



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
SEDE DI CESENA

Ecosistemi di servizi per sistemi pervasivi auto-organizzanti

Marco Sbaraglia
matricola 0000290922
`marco.sbaraglia@studio.unibo.it`

corso di Sistemi Multi-Agente LS
Seconda Facoltà di Ingegneria
giugno 2009

1 Introduzione

Una schiera sempre più fitta, densa e decentralizzata, di dispositivi computazionali (smartphone, notebook, navigatori satellitari, ma anche reti wireless di sensori) sta ogni giorno di più popolando il nostro ambiente quotidiano, integrandosi nella rete di telecomunicazioni e in internet, grazie alla diffusione sempre più imponente delle connessioni senza fili.

Questa straordinaria esplosione impone di considerare l'introduzione di innovativi servizi pervasivi, servizi *context-aware*, tramite i quali sfruttare le enormi potenzialità di questa nube computazionale e in grado di rendere matura e produttiva l'interazione degli eterogenei dispositivi digitali con il mondo fisico e sociale e che ci circonda.

In questo contesto emerge la necessità di un'innovativa piattaforma software, unica e aperta, che ospiti e orchestri l'esecuzione dei nuovi servizi general-purpose e gestisca la notevole quantità di dati contestuali coinvolta in questi scenari, tenendo conto dell'intrinseco non determinismo. Essa dovrà prevedere comportamenti auto-adattativi, tenere conto della diversità crescente delle richieste degli utenti (non più semplici fruitori, ma veri e propri distributori di conoscenza e servizi), tollerare future evoluzioni e innovazioni, senza prevedere significative re-ingegnerizzazioni.

A tal fine sembra insostenibile l'idea di estendere i modelli e i framework attuali in termini di componenti aggiuntivi, soluzioni add-on che cerchino di catturare le emergenti caratteristiche di una rete sempre più pervasiva e imprevedibile e di risponderne alle sue rinnovate esigenze: questo porterebbe ad un ulteriore aumento della complessità dei modelli e delle soluzioni esistenti, introducendo tra l'altro contrastanti trade-off tra le differenti soluzioni.

Inoltre, alla luce della seguente, centrale domanda [2]:

Is it possible to conceive a radically new way of modelling integrated network services and their execution environments, such that the apparently diverse issues of enabling pervasiveness, context-awareness, dependability, openness, diversity, flexible and robust evolution, can all be uniformly addressed once and for all?

appare necessario un ripensamento fondazionale dei modelli e una decostruzione delle correnti architetture e delle infrastrutture che le supportano.

2 Nuove esigenze e scenari

Le caratteristiche chiave che i servizi di nuova generazione dovranno esibire sono [2]:

Situatedness — Sia dovranno saper leggere il contesto spaziale e sociale degli utenti, adattandosi per meglio catturare la situazionalità dei loci del sistema, sia, data l'integrazione delle infrastrutture nello spazio fisico, dovranno

gestire concetti di spazio e dati spaziali, al fine di interagire sulla base della prossimità.

Adaptivity — In scenari fortemente dinamici e decentralizzati, i servizi dovranno essere in grado di auto-organizzarsi, auto-adattarsi e auto-gestirsi, al fine di sopravvivere all'inaffidabilità dei componenti, alla completa apertura e ai multiformi contributi degli utenti, il tutto senza (o comunque limitato) intervento umano.

Prosumption and Diversity — Al fine, tra l'altro, di permettere agli utenti di essere anche produttori di dati e servizi, è necessaria la capacità di supportare modelli aperti di produzione ed utilizzo, senza limitare a priori il numero e le classi di servizi disponibili.

Eternity — Sostenere il cambiamento delle richieste degli utenti e l'evoluzione tecnologica nel tempo, impone la capacità di tollerare l'evoluzione della struttura, dei componenti e dei pattern di utilizzo, senza significative (e costose) re-ingegnerizzazioni.

Un semplice caso di studio [2] può aiutare a chiarire i concetti espressi, delineando gli scenari dell'innovativo modello proposto.

Si considerino i display digitali che sempre di più ci circondano, a partire da quelli dei nostri dispositivi portatili o domestici sino alle grandi installazioni pubblicitarie permanenti, tipiche delle metropoli. Attualmente, nella stragrande maggioranza dei casi, essi sono concepiti per mostrare le informazioni in modo cablato/manuale o comunque predefinito, ad esempio ciclando una serie di messaggi pubblicitari prepagati, risultando quindi del tutto insensibili al contesto e alle specifiche necessità degli utenti.

Affinché un tale dispiego di risorse possa essere molto più efficiente e vantaggioso sia per i committenti sia per gli utenti, è necessario un approccio completamente diverso. Le informazioni dovrebbero essere mostrate in base allo stato corrente dell'ambiente fisico (si immagini lo spot di una bevanda rinfrescante in una giornata fredda: non sarebbe gradito né al committente né agli utenti), sfruttando le informazioni provenienti da sensori circostanti (ad es, quelli usati dalle stazioni meteo) e possibilmente orchestrando l'attività tra display vicini, per evitare di irritare gli utenti mostrando ovunque lo stesso messaggio. Inoltre i display potrebbero adattare le informazioni da mostrare in base alle caratteristiche degli utenti che si trovano nelle vicinanze (ad es, proiettare spot di abbigliamento sportivo e casual in presenza di una maggioranza di teenager), tenendo conto di ciò che hanno già visto nel recente passato, fino a prevedere la possibilità da parte degli utenti stessi di arricchire l'offerta informativa, esprimendo preferenze e necessità.

Infine, tale infrastruttura deve essere in grado di redistribuire e reinterpretare le informazioni da mostrare al variare delle condizioni, anche topologiche e fisiche dell'infrastruttura stessa (il guasto di un display o l'introduzione di uno nuovo),

tollerando in più future variazioni tecnologiche dei display o dei sensori attorno ed eventuali evoluzioni delle strategie di presentazione.

3 Le metafore naturali

Per raggiungere questi traguardi, la direzione più promettente sembra essere quella di trarre ispirazione dai sistemi naturali, dove concetti di spazialità, auto-organizzazione, auto-gestione e evoluzione a lungo termine sono intrinseci ai sistemi stessi.

E' pur vero che le metafore naturali sono già state ampiamente studiate nell'ambito dell'intelligenza artificiale e della *distributed computing*, tuttavia gran parte degli sforzi in questo senso hanno portato solamente all'implementazione di soluzioni algoritmiche specifiche o a realizzazioni particolari di servizi distribuiti. La diversità e la portata innovativa di questa proposta consistono nell'individuare nei sistemi naturali la metafora chiave attorno alla quale concepire, modellare e sviluppare framework in grado di supportare in modo completamente maturo i servizi pervasivi di nuova generazione.

In essi, come nell'architettura concettuale proposta, è possibile riconoscere le seguenti caratteristiche comuni [2]:

above a spatial environmental substrate, individuals (or particles, or components) of different kinds interact, compete, and combine with each other according to complex patterns in respect of same basic laws of nature.

That is, what you eventually perceive is always a sort of natural "ecosystem".

Il sostrato è un'infrastruttura condivisa e pervasiva, all'interno della quale individui diversi e spazialmente situati (i servizi), perseguendo i loro specifici obiettivi nel rispetto di un limitato numero di leggi fondamentali, danno vita ad un ecosistema. Le dinamiche di questo sistema, determinate dall'attuazione delle regole di base, mostreranno spontaneamente caratteristiche di auto-organizzazione, auto-adattabilità ed evoluzione a lungo termine, piuttosto che le peculiarità dei suoi individui.

La differenza chiave nell'ambito dei possibili approcci alla realizzazione del meta-modello illustrato, sta nella metafora scelta come fondamento dell'ecosistema. Gli studi finora condotti in tali ambiti suggeriscono quattro metafore su tutte, le quali corrispondono tra l'altro a livelli d'astrazione via via più alti: metafore fisiche, chimiche, biologiche e sociali. Di esse segue una breve disanima [2].

Metafore fisiche — I vari servizi presenti nel sistema sono considerati come particelle computazionali di specie differenti, influenzate da altre particelle e da campi computazionali virtuali. Le leggi dell'ecosistema ne determinano l'evoluzione: al variare delle condizioni del sistema, variano forma e gradiente dei campi e di conseguenza le particelle, in base alla loro descrizione e ad una

qualche funzione di matching, possono cambiare il loro stato, possono aggregarsi o diffondersi, sempre nel perseguimento dei loro specifici obiettivi.

Metafore chimiche — I molteplici servizi presenti nel sistema sono considerati come atomi/molecole computazionali, le cui proprietà sono esplicitate da descrizioni semantiche. Le regole che realizzano l'ecologia sono una sorta di leggi chimiche che stabiliscono le reazioni tra i componenti, tenendo conto dello stato del sistema, grazie alle concentrazioni dei componenti stessi. La loro attuazione può portare ad aggregazioni, alla crescita o al decadimento dei componenti, alla nascita di nuovi. L'infrastruttura è strutturata in un set di località (tipicamente gli space), intese come soluzioni dove si realizzano le reazioni chimiche.

Metafore biologiche — Tipicamente si focalizzano sui sistemi biologici di piccola scala, come ad es. le cellule e le loro interazioni oppure colonie di organismi semplici; di conseguenza i servizi del sistema sono modellati da semplici cellule oppure da organismi triviali (ad es. formiche), guidati da obiettivi elementari. Le interazioni hanno luogo in base a segnali (feromoni) di vario genere, lasciati dai componenti stessi e ai quali i componenti reagiscono, non solo in base alla tipologia, ma anche al loro stato corrente. Le leggi fondamentali si occupano unicamente di stabilire come i feromoni si diffondono o decadono nel tempo.

Metafore sociali — Focalizzata sui sistemi biologici a livello delle specie animali e delle loro interazioni, concettualizza i componenti del sistema come individui di diverse specie, mossi dalla ricerca di specifici nutrimenti (risorse computazionali o altri componenti), senza i quali sono destinati ad estinguersi. I dati e le risorse sono modellati come forme di vita passive (vegetali) e le leggi dell'ecosistema determinano come e sotto quali condizioni gli individui siano autorizzati a cercare cibo, a nutrirsi ed eventualmente a riprodursi. L'infrastruttura è organizzata in una serie di nicchie che abilitano le interazioni e permettono la diffusione delle specie attraverso le località.

Ciascuna delle metafore descritte presenta pregi e difetti nel rispondere alle diverse esigenze di auto-organizzazione. In particolare, sempre tenendo sotto controllo la complessità del modello, delle leggi fondamentali, dei componenti e della struttura, i differenti approcci devono essere valutati in termini di [2]:

- capacità dell'ecosistema di esprimere autonomamente varie forme di auto-organizzazione, in modo da adattare struttura e comportamento del sistema alle specifiche contingenze;
- capacità dell'ecosistema di tollerare i cambiamenti nel tempo, così da poter accogliere nuove specie di servizi e allo stesso tempo di sopravvivere all'estinzione di specie, prevedendo comportamenti assai differenziati;
- necessità da parte degli operatori umani di esercitare controllo diretto sul sistema e sulle sue parti, sia in fase di progetto sia a run-time.

Tuttavia i pro e i contro delle metafore non verranno trattati in queste poche pagine, anche perché un discorso strettamente teorico e slegato da concreti percorsi di ricerca appare piuttosto sterile; più avanti ci si limiterà ad illustrare brevemente i motivi della scelta che sta alla base dei *Biochemical Tuple Spaces for Self-Organising Coordination* [3].

4 Spazi di tuple programmabili

Quale strumento di coordinazione permette di realizzare l'infrastruttura descritta? Dalla rassegna che segue, una breve analisi delle caratteristiche portanti del meta-modello individuato [1], gli spazi di tuple programmabili (ad es. TuCSoN) risultano naturalmente adatti a rispondere alle esigenze della *self-organising coordination*.

Topology — I media di coordinazione e gli agenti (i servizi) sono distribuiti in varie location: un agente risiede generalmente su un nodo, mentre il media di comunicazione è distribuito su di una sottoporzione connessa del sistema. Ogni location è quindi direttamente connessa ad un ristretto numero di nodi vicini.

Locality — Dalla topologia dipende la portata delle interazioni; quale che sia il tipo di interazione tra i servizi (mediata o non mediata), essa avviene sempre localmente. Interazioni di lungo raggio risultano come composizione di interazioni locali e gli agenti devono avere almeno una visione locale del sistema.

On-line character — In più rispetto alle tipiche regole di coordinazione, reattive alle interazioni, le leggi fondamentali che determinano l'ecosistema dei servizi devono reagire anche al trascorrere del tempo. Si può immaginare quindi un servizio di background sempre attivo, che evolva lo stato del sistema nel tempo.

Time — Come evidenziato dalla precedente proprietà, il tempo è una variabile fondamentale nei processi auto-organizzanti. La coordinazione deve quindi essere arricchita del concetto di tempo, tramite primitive *time-dependent* oppure definendo regole che ne attivano altre con il passare del tempo.

Probability — Proprio come avviene in natura, deve poter essere realizzato un modello stocastico. Oltre il non determinismo del sistema da orchestrare e dei semplici spazi di tuple (ad es. LINDA), è necessario poter specificare che alcuni risultati siano più probabili di altri e quindi prevedere anche quei percorsi caratterizzati da più bassa probabilità, comunque necessari per affrontare l'imprevedibilità del sistema.

TuCSoN, oltre a supportare la fondamentale caratteristica di disaccoppiamento spaziale e temporale, dà la possibilità di rispondere concretamente a queste esigenze: ogni nodo della rete deve essere dotato di un centro di tuple, opportunamente equipaggiato con regole dichiarative scritte in linguaggio

ReSecT, tramite le quali realizzare le leggi fondamentali dell'ecologia, in base alla metafora scelta ed alle esigenze del sistema. In particolare, è la possibilità di implementare e gestire i concetti di tempo e probabilità che rende TuCSOn particolarmente adatto ad essere utilizzato come infrastruttura di base per la *self-coordination*.

5 Biochemical Tuple Spaces for Self-Organising Coordination [3]

Dallo scenario finora delineato e in virtù degli elementi riportati, ha origine l'idea dei *Biochemical Tuple Spaces for Self-Organising Coordination*.

In particolare la necessità di reificare lo stato dei componenti e del sistema nei nodi dell'infrastruttura, evolvendolo alla stregua di uno stato chimico, introduce l'utilizzo di strutture dati (tuple) contraddistinte da un peso/concentrazione. Tale valore misura la pertinenza/attività del componente che la tupla rappresenta e determina, oltre alla probabilità di prelevarla dallo spazio, l'influenza che il componente avrà sulla coordinazione. Quanto detto costituisce un'estensione stocastica dei modelli classici, quali LINDA, e abilita la possibilità di evolvere la concentrazione delle tuple nel tempo tramite regole *chemical-like*.

Le tecniche usate per realizzare questo meccanismo sono ereditate dagli studi di *Computational Systems Biology* (CSB) e sono fondate concretamente sulla teoria del π -calculus stocastico.

Più nel dettaglio, il modello in questione si basa sulle catene di Markov tempo-continue (*Continuous-Time Markov Chains*, CTMC), cioè grafici i cui nodi rappresentano gli stati del sistema e in cui gli archi, le transizioni da stato a stato, sono caratterizzati da numeri reali positivi, detti *rates*. Un rate rappresenta la frequenza media della transizione e la distanza di tempo tra due occorrenze della stessa segue una distribuzione di probabilità esponenziale negativa. La probabilità che una transizione (le transizioni non sono altro che le leggi di stampo chimico che regolano l'ecologia) sia attivata dipende, oltre che dal rate che la caratterizza, anche dalla concentrazione dei reagenti coinvolti (ad es, per la transizione $x+y \rightarrow z$ di rate r , il rate markoviano R è dato dal prodotto di numero di molecole di x , per numero di molecole di y , per r). Ad ogni step si calcolano tutti i rate R_i equivalenti e, detta s la loro somma, si applica probabilisticamente solo una transizione, quella caratterizzata dal più alto R_i/s . L'intero processo è ripetuto ogni intervallo di tempo (l'intervallo di tempo è determinato dalla reazione attivata e calcolato come detto) ed è chiaramente influenzato dal variare delle concentrazioni dei reagenti, cioè dal tipo e dal numero di servizi presenti nel dato centro di tuple.

Questo algoritmo implementa il comportamento sempre attivo dell'infrastruttura di coordinazione e sarà il programma che, insieme con le leggi che definiscono l'ecologia del sistema, equipaggia i centri di tuple, facendo così emergere le

proprietà di evoluzione spontanea e di auto-organizzazione che caratterizzano le dinamiche dei sistemi biochimici.

Infine, il sistema di coordinazione, organizzato in un set di centri di tuple dotato di una struttura topologica, prevede che ogni nodo abbia un certo numero di centri di tuple “vicini”, collegati direttamente ad esso. L’interazione tra i centri di tuple segue il *linkability-model* ed è ottenuta tramite un particolare tipo di legge chimica che, oltre a modificare le concentrazioni, dà origine a *firing tuples*, cioè tuple pronte per essere inviate ad uno dei centri nelle vicinanze, scelto probabilisticamente. Questo meccanismo realizza il concetto di compartimenti biologici, i cui confini possono essere oltrepassati dalle sostanze chimiche, e dà corpo alla metafora che consente di concepire il sistema come una rete biologica di nodi, giustificando il termine “*biochemical*” *tuple spaces*.

6 Un caso di studio [3]

Tralasciando per ragioni di sintesi l’analisi della sintassi e della semantica operativa alla base del modello, si prende in esame un semplice scenario di competizione spaziale dei servizi, ulteriormente utile alla comprensione dell’espressività e delle opportunità del modello di coordinazione proposto.

Si inizia considerando un singolo centro di tuple che media le interazioni tra servizi ed utenti. Non sono presenti a priori informazioni sui servizi disponibili, sulla loro capacità o meno di soddisfare le richieste degli utenti; di conseguenza è necessario definire un matching semantico che sia in grado di associare dinamicamente ciascuna richiesta al servizio più appropriato.

Non è d’interesse, tuttavia, affrontare la questione ontologica della funzione di matching, l’intento è quello di concertare un sistema che, indipendentemente dalla funzione scelta, manifesti un comportamento ecologico. In particolare si desidera che autonomamente: (i) i servizi che non attraggono utenti si estinguano dal sistema, (ii) i servizi di successo, viceversa, attraggano sempre più nuove richieste e infine (iii) i servizi con aree di competenze sovrapposte competano tra loro per la sopravvivenza. Seguono esempi di protocollo per i servizi e per gli utenti:

```
def Ds := out(σ, publish(service(ids, desc))).call D's
def D's := in(σ, toserve(service(ids, desc), request(Idc, Req))).
           out(σ, reply(Idc, Rep)).call D's
def Dc := out(σ, request(idc, req)).in(σ, reply(idc, Rep))
```

Gli agenti che realizzano i servizi, effettuando inizialmente *call Ds*, dichiarano per prima cosa il loro ruolo, pubblicando la descrizione del servizio; da quel momento in poi consumano ciclicamente richieste per il loro servizio, inserendo una tupla di reply per ciascuna richiesta presa in carico (*call D's*).

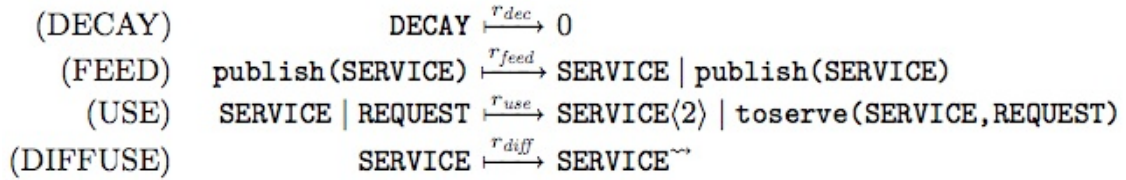


Fig. 3 Leggi chimiche per l'ecosistema di servizi

Dualmente i client, effettuando *call Dc*, pubblicano tuple di richieste di servizio nello spazio e rimangono in attesa di riceverne le reply. Da notare che è il centro di tuple a farsi carico di legare richieste e servizi, creando tuple *toserve* nel momento in cui la funzione di matching associ una richiesta al servizio più adatto a gestirla. Questo comportamento può essere automaticamente messo in atto installando in ogni nodo le regole mostrate nella Figura 3 e definendo una opportuna funzione di matching che, oltre a legare richieste e servizi (determinandone quindi il successo o l'estinzione), regoli opportunamente i rate chimici, come spiegato nei successivi esempi.

Pubblicazione di servizi La legge chimica DECAY asserisce che ogni tupla τ può eventualmente decadere con dinamica esponenziale negativa (**DECAY** è una variabile): l'idea è che la funzione di matching μ dovrebbe regolare il rate di decadimento complessivo, dato da $\mu(\text{DECAY}, \tau) * r_{dec}$; essa restituisce un'associazione perfetta con quelle tuple che dovrebbero decadere più velocemente come quelle che rappresentano i servizi, oppure non corrispondere affatto nel caso delle tuple di richiesta o di risposta.

La regola chimica FEED è usata per inserire tuple **service(ids,desc)** nello spazio, in modo da mantenere crescente la relativa concentrazione del servizio: questo processo di promozione si arresta quando decade la tupla **publish** che rappresenta il servizio. Anche in questo caso il rate che caratterizza questa legge dipende dalla funzione di matching. La Figura 4 (a) mostra un esempio in cui il rate di decadimento è 0.01, il feed rate di 10 (si noti come la concentrazione raggiunga un equilibrio attorno al valor medio di 1000) e il rate di decadimento relativo alle tuple **publish** è di 0.001 (la reazione FEED cessa di essere attivata attorno alle 1000 unità di tempo). Una volta terminato il processo di promozione e senza utenti che facciano uso del servizio in questione, il servizio inevitabilmente diminuisce la sua concentrazione fino ad estinguersi.

Raggiungimento dell'equilibrio Mentre il meccanismo di pubblicazione può essere usato per promuovere un servizio, è successivamente necessario che ci siano utenti che comincino a sfruttare il servizio, dando così un feedback positivo che sostenga e aumenti la concentrazione del servizio nello spazio di tuple. La regola USE fornisce il meccanismo tramite il quale servizi ed utenti si

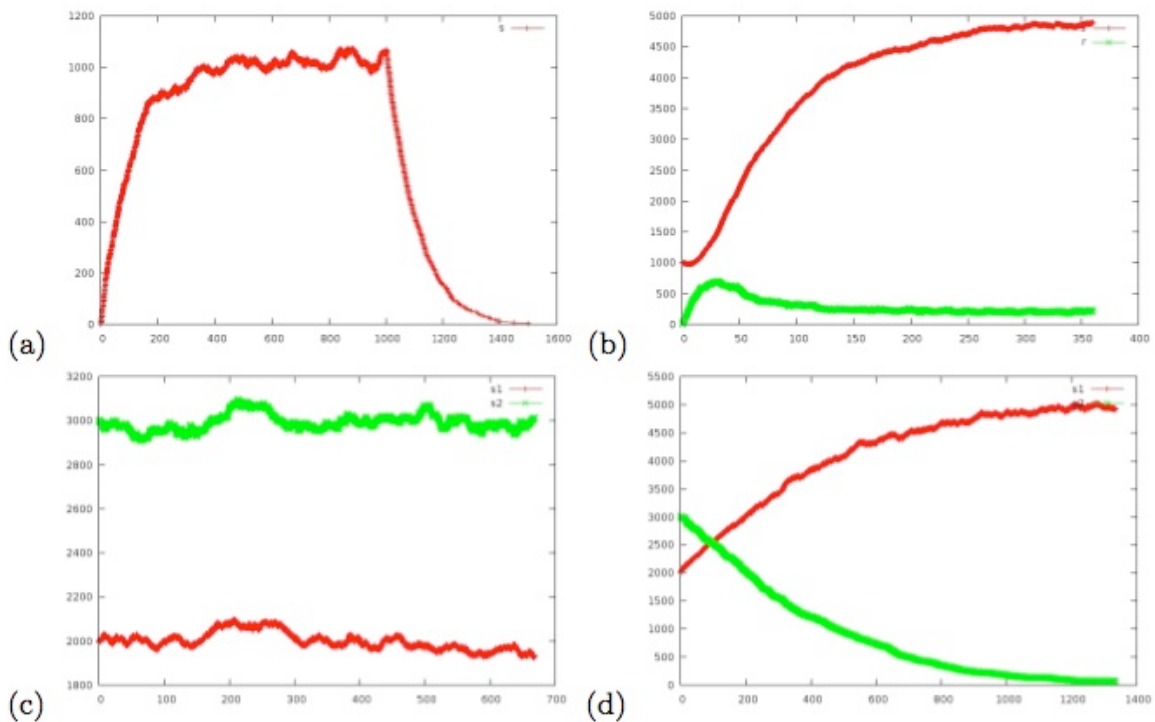


Fig. 4 (a) Pubblicazione di servizi, (b) Uso permanente di un servizio (c) Competizione simmetrica di servizi, (d) Competizione asimmetrica di servizi

legano: sono necessarie una tupla **service** ed una **request** che corrispondano e ne scaturirà sia una tupla **toserve** sia un incremento unitario della concentrazione del servizio. La funzione di matching può essere costruita per dare a questa legge un alto rate nel caso di una forte corrispondenza tra richiesta e servizio, in base ad un qualche criterio semantico (ad es, quanto è specializzato un servizio nel servire una data richiesta oppure quale porzione di una richiesta articolata esso può servire). Come esempio si supponga che un servizio abbia concentrazione di 1000, rate di decadimento di 0.01 e che le richieste arrivino al rate di 50, con la regola USE che lega il servizio alle richieste con un rate di 0.00005: il comportamento che ne risulta è mostrato in Figura 4 (b). Il sistema raggiunge l'equilibrio per una concentrazione del servizio di 5000 e fissa il numero di richieste non soddisfatte a poche centinaia. In altre parole, dopo un iniziale crescita, la concentrazione del servizio si stabilizza ad una concentrazione proporzionale al rate con cui serve le richieste, mentre il rate di utilizzo è responsabile della reattività del servizio.

Competizione tra servizi Le leggi chimiche viste finora supportano intrinsecamente la competizione tra i servizi. Si consideri lo scenario precedente, ora con due servizi che soddisfano in modo equivalente le stesse richieste, entrambi con rate di decadimento di 0.01 e rate di utilizzo di 0.05, ma con concentrazioni iniziali di 2000 e 3000. Dalla Figura 4 (c) in cui è riportata

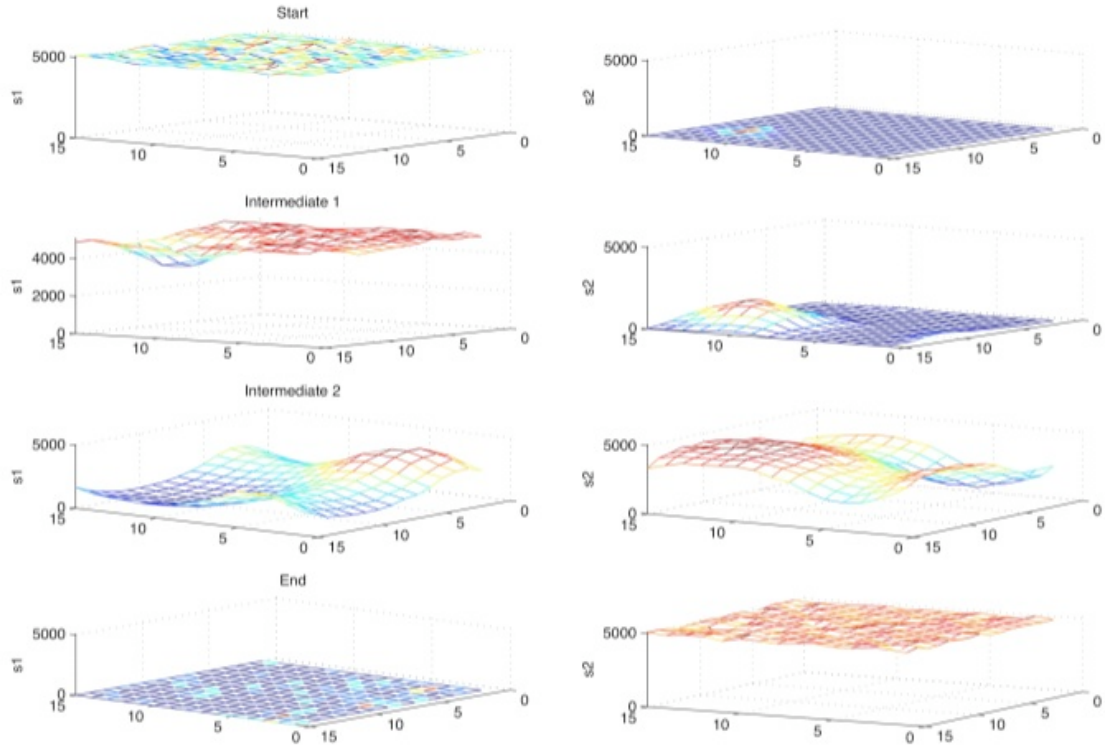


Fig. 5 Competizione spaziale tra due servizi in una *torus-like network*: 4 passaggi

l'evoluzione di questo scenario si può vedere che le concentrazioni restano in equilibrio allo stato iniziale.

Questo equilibrio è tuttavia improbabile in pratica, poiché nell'esempio considerato i due servizi sono del tutto identici, ovvero due istanze dello stesso servizio. Infatti, se i servizi avessero anche solo una piccola differenza nel rate d'utilizzo (ad es, a causa di prestazioni diverse rispetto alla funzione di matching), uno dei due perderebbe la competizione, diminuendo fino a scomparire, come mostrato in Figura 4 (d), dove i rate di utilizzo sono 0.06 e 0.04. Nonostante sia presente in una minore concentrazione iniziale, il processo con maggiore rate di utilizzo vince la competizione e sopravvive all'estinzione dell'altro.

Competizione in un sistema spaziale Ora si supponga che, invece di un singolo centro di tuple, si abbia una rete di nodi, tutti programmati con lo stesso set di leggi chimiche, mostrate in Figura 3, per la diffusione e la competizione di servizi. Si supponga inoltre che la diffusione riguardi solamente i servizi, cioè che la funzione di matching $\mu(\text{SERVICE}, \tau)$ restituisca esattamente 1 per le tuple di servizio e 0 per tutte le altre. Tale sistema può essere usato per supportare uno scenario pervasivo in cui l'infrastruttura coordini utenti e servizi e in cui, quando un servizio viene iniettato in un nodo della rete, cominci a diffondersi un passo alla volta, fino a coprire potenzialmente tutta la rete.

Questa diffusione vorrebbe riprodurre il modo in cui le sostanze chimiche si diffondono in un tessuto cellulare.

Si consideri come scenario di riferimento per l'esempio una *torus-like network* di 15x15 nodi, cioè una griglia dove i nodi sulle estremità sono connessi con quelli sul confine opposto, in modo che ogni nodo abbia 4 nodi vicini. In ogni nodo le richieste di utilizzo del servizio arrivano ad una frequenza fissata; inizialmente è presente un unico servizio s1 in grado di servire le richieste. Ne consegue che in tutti i nodi si riscontra la situazione esemplificata in Figura 4 (b) dove l'unico servizio s1 ha approssimativamente concentrazione 5000, dato che per la diffusione dello stesso si è usato un rate molto alto (regola DIFFUSE con rate 10^6).

Successivamente viene sviluppato un nuovo servizio s2 in grado di servire le richieste con un rate superiore, 0.1 invece dello 0.05 di s1, cioè un servizio doppiamente efficiente nel servire le stesse richieste. Questo servizio viene iniettato in un nodo qualsiasi della rete con una concentrazione molto bassa (10 nell'esempio). L'evoluzione di questa situazione è mostrata in Figura 5, dove sulla sinistra sono raffigurate le concentrazioni di s1 in tutti i nodi della rete, a destra quelle di s2. Dalla situazione iniziale, appena descritta e riportata nella prima riga, si passa alla seconda riga di Figura 5, dove si nota che il servizio s2 comincia a diffondersi nella zona della rete centrata nel nodo in cui è stato iniettato e nella stessa area la concentrazione di s1 decresce. Questo comportamento, prevedibile da quanto visto nell'esempio riportato in Figura 4(d), porta, dopo circa 1500 unità di tempo e come riportato nella terza riga, s2 a prevalere su s1, fino a giungere ad una situazione completamente invertita: il servizio s1 si estingue e viene rimpiazzato da s2.

Gli esempi riportati sottolineano come in questo modello le tuple siano usate per reificare lo stato di ciascun individuo e, in un senso più esteso, come un set di tuple nella rete rappresenti lo stato corrente di un'ecologia, in base al quale il comportamento dei singoli individui computazionali è profondamente influenzato.

7 Conclusioni

Il modello di coordinazione descritto in queste pagine è fondato sull'ispirazione naturalistica; in particolare, se confrontato con soluzioni tradizionali basate su standard tuple space non deterministici, supporta naturalmente una significativa estensione di quelle proprietà di auto-organizzazione individuate in precedenza come caratteristiche chiave dei nuovi sistemi distribuiti e pervasivi. Esso può quindi essere la base su cui realizzare infrastrutture di coordinazione radicalmente nuove, in grado di rispondere alle esigenze di apertura e pervasività dei sistemi emergenti. Questa infrastruttura può essere vista come

una virtual machine distribuita, la quale attivi in ogni nodo della rete le leggi chimiche (realizzate come programmi dichiarativi) in grado di trasformare il sistema in un'ecologia di servizi, auto-organizzante, aperta e robusta ai cambiamenti.

Gli sforzi futuri, ancor prima di definire un supporto concreto allo sviluppo, individuare ed analizzare le problematiche relative alla sicurezza e prima di cercare strade possibili tramite le quali poter introdurre questo nuovo approccio in ambiente legacy, si concentreranno sull'implementazione di prototipi funzionanti. Come visto, candidato principe per l'infrastruttura di coordinazione è TuCSoN, tecnologia che mette a disposizione centri di tuple programmabili così da poter gestire, grazie alle primitive originarie arricchite dalla possibilità di definire politiche di coordinazione ad hoc, il concetto di concentrazione e i comportamenti auto-organizzanti guidati dalle regole chimiche.

E' infine doveroso sottolineare come in questo modello sia particolarmente critica la progettazione delle leggi e la scelta dei relativi rate, in quanto anche piccoli errori possono dare luogo a sistemi instabili e non utilizzabili: questo processo richiede quindi estese simulazioni su scenari noti ed inoltre è auspicabile una qualche forma di autoregolazione a run-time dei rate chimici, ad esempio in base al carico di sistema.

Riferimenti

1. Mirko Viroli, Matteo Casadei, Andrea Omicini, *A Framework for Modelling and Implementing Self-Organising Coordination*
2. Franco Zambonelli e Mirko Viroli, *Towards Eternally Adaptive Pervasive Service Ecosystems*
3. Mirko Viroli and Matteo Casadei, *Biochemical Tuple Spaces for Self-Organising Coordination*