E: l'insieme dei controllori che fanno rispettare la legge.

Anche se la legge può essere teoricamente formulata in qualunque linguaggio, i progettisti di Linda Law-Governed hanno scelto un sottoinsieme ristretto del Prolog arricchito da altre primitive che vanno ad aggiungersi alle primitive base di Linda ma senza modificare quest'ultime. Ogni processo del sistema si trova sempre in un certo stato rappresentato dalla variabile globale CS, la quale può essere sempre richiamata per mezzo di un opportuno operatore. Formulando una legge adatta è dunque permesso ad un utente di arricchire le funzionalità del modello base di Linda. Attraverso una legge è anche possibile creare spazi di tuple multipli.

2.4 Approcci object-oriented

La trasposizione di Linda in un ambiente sviluppato intorno all'astrazione object-oriented sembrò essere fin dal principio una simbiosi particolarmente felice. Infatti, un oggetto può essere facilmente identificato con una tupla e il riscontro dell'esistenza di una tupla nello spazio di tuple è semplificato dai tipi Java. A partire dagli anni '90 si sono perciò susseguite una serie di implementazioni del modello Linda combinate con Java.

Sul fronte del coordinamento di oggetti mobili e del multithreading ricordiamo Jada [29], particolarmente adatto per sistemi aperti. Per Jada sono state definite opportune classi quali TupleServer e TupleClient, le quali consentono l'accesso remoto ad uno spazio di tuple e rendono possibile la comunicazione attraverso i socket. Un TupleClient, per connettersi, ha bisogno di conoscere l'host e l'id della porta alla quale è associato il socket di un TupleServer. Java fornisce i costrutti

appropriati per specificare queste informazioni. Jada può essere utilizzato sia come linguaggio di coordinamento a sé stante sia come linguaggio kernel per la progettazione di linguaggi di coordinamento più complessi per il World Wide Web.

Successivamente IBM ha proposto TSpaces [30], una soluzione che fornisce operatori avanzati e capacità di gestione dei database così da poter supportare applicazioni più complesse. È costituito da uno strato di gestione dei dati che fornisce caratteristiche simili a quelle offerte dai database relazionali. Questo strato utilizza indici per reperire in maniera efficiente dati e supporta l'utilizzo di query più espressive rispetto al meccanismo di confronto a template di Linda. Oltre all'esteso insieme di primitive di coordinamento, nuovi operatori possono essere aggiunti dinamicamente. Gli agenti possono aggiungere qualunque nuova primitiva allo spazio di tuple al fine di implementare qualunque transazione richiesta che coinvolga i dati immagazzinati. Tuttavia questo approccio non è generalmente adatto all'impiego in scenari aperti poiché per sfruttare le funzionalità aggizionali è richiesto uno stretto accoppiamento tra le entità interagenti. Infatti gli agenti devono sapere quali operazioni sono in grado di svolgere gli altri agenti partecipanti allo spazio di tuple.

La Sun ha invece proposto i JavaSpaces [15] che forniscono un middleware di comunicazione per Java. In questa versione object-oriented del modello Linda le tuple incapsulano oggetti Java. I JavaSpaces supportano la mobilità del codice, una memoria temporanea distribuita, il pattern publish-subscribe, le transazioni e il bilanciamento del carico. Più recentemente è stata introdotta un'estensione dei JavaSpaces chiamata GigaSpaces [16] per supportare l'architettura basata sugli spazi di tuple in un ambiente dove i client comunicano partecipando ad una Grid network.

2.5 Limiti degli spazi di tuple

Al tempo della sua introduzione, Linda era dunque considerata una metafora semplice ed elegante per il coordinamento dei processi poiché introduceva una distinzione chiara fra la progettazione dell'agente ed il suo coordinamento, semplificando l'ingegnerizzazione del sistema e affidandogli una forte connotazione data-centrica. Tuttavia, nella sua semplicità, Linda trascurava i seguenti aspetti [2]:

- capacità di rappresentabilità dell'informazione
- meccanismo di recupero dei dati
- supporto per le transazioni integrato
- disaccoppiamento del flusso di controllo dal lato client

Per via della semplicità del protocollo di interazione, la rappresentazione dell'informazione in forma di tuple è dunque l'origine di uno dei principali limiti dei modelli di coordinamento che sfruttano lo spazio di tuple. In quest'ambiente non è infatti possibile distinguere l'informazione che è realmente trasportata dalle tuple dalla rappresentazione dell'informazione stessa nello spazio di tuple. Perciò non vi è possibile introdurre alcun meccanismo che sia in grado di separare il modo in cui l'informazione è rappresentata dal modo in cui l'informazione è percepita dagli agenti. Ciò risulta chiaro se si prende in esame l'esempio di filosofi a tavola [18].

In uno spazio di tuple vi è la duplice possibilità di rappresentare le posate individualmente, come un'entità indipendente, oppure come coppie. Ricordando che un filosofo abbisogna di entrambe le posate per mangiare (quella alla sua destra e quella alla sua sinistra), il primo scenario, nonostante sia la scelta più naturale per la rappresentazione del dominio in un'ottica object-oriented, può condurre facilmente al deadlock, condizione in cui ogni filosofo si impossessa di una singola posata e, non riuscendo dunque ad ottenere l'altra, non porta a termine il processo e non rilascia pertanto la risorsa che utilizza. Il secondo scenario, nonostante sia potenzialmente in grado di risolvere il problema del deadlock, introduce tuttavia la questione della rappresentazione coerente del dominio, lì dove ogni oggetto "coppia di posate" è di fatto in sovrapposizione con le due coppie adiacenti.

Per quanto riguarda il mancato supporto alle transazioni da parte dello spazio di tuple, il comportamento, rappresentato dalle sue transizioni di stato in risposta al presentarsi di eventi di comunicazione, bisogna sottolineare il fatto che il comportamento è deciso una volta per tutte dal modello stesso e non può essere modificato per far fronte alle esigenze delle specifiche applicazioni. In aggiunta, ciò rende assolutamente necessario che, durante la fase di progettazione degli agenti che parteciperanno al sistema, si sia già consapevoli di tutte le esigenze che avranno i fruitori del sistema, ponendo un grave limite all'apertura del sistema ed alla sua evoluzione. Infatti, utilizzando un basilare modello a spazio di tuple, poiché il fardello del coordinamento è caricato sugli agenti, sarà necessario che ogni sviluppatore di software presente e futuro sia al corrente delle esatte specifiche di sistema poiché dovrà implementarle in ogni applicazione da sviluppare, rendendo estremamente

difficile l'estendibilità del sistema e del tutto impossibile la modifica delle caratteristiche di comportamento a runtime.

Nell'originale modello Linda [13] era solo possibile leggere, scrivere o consumare una singola tupla alla volta e l'accesso alle tuple era possibile solo sfruttando il meccanismo di confronto integrato nel mezzo di comunicazione condiviso (spazio di tuple). Nonostante alcune estensioni del modello base abbiano cercato di aumentare l'espressività del mezzo condiviso, nel tentativo di far fronte alle questioni relative al coordinamento che emergevano in un sistema multiagente riducendo così la complessità degli agenti, può anche accadere che si presentino questioni critiche irrisolvibili all'interno del modello Linda. Per esempio il problema delle letture multiple può essere affrontato solo introducendo una specifica primitiva di base.

Ad ogni modo le soluzioni proposte per le specifiche problematiche individuate non sono eleganti ed entrano in conflitto con l'ortogonalità tra coordinamento e computazione. Questi due aspetti risultano infatti interlacciati e non ben distinti all'interno dei coordinabili. Da un punto di vista olistico, le problematiche precedentemente individuate possono essere interpretate come la mancanza ereditaria dei meccanismi di controllo tipica dei modelli di coordinamento incentrati sui dati, la quale è particolarmente evidente quando si pongono come pietra di confronto i modelli incentrati sul controllo [12].

2.6 Centri di tuple

Le problematiche precedentemente esposte ed i limiti evidenziati, riassumibili nell'indeterminabilità del posizionamento del carico di coordinamento e computazione tra mezzo di coordinamento e coordinabili, possono essere superate mantenendo la rappresentazione dell'informazione chiaramente separata dal modo in cui l'informazione è percepita dagli agenti stessi all'interno del mezzo condiviso. Il raggiungimento di questo obiettivo permetterebbe infatti la realizzazione di protocolli di interazione tra agenti organizzati intorno ad una comune percezione dello spazio di comunicazione in maniera del tutto indipendente dalla sua reale

rappresentazione in termini di tuple. Si è giunti così alla nozione di centro di tuple [3], ovvero l'evoluzione di uno spazio di tuple esteso con la capacità di esprimere specifiche sul comportamento dello spazio di tuple stesso in risposta ad eventi di comunicazione. Le specifiche di comportamento di un centro di tuple sono espresse in un opportuno linguaggio di specifica delle reazioni tramite il quale è possibile associare l'arrivo nel centro di tuple di ogni possibile evento di comunicazione ad un insieme di attività computazionali chiamate reazioni.

Un linguaggio di specifica delle reazioni permette dunque la definizione di attività di computazione all'interno del centro di tuple, appunto le reazioni, e rende possibile associarne una ad ogni evento di comunicazione, sia in entrata che in uscita dal centro di tuple. Una reazione ha la possibilità di modificare lo stato corrente del centro di tuple aggiungendo e rimuovendo tuple. Inoltre, una reazione, possedendo il riferimento agli agenti ed alle operazioni che innescano la reazione stessa, può accedere a tutte le informazioni correlate all'evento, il quale è reso perciò completamente osservabile. Grazie a questa estensione, la semantica delle primitive di comunicazione di uno spazio di tuple non è più vincolata ad essere semplice come nel modello Linda ma può essere resa ben più complessa per far fronte alle esigenze delle specifiche applicazioni. Inoltre, grazie alla relazione così instaurata tra la rappresentazione dell'informazione e la sua percezione, gli agenti vengono scaricati dal fardello del coordinamento poiché le leggi di coordinamento vengono adesso incorporate nel mezzo di coordinamento stesso.

Nello specifico, un centro di tuple mantiene un'interfaccia standard ereditata dallo spazio di tuple, cosicché gli agenti che interloquiscono non notino la differenza, mentre allo stesso tempo è possibile arricchire il comportamento dello spazio di tuple in termini di transizioni di stato eseguite in risposta all'arrivo di qualunque evento di comunicazione.

Nel seguito prenderemo come modello di riferimento per i centri di tuple TuCSoN [21] (Tuple Centres Spread over Networks). Questo modello di coordinamento permette alle applicazioni Internet di servirsi di agenti mobili e con la capacità di adattarsi alla struttura di rete.

Lo spazio di coordinamento di TuCSoN possiede la duplice natura di spazio di interazione globale, costituito da astrazioni di comunicazione univocamente denotate, e quella di collezione di spazi di interazione locali. In questa maniera TuCSoN supporta efficacemente il ruolo duale degli agenti internet, sia come entità in

grado di trovare e accedere a dati e risorse sia come entità vaganti che trasferiscono la loro propria computazione di server in server. Come detto per i centri di tuple, TuCSon è anche un mezzo di coordinamento, cioè una astrazione di comunicazione il cui comportamento può essere definito in modo da promulgare leggi di coordinamento globali, così da arricchire il modello di coordinamento con quella flessibilità e quel controllo che gli originari modelli data-centrici non possedevano.

Come linguaggio di specifica delle reazioni per i centri di tuple TuCSon prenderemo invece come riferimento ReSpecT [22] (Reaction Specification Tuples): un linguaggio logico Turing equivalente per mezzo del quale è teoricamente possibile implementare qualsiasi politica di coordinamento computabile richiesta da un qualunque scenario applicativo. Inoltre, come vedremo più in dettaglio nel capitolo 4, ReSpecT supporta i centri di tuple come astrazioni di prima classe, cioè come astrazioni prese esplicitamente in considerazione come paradigma nel meta-modello per la progettazione di un sistema. Queste astrazioni sono tenute vive attraverso tutto il processo di progettazione del software, dalla fase di analisi fino a quella di rilascio, e permettono sia la verifica che la modifica dello stato di un componente a runtime.

Capitolo 3

Centri di tuple semantici

3.1 Esigenza dell'estensione semantica

Poiché l'apertura di un sistema ha a che vedere con l'eterogeneità dei componenti del sistema stesso, i sistemi computazionali complessi dovrebbero essere aperti all'integrazione di componenti diversi ed eterogenei dal punto di vista dell'architettura e della tecnologia. La questione dell'eterogeneità di taluni componenti del sistema, visti come tuple oppure come altre fonti di informazione, potrebbe essere affrontata rendendo gli agenti mobili consapevoli di tutti i modi possibili nei quali la conoscenza può essere rappresentata e organizzata [21].

Tuttavia, in questa maniera, la progettazione degli agenti software diventa un compito particolarmente complesso e porta alla realizzazione di una struttura altamente inflessibile che contrasta perciò con il principio di apertura del sistema.

Le tendenze attuali della ricerca nell'area dei middleware di coordinamento propongono il "semantic tuple space computing" [2], il quale arricchisce gli spazi di tuple semanticamente per far fronte all'eterogeneità della struttura delle tuple scambiate. Nell'ambiente TuCSoN, proprio in ragione di ciò, è fornita la possibilità di caricare sui centri di tuple il fardello dell'eterogeneità, colmando il dislivello tra i protocolli di interazione dell'agente e lo specifico modello di rappresentazione dell'informazione adottato in ogni scenario.

3.2 Semantica per il Web

Il "semantic tuple space computing" è stato introdotto inoltre per far fronte all'apertura nel contesto del Web Semantico e in quello dei Web Services Semantici che adottano il coordinamento basato sulle tuple al fine di ottenere un migliore disaccoppiamento nella comunicazione.

L'estensione è resa possibile facendo uso di spazi di tuple semanticamente arricchiti in modo da sfruttare gli spazi di tuple come magazzini di informazione semantica. Per esempio TSC (Triple Space Computing) [28] e Semantic Web Spaces [2] forniscono uno spazio di tuple in grado di immagazzinare triplette RDF sotto forma di tuple che descrivono rispettivamente Web Services [24] e risorse Web. In questi casi, gli spazi di tuple sono appunto sfruttati per coordinare Web Services e risorse Web. Ad ogni modo il limite del "semantic tuple space computing" è che si focalizza su di un contesto piuttosto specifico, perciò le soluzioni adottate dipendono dal contesto e sono strettamente accoppiate con tecnologie quali RDF e XML. Ciò le rende poco adatte come soluzioni di carattere generale o per scenari differenti come quelli ad informazione intensiva ed i sistemi a computazione è pervasiva.

3.3 Semantica e sistemi ad informazione intensiva

Una sfida ragguardevole per i modelli data-centrici che si basano sulle tuple viene appunto dagli scenari applicativi ad informazione intensiva, ovvero quegli scenari dove la maggior parte delle attività sono basate sulla conoscenza in qualche specifica forma e l'informazione diventa il mezzo principale attraverso il quale il sistema viene coordinato. Infatti, maneggiare l'informazione in sistemi basati sulle tuple porta spesso problemi in termini di sintassi, facendo svanire i benefici del coordinamento basato sulle tuple precedentemente individuati. Ad esempio due tuple che contengono lo stesso dato potrebbero non corrispondere a causa delle differenze nella struttura della tupla.

Finora il problema è stato affrontato in due maniere completamente diverse: *i)* sfruttando il coordinamento basato sulle tuple all'interno di un middleware per ambienti a informazione intensiva (esperimenti con il coordinamento basato sulle tuple all'interno di un middleware per il Web semantico [2]), oppure, *ii)* arricchendo l'astrazione dello spazio di tuple con un'interpretazione semantica, estendendo gli spazi di tuple con un'opera di descrizione logica così da supportare ogni tupla con opportuni template e operazioni sugli spazi di tuple come garanti di un'interpretazione semantica ben impostata.

Quest'ultima prospettiva, che indagheremo più in dettaglio, ci porta a definire la nozione di coordinamento semantico auto-organizzate [20] come una generalizzazione dei principi base e dei meccanismi del coordinamento e dell'auto-organizzazione negli ambienti a informazione intensiva. Ciò può essere definito come la gestione delle interazioni nei sistemi informazione intensiva dove le interazioni sono locali e richiedono la condivisione e il processamento della conoscenza. Gli effetti del coordinamento sull'informazione distribuita appaiono per emersione e attraverso l'auto-organizzazione.

3.4 Requisiti e componenti per l'estensione semantica

Al fine di ottenere una descrizione semantica dell'informazione è necessario rappresentare il contenuto delle tuple attraverso un linguaggio ontologico [19]. Su queste descrizioni sarà poi possibile effettuare un ragionamento logico al fine di supportare il confronto dell'informazione tra le tuple ed i templates. Dunque, introducendo un linguaggio ontologico per specificare una conoscenza tassonomica si avrà la possibilità di affrontare problemi eterogenei riguardanti i coordinabili che si riferiscono a informazioni semanticamente equivalenti ma aventi una rappresentazione sintattica differente.

Si potrà inoltre affrontare il problema dei processi di comunicazione tra i sistemi software e gli utenti nonostante le complicazioni che emergono dai differenti modi in cui l'informazione può essere rappresentata. Attraverso un linguaggio ontologico sarà anche possibile affrontare le dinamiche di sistema nella loro evoluzione ottenendo,

estendendo e adattando il dominio ontologico stesso. Da una prospettiva orientata all'informazione, un centro di tuple è interpretabile in maniera semplice e naturale come un magazzino di conoscenza costituito da un insieme di tuple, i cui comportamenti sono programmabili attraverso un insieme di specifiche delle reazioni.

Le tuple possono essere viste come oggetti rappresentanti del dominio applicativo il cui significato è descritto attraverso un'ontologia che divide i concetti e istituisce opportune relazioni tra di loro. Gli ingredienti di base dei centri di tuple semantici sono dunque [14]:

- **Ontologia del dominio:** un'ontologia che descrive i concetti del dominio connessi a un centro di tuple e che permette alle tuple immagazzinate di essere interpretate semanticamente. L'ontologia di un dominio applicativo può essere formalmente definita attraverso un componente di sistema chiamato TBox che contiene gli assiomi terminologici, i quali descrivono: *i*) i concetti, evidenziando l'insieme di individui del dominio significativi; *ii*) i ruoli, evidenziando le relazioni tra gli individui; *iii*) la tassonomia, relativa alla descrizione di concetti e ruoli.
- **Tuple semantiche:** una tupla rappresentante un individuo che può essere semanticamente interpretato per mezzo del dominio ontologico associato al centro di tuple. Le tuple semantiche possono essere formalmente definite attraverso un componente di sistema chiamato ABox, il quale definisce gli assiomi in grado di asserire specifici oggetti del dominio e le loro proprietà. Al fine di rappresentare un individuo sotto forma di una tupla attraverso il formalismo utilizzato dalla ABox è richiesto un linguaggio che possa esprimere nome, concetto e insieme di relazioni nelle quali l'individuo è coinvolto.
- **Templates delle tuple semantiche:** template di tuple usato per reperire tuple semantiche e che consiste nella specificazione di un insieme di individui del dominio per mezzo dell'ontologico del dominio. Dal momento che i template semantici sono specifiche relative a insiemi di individui del dominio, possono anche essere interpretati come descrizioni di insieme di individui esprimibili nel formalismo del TBox.
- **Primitive semantiche:** operazioni quali *in*, *rd*, *out*, che rappresentano il linguaggio attraverso il quale i componenti del sistema possono leggere, consumare e scrivere tuple. Quest'ultime rappresentano la conoscenza descritta per mezzo dell'ontologia del dominio. Le primitive semantiche rappresentano dunque il linguaggio utilizzato dai componenti del sistema. Per

questo motivo, estendere il modello a centro di tuple con le tecniche semantiche richiede di rivedere il comportamento delle primitive di base del centro di tuple. Una prima estensione consiste nel verificare che i ruoli ed i concetti associati ad un individuo esistano nel dominio ontologico.

- **Reazioni semantiche:** insieme di attività computazionali che avvengono all'interno del centro di tuple e che sono definite per mezzo di un linguaggio di specifica delle reazioni, il quale può associare ad un evento un insieme di reazioni che devono essere eseguite. Proprio grazie a queste reazioni è possibile, oltre ad accedere a tutte le informazioni relative allo scattare dell'evento di comunicazione, leggere, rimuovere e scrivere tuple dal/nel centro di tuple. Pertanto, proprio come i componenti coordinati, le reazioni possono adottare primitive di coordinamento per accedere e modificare la conoscenza semantica immagazzinata nel centro di tuple, attraverso tuple semantiche e template semantici.
- **Meccanismo di confronto semantico:** algoritmi che verificano le relazioni tra gli individui ed i concetti descritti dalle tuple e dai templates nell'esecuzione delle primitive e delle reazioni di coordinamento. Attraverso questo meccanismo è possibile identificare tuple utilizzando i template semantici. Il meccanismo di confronto deve inoltre identificare e recuperare l'insieme di individui specificati attraverso la descrizione di un concetto per mezzo dell'ontologia del dominio. Per poter implementare un meccanismo di confronto è possibile utilizzare un servizio di ragionamento come quello utilizzato per gli ABox ed i TBox.

Inoltre, per definire accuratamente un modello a centro di tuple semantico, è necessario introdurre una serie di formalismi, riguardanti le modalità in cui la conoscenza è rappresentata, ovvero le Logiche Descrittive (DL) [23].

Queste DL sono linguaggi logici con semantiche ben definite che permettono alle ontologie di essere condivise e scambiate in maniera coerente. Le DL consentono inoltre l'utilizzo di tecniche automatizzate di ragionamento sulle ontologie che possono essere sfruttate per fronteggiare l'evoluzione dinamica del vocabolario utilizzato per descrivere la conoscenza.

Tra le DL più espressive vogliamo ricordare SHOIN(D) che è la controparte teorica di OWL DL, una delle tre specie di OWL, un linguaggio descrittivo ontologico standard per il Web semantico definito dal consorzio W3C. Essendo OWL uno standard è anche in grado di soddisfare i requisiti di apertura. Di conseguenza anche

SHOIN(D) è in grado di arricchire i centri di tuple con ontologie e oggetti del dominio. Nella definizione formale di modello di centro di tuple semantico compaiono perciò i seguenti componenti appartenenti alla DL [19]:

- **TBox (Terminological Box):** descrive formalmente l'ontologia del dominio sottoforma di una terminologia ovvero un insieme di concetti e descrizioni di ruolo attraverso un insieme di costruttori (unione, intersezione, negazione, esistenza) ed operatori (inclusione, eguaglianza). Ogni centro di tuple è associato ad uno specifico TBox che descrive le semantiche relative alle informazioni contenute.
- **ABox (Assertional Box):** una serie di asserzioni che riguardano individui e ruoli che utilizzano la terminologia definita attraverso il TBox. Ogni tupla immagazzinata nel centro di tuple è descritta come un individuo ABox specificandone nome, concetto e insieme di ruoli.
- **Servizio di ragionamento:** opera sulle descrizioni dell'informazione fornite attraverso il linguaggio ontologico per supportare il confronto dell'informazione tra le tuple ed i templates.

Riassumendo, i centri di tuple semantici estendono i centri di tuple standard seguendo i concetti espressi dalle Logiche Descrittive. Scelta un'opportuna DL, è possibile codificare la conoscenza, anche incompleta, riguardo all'esistenza di alcuni individui del discorso e delle loro relazioni, nella ABox. La definizione dei concetti del dominio del discorso è codificata invece nella TBox. In un centro di tuple semantico ogni spazio di tuple è equipaggiato con un'unica TBox ed una iniziale ABox che descrive il dominio in questione come un'ontologia OWL.

Le tuple che popolano il centro di tuple, ora arricchite semanticamente, modellano gli individui ai quali si riferiscono le entità coordinate. Per reperire le tuple semantiche è necessario utilizzare dei templates semantici che esprimono concetti dell'ontologia. Intervengono qui gli algoritmi di ragionamento in grado di verificare eventuali problemi a livello della DL. I centri di tuple semantici promuovono dunque il disaccoppiamento semantico tra i componenti coordinati incapsulando la rappresentazione semantica del dominio del discorso all'interno del mezzo di coordinamento.

Capitolo 4

Semantic Health Coordination

4.1 Interoperabilità

L'interoperabilità, che può essere vista come la capacità di due o più sistemi eterogenei di comunicare insieme, è un obiettivo chiave nella progettazione di sistemi aziendali. Anche se il significato dell'interoperabilità è generalmente ben compreso [1], i metodi ed i processi per imbrigliarne le molteplici possibilità in casi reali appartenenti al dominio medico non lo sono.

Tra le diverse attività di ricerca nell'ambito degli e-Health, l'interesse sugli Electronic Health Record (EHR) [11], frammenti d'informazione sanitaria digitalizzata, è particolarmente alto. L'obiettivo, per quanto riguarda il dominio EHR, è di assicurare l'interoperabilità tra i frammenti, i quali vengono raccolti ed immagazzinati da diverse istituzioni sanitarie che costituiscono un ambiente distribuito e aperto. Poiché la definizione di standard è del tutto insufficiente a garantire l'interoperabilità in questo dominio, il progetto SemHealthCoord (Semantic Health Coordination) nasce con l'obiettivo di ingegnerizzare un middleware capace di fondere insieme la conoscenza delle diverse istituzioni sanitarie e di renderla fruibile anche al di fuori del sistema sanitario stesso, con particolare riferimento ad un contesto, come quello svizzero, nel quale ogni cantone dispone di un sistema sanitario autonomo.

Al fine di valorizzare i benefici apportati dai frammenti EHR, in termini di miglior sicurezza dei pazienti, riduzione del costo dei servizi sanitari e miglioramenti nelle tecniche di estrapolazione dei sondaggi, i sistemi dovrebbero assicurare l'interoperabilità adottando i seguenti requisiti [25]:

- **Distribuzione:** i frammenti EHR devono essere facilmente condivisi nonostante tali informazioni siano immagazzinate in maniera largamente diffusa tra una molteplicità di sistemi EHR.

- Apertura: i server localizzati nelle diverse istituzioni sanitarie che supportano i frammenti EHR potrebbero essere eterogenei in termini di tecnologia e potrebbero cambiare dinamicamente nel tempo.
- Sicurezza: è necessario supportare meccanismi di sicurezza al fine di evitare casi avversi che possano causare danni ai pazienti e violazioni della privacy.

4.2 Standard pre-semantica

Lo scenario di partenza, che andremo adesso ad analizzare rapidamente, vede la presenza di una molteplicità di standard per il formato dei frammenti EHR e la loro trasmissione, tra i quali i più rappresentativi sono [25]:

- Health Level Seven (HL7)[26]: un insieme di standard aperti per lo scambio, la gestione e l'integrazione di frammenti EHR. HL7 fornisce una Architettura dei Documenti Clinici(in breve CDA) ovvero uno standard che copre gli aspetti di messaggistica nella trasmissione dei frammenti EHR.
- Digital Imaging and Communication in Medicine (DICOM)[27]: uno standard per maneggiare e trasmettere informazione sotto forma di immagini mediche. Include sia la definizione di un formato che un protocollo di comunicazione di rete.

Standard come HL7 e DICOM non sono tuttavia sufficienti per raggiungere l'interoperabilità in un sistema sanitario. Infatti i vari sistemi EHR utilizzano formati e protocolli di trasmissione diversi, i quali risultano spesso incompatibili o sovrapposti nella loro area di copertura tematica.

4.3 Standard e iniziative post-semantica

Un ulteriore avanzamento verso il requisito dell'interoperabilità tra i frammenti EHR è stato introdotto dall'iniziativa openEHR [11] e dallo standard CEN EN 13606 [29] i quali condividono l'utilizzo di approcci semantici utili a descrivere i frammenti EHR basandosi sulla Definizione di un Archetipo di Linguaggio (ADL), un linguaggio formale per esprimere concetti del dominio applicativo. I vantaggi

dell'utilizzo delle tecniche semantiche consistono nella possibilità di supportare l'interoperabilità tra i frammenti anche se questi possiedono una differente struttura sintattica.

Quanto alla questione della realizzazione di un middleware che gestisca le transazioni soddisfacendo i requisiti di distribuzione, apertura e sicurezza, sono stati proposti approcci differenti. Integrating the Healthcare Enterprise (IHE) [30], un'iniziativa non-profit di esperti nel settore e-Health, raccomanda, per la descrizione semantica dei frammenti, l'utilizzo dello standard XDS [31]. XDS, che aspira a fornire un archivio di documenti sanitari provenienti da diverse istituzioni, richiede tuttavia l'immissione di dati manualmente per descrivere semanticamente i frammenti EHR in un apposito registro ebXML che sfrutta le tecniche di markup ereditate dall'XML. Il limite principale di questo approccio consiste nell'incompatibilità dei registri ebXML con le tecniche semantiche più avanzate che sfruttano ontologie e confronto semantico descritte al paragrafo 3.4. Un contributo importante in questo senso viene invece da Triple Space Computing (TSC) [28], che fornisce una soluzione differente basata sul modello degli spazi di tuple Linda.

Attraverso il modello a spazio di tuple, la coordinazione tra le entità del sistema avviene scambiandosi informazioni sottoforma di tuple in uno spazio comune condiviso come descritto al paragrafo 2.1. Le entità del sistema che devono essere coordinate comunicano tra di loro attraverso le primitive out, rd e in, rispettivamente per immettere, leggere e consumare associativamente tuple. TSC, in particolare, fornisce un modello di coordinamento generale per gestire tutti tipi di interazione tra le entità del sistema, le quali, non potendo fare assunzioni a priori sugli altri componenti del sistema, si avvalgono dell'estensione semantica per scambiare dati, sottoforma di tuple, contenenti descrizioni semantiche ottenute per mezzo di un'ontologia. Il limite del modello proposto da TSC deriva dall'adozione degli spazi di tuple Linda, i quali, una volta definito il comportamento, non possono essere aggiornati per far fronte alle esigenze dinamiche del sistema e delle specifiche applicazioni se non al costo eccessivo della riprogettazione di ogni agente partecipante. Ciò risulta del tutto impraticabile in uno scenario come quello sanitario dov'è essenziale garantire la continuità del servizio.

4.4 Obiettivi di SemHealthCoord

L'obiettivo che SemHealthCoord si propone è la realizzazione di un middleware di coordinamento che superi le limitazioni imposte dai modelli forniti da IHE e TSC. I servizi offerti dovrebbero essere supportati in maniera continuativa ed il sistema dovrebbe avere l'abilità di cambiare la sua configurazione a runtime con lo scopo di far fronte al dinamismo delle applicazioni.

Infatti, durante il ciclo di vita di un'applicazione, i requisiti possono cambiare, essere aggiunti o rimossi. Per esempio nuovi nodi potrebbero essere aggiunti alla rete e gli algoritmi di coordinazione potrebbero cambiare al fine di migliorare l'efficienza e l'efficacia dei componenti del sistema. In quest'ottica torna particolarmente utile l'utilizzo di modelli che separino il fardello del coordinamento dagli agenti partecipanti semplificandone la struttura e, avendo preso i servizi chiave come astrazioni di prima classe, sarà così possibile estendere il sistema con relativa semplicità anche a runtime. Al contrario, un middleware che non consente cambiamenti a runtime richiede lo spegnimento dell'intero sistema sia per effettuare aggiornamenti che per modificarne la configurazione. Casi del genere devono essere scongiurati in ogni modo, specialmente in scenari e-Health, che richiedono la disponibilità continua del servizio.

Capitolo 5

Conclusione

Nonostante gli sforzi profusi finora, il sistema sanitario completamente libero dalla limitazione cartacea resta ancora un obiettivo da centrare. I sistemi ad informazione intensiva basati sugli spazi di tuple, nonostante possedessero in teoria la capacità di gestire semplici sistemi EHR, si sono rivelati del tutto incapaci di gestire l'ardua questione dell'interoperabilità negli scenari e-Health reali.

Per far fronte al dinamismo e all'imprevedibile, che è sempre dietro l'angolo, un tassello chiave si è rivelato essere la programmabilità del comportamento del mezzo condiviso e quindi la nozione di centro di tuple [3]. La netta separazione delle questioni concernenti il coordinamento dai coordinabili, che il modello a centro di tuple porta con sé, è in grado di semplificare la progettazione dei componenti del sistema ed apre le porte all'estendibilità. Infatti, non è più necessario decidere tutte le caratteristiche di un sistema durante la fase di progettazione ma è possibile modificare il comportamento del sistema anche a runtime per far fronte alle esigenze di specifiche applicazioni che potrebbero occorrere successivamente.

Recentemente, l'evoluzione dei sistemi distribuiti ha portato inevitabilmente un accrescimento della complessità delle infrastrutture di coordinamento e, come logica conseguenza, alla teoria dei sistemi a coordinamento auto-organizzante [20]. Applicare questa teoria ad uno scenario ad informazione intensiva come quello e-Health richiede tuttavia un'estensione semantica del modello a centro di tuple.

Un approccio interessante, seppur ancora allo stato sperimentale, è stato implementato per il modello TuCSon [19]. L'estensione semantica porta con sé una rivoluzione concettuale nel modo in cui l'informazione è intesa all'interno di un sistema. Con l'ausilio di queste tecniche sarebbe infatti possibile semplificare sia l'usuale modalità di accesso all'informazione che le attività di rilevamento statistico tra i dati di sistema fino al limite della naturalezza.

Bibliografia

- [1] European Union, "Commission Recommendation on cross-border interoperability of electronic health record systems," *Official Journal of the European Union* 2008/594/EC, 2008, (notified under document number C(2008) 3282).
- [2] Lyndon J. B. Nixon, Elena Simperl, Reto Krummenacher, and Francisco Martín-Recuerda. Tuplespace-based computing for the Semantic Web: A survey of the state-of-the-art. *The Knowledge Engineering Review*, 23(2):181-212, 2008.
- [3] A. Omicini and E. Denti, "From Tuple Spaces to Tuple Centres," *Science of Computer Programming*, vol. 41, no. 3, pp. 277-294, November 2001.
- [4] L.T. Khon, J.M. Corrigan, and M.S Donaldson. *To Err is Human: building a safer health system*. National Academy Press, 2000.
- [5] J. Denny, D. Giuse, and J. Jirjis. *The Vanderbilt Experience with Electronic Health Records*. *Seminars in Colon and Rectal Surgery*, 12:59-68, 2005.
- [6] W.J. van der Kam, P. W. Moormanb, and M.J. Koppejan-Mulder. Effects of electronic communication in general practice. *International Journal of Medical Informatics*, 60:59-70, 2000.
- [7] G.O. Barnett and H.J. Sukenik, "Hospital Information Systems," in *Future Goals of Engineering in Biology and Medicine*, J.F. Dickson and J.H.U. Brown, Eds. New York: Academic Press, 1969.
- [8] R.S. Dick, E.B. Steens, and (Editors), *The Computer-Based Patient Record: An Essential Technology for Health Care*: Institute of Medicine, National Academic Press, 1991.
- [9] C. Safran, D.Z. Sands, and D.M. Rind, "Online medical records: a decade of experience," *Methods Inf Med*, vol. 38, pp. 308-312, 2000.
- [10] F. Zambonelli, H. V. D. Parunak, *Towards a paradigm change in computer science and software engineering: A synthesis*, *The Knowledge Engineering Review* 18 (4) (2003) 329-342.
- [11] eHealth Media Ltd., "OpenEHR Foundation launches international standard," 2006.

- [12] G.A. Papadopoulos and F. Arbab, "Coordination models and languages," in *The Engineering of Large Systems.*: Academic Press, 1998, pp. 329-400.
- [13] D. Gelenter, "Generative communication in Linda," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 7, no. 1, pp. 80-112, 1985.
- [14] E. Nardini, *Semantic coordination through programmable tuple spaces*. <http://apice.unibo.it/xwiki/bin/download/Theses/NardiniPhdTheses/NardiniPhdThesis.pdf>, 2011.
- [15] Eric Freeman, Susanne Hupfer, and Ken Arnold. *JavaSpaces Principles, Patterns, and Practice: Principles, Patterns and Practices*. The Jini Technology Series. Addison-WesleyLongman, June 1999.
- [16] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: abstraction and reuse of object-oriented design*, pages 701-717. Springer-Verlag New York, Inc., New York, NY, USA, 2002.
- [17] Rossi, D., Cabri, G. & Denti, E. 2001 Tuple-based technologies for coordination. In *Coordination of Internet Agents: Models, Technologies, and Applications*. Berlin, Germany: Springer-Verlag, pp. 83–109.
- [18] F. Arbab, P. Ciancarini, C. Hankin, *Coordination languages for parallel programming (special issue on coordination languages for parallel programming)*, *Parallel Comput.* 24(7) (1998) 989–1004.
- [19] E. Nardini, M. Viroli, and E. Panzavolta, "Coordination in Open and Dynamic Environments with TuCSon Semantic Tuple Centres," in *The 25th Annual ACM Symposium on Applied Computing (SAC 2010)*, Sierre, Switzerland, 2010, In press.
- [20] Andrea Omicini and Mirko Viroli. *Coordination models and languages: From parallel computing to self-organisation*. *The Knowledge Engineering Review*, 26(1):53-59, March 2011. Special issue for the 25th Years of the Knowledge Engineering Review.
- [21] A. Omicini and F. Zambonelli, "Coordination for Internet Application Development," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 3, pp. 251-269, September 1999.
- [22] A. Omicini and E. Denti, (2001a). *Formal ReSpecT*. *Electronic Notes in Theoretical Computer Science*, 48:179-196. *Declarative Programming - Selected Papers from AGP 2000*, La Habana, Cuba, 4-6 December 2000.

- [23] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 3rd edition, 2010.
- [24] (2006) TripCom - Triple Space Communication. [Online]. <http://www.tripcom.org/>
- [25] D. Kalra, "Electronic health record standards," *IMIA Yearbook of Medical Informatics*, 2006.
- [26] N. Carriero, D. Gelernter and L. Zuck, "Bauhaus Linda", in [29], pp. 66-76.
- [27] N. H. Minsky and J. Leichter, "Law-Governed Linda as a Coordination Model", in [29], pp. 125-145.
- [28] Picco, G. P., Balzarotti, D. & Costa, P. 2005 LIGHTS: a lightweight, customizable tuple space supporting context-aware applications. In *Proceedings of the 20th ACM Symposium on Applied Computing*. Sante Fe, NM, USA: ACM Press, pp. 1134–1140.
- [29] P. Ciancarini and D. Rossi, "Jada: Coordination and Communication for Java Agents", *Second International Workshop on Mobile Object Systems: Towards the Programmable Internet (MOS'96)*, Linz, Austria, July, 1996, LNCS 1222, Springer Verlag, pp. 213-228.
- [30] P. Wyckoff, S. W. McLaughry, T. J. Lehman and D. A. Ford, "T Spaces." *IBM Journal of Research and Development*, vol. 37 (3- Java Technology); pp. 454–474, 1998.