

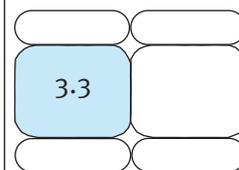


SISTEMI AUTONOMICI

STATO E PROSPETTIVE

La penetrazione della banda larga e l'aumento delle capacità di calcolo e di memoria dei terminali stanno creando le condizioni per lo sviluppo di reti di crescente complessità e pervasività e di soluzioni decentralizzate per l'offerta di servizi. Questa evoluzione richiede sistemi capaci di auto-adattarsi alla forte dinamicità del contesto e al tempo stesso di auto-gestirsi. Inoltre, determinerà l'evoluzione delle attuali catene del valore in ecosistemi di servizi, creando nuove opportunità di sviluppo. Nasce da queste premesse la sfida dei Sistemi Autonomici (*Autonomic Computing & Networking*).

Antonio Manzalini
Corrado Moiso



1. INTRODUZIONE

L' aumento della penetrazione della banda larga (sia fissa sia mobile) e la riduzione dei costi dell'hardware (per esempio memorie di massa, server ecc.) stanno creando già oggi le condizioni per lo sviluppo di sistemi decentralizzati. Tali sistemi permettono di abilitare una svariata gamma di applicazioni e servizi, come la diffusione e la condivisione di contenuti, le prestazioni per la memorizzazione o salvataggio in rete di dati personali, la virtualizzazione di PC (modello del *Network PC*), le applicazioni fornite come servizi in rete, come per esempio, secondo il paradigma *Software as a Service* ecc.. Anche lo sviluppo tecnologico dei terminali di Utente va nella stessa direzione: nei prossimi anni assisteremo ad un aumento sensibile delle capacità di calcolo, mentre le memorie raggiungeranno dimensioni dell'ordine dei Terabyte. Inoltre lo sviluppo della "Internet delle Cose", che permetterà di mettere in rete una vasta gamma di oggetti digitali (dai sensori agli elettrodomestici ecc.), determinerà un progressivo aumento della complessità della rete, anche in termini di gestione e configurazione.

In questo contesto evolutivo, è prevedibile che assisteremo allo sviluppo di "reti di reti" di crescente pervasività; le soluzioni decentralizzate saranno più efficaci di quelle di tipo tradizionale, basate su server centralizzati: ad esempio permetteranno una naturale scalabilità di rete (in termini di traffico, di distribuzione delle informazioni ecc.), una maggiore resistenza ai guasti, e un'ottimizzazione nell'utilizzo delle risorse. Questa crescente distribuzione e complessità delle reti del futuro richiederanno soluzioni capaci di auto-adattarsi alla forte dinamicità del contesto servizi e al tempo stesso di auto-gestirsi, con un limitato intervento umano.

Negli ultimi anni si sono affermate differenti soluzioni in questa direzione quali, per esempio:

1. i sistemi di *Grid Computing* e di *Cloud Computing* che offrono capacità di elaborazione e di memoria, distribuite e virtualizzate in rete;
2. i centri di servizio distribuiti realizzati dai fornitori di applicazioni Web2.0, tra cui Google, Yahoo, Amazon;
3. le soluzioni *Peer-to-Peer* (P2P) grazie alle quali è possibile realizzare servizi utilizzando

la potenza di calcolo e di memoria messa a disposizione dei nodi interconnessi tramite *reti overlay*, reti virtuali che astraggono le caratteristiche delle connessioni Internet secondo le specifiche esigenze applicative.

Il più grande successo (almeno dal punto di vista mediatico) della tecnologia *Peer-to-Peer* è la condivisione di dati, ma tale tecnologia comprende molti altri utilizzi campi di applicazione, tra cui servizi di comunicazione, di messaggistica e di gestione di reti sociali.

Questo articolo propone una visione secondo la quale l'integrazione e sinergia dei terminali d'Utenti e della sensoristica diffusa con i sistemi di rete porteranno alla creazione di "cloud" di risorse (di elaborazione, di memoria, di comunicazione, di dati ecc.) che si distribuiscono in maniera pervasiva e senza soluzione di continuità dai terminali ai nodi della rete, e ai centri di servizio. Inoltre, propone l'utilizzo di tecnologie e soluzioni autonome al fine di sviluppare soluzioni di elaborazione distribuita in grado di rispondere ai requisiti della crescente complessità ed eterogeneità delle reti del futuro in grado di migrare l'intelligenza e la condivisione delle risorse fino all'estrema periferia.

In particolare, il paragrafo 2 introduce alcuni concetti di base relativi alle tecnologie autonome per l'elaborazione di dati e per reti di comunicazione; il paragrafo 3 descrive alcuni possibili scenari applicativi; il paragrafo 4 presenta i prodotti commerciali attualmente disponibili ed illustra alcune iniziative di ricerca; il paragrafo 5 propone uno scenario innovativo di applicazione per l'abilitazione di ecosistemi di servizi, il paragrafo 6 riporta, infine, alcune considerazioni finali sullo sviluppo delle tecnologie autonome.

2. AUTONOMIC COMPUTING E NETWORKING

Il termine *Autonomic Computing* è stato coniato da IBM nel 2001 [1] con l'obiettivo di sviluppare soluzioni di elaborazione distribuita in grado di rispondere ai requisiti della crescente complessità ed eterogeneità dei sistemi informatici. La visione s'ispira alla metafora del sistema nervoso neuromotorio: in particolare alla sua funzione di regolare l'omeostasi dell'organismo attraverso meccanismi autonomi, mascherandone la complessità alla volontà cosciente.

I sistemi autonomi sono in grado di prendere decisioni in modo autonomo, costantemente controllando il loro stato e adattando il proprio comportamento al variare delle condizioni interne ed esterne di esecuzione, per esempio per migliorare le prestazioni, oppure per recuperare situazioni critiche o di errore.

Nella visione iniziale, la tecnologia autonoma era utilizzata per realizzare funzioni di supervisione di un sistema, le quali, però erano funzionalmente separate dal sistema stesso. Esempi di sistemi di supervisione autonoma di questo tipo sono quelli strutturati secondo il modello MAPE-K (*Monitor, Analyze, Plan, Execute - Knowledge*): il sistema sotto supervisione invia eventi che sono ricevuti, filtrati, correlati dal sistema di supervisione, il quale elabora, secondo logiche auto-adattative che operano secondo determinati piani di azione, dati sullo stato dei sistemi, e regole di gestione (Figura 1), e restituisce azioni correttive che devono essere eseguite dal sistema sotto supervisione. In questo modello il sistema gestito non ha un comportamento autonomo, in quanto tutta l'"intelligenza" autonoma risiede nel sistema di gestione [1].

I principi dell'*Autonomic Computing* sono stati ben presto estesi nell'ottica di sviluppare delle soluzioni di rete che manifestino capacità di auto-adattamento e auto-gestione; nasce così il termine *Autonomic Computing e Networking*.

L'astrazione delle risorse di elaborazione e rete attraverso moduli software autonomi permetterebbe la realizzazione di sistemi distribuiti capaci di rispondere meglio a requisiti di scalabilità e robustezza, rispetto alle soluzioni centralizzate. In particolare, un sistema autonomo distribuito può essere definito co-

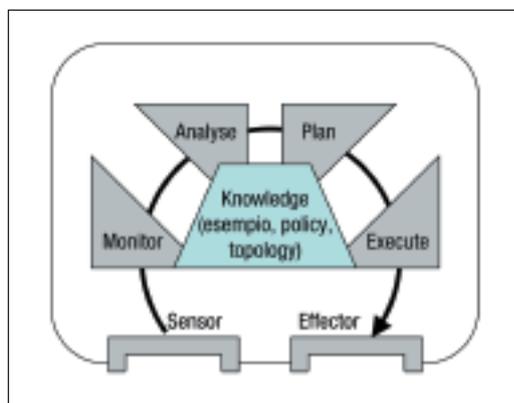


FIGURA 1

Il modello MAPE-K per la supervisione autonoma [1]

me un insieme di unità elementari (componenti) che interagiscono tra di loro e mutano il loro comportamento e le loro relazioni in base a regole codificate al loro interno.

In generale, il modello di un componente autonomo (AC) può essere schematizzato secondo due cicli di controllo (Figura 2): uno interno ed uno esterno. Il primo ha la funzione di garantire la gestione interna del componente (come la riconfigurazione in caso di guasto, l'ottimizzazione dei parametri di configurazione ecc.); il secondo abilita l'interazione con l'ambiente esterno garantendo l'adattamento al contesto. Un AC è, quindi, un'entità capace di raccogliere eventi ed auto-adattare il proprio comportamento secondo le variazioni del proprio stato interno e le interazioni con altri componenti, applicando le cosiddette logiche di *self-**, quali, per esempio, quelle di auto-configurazione, sopravvivenza, ottimizzazione e protezione, denominate in gergo *self-CHOP* (*Configuration, Healing, Optimization, Protection*), le quali, per esempio, permettono alle componenti di auto-adattarsi al fine di risolvere situazioni di errore, ottimizzare le prestazioni, o proteggersi da possibili eventi critici.

Un esempio di implementazione di AC è l'*Autonomic Communication Element* (ACE), sviluppato dal progetto europeo IST CASCADAS

[2, 3], il cui obiettivo era quello di sviluppare e sperimentare un prototipo di ecosistema di servizi. La figura 3 riporta la struttura interna di un ACE, le cui funzioni sono strutturate come un insieme di "organi". Il comportamento autonomo è realizzato essenzialmente dal *Facilitator*, il quale analizza i *Self-model*, automi che descrivono il comportamento del componente a fronte di specifici eventi, e li compone in piani di esecuzione.

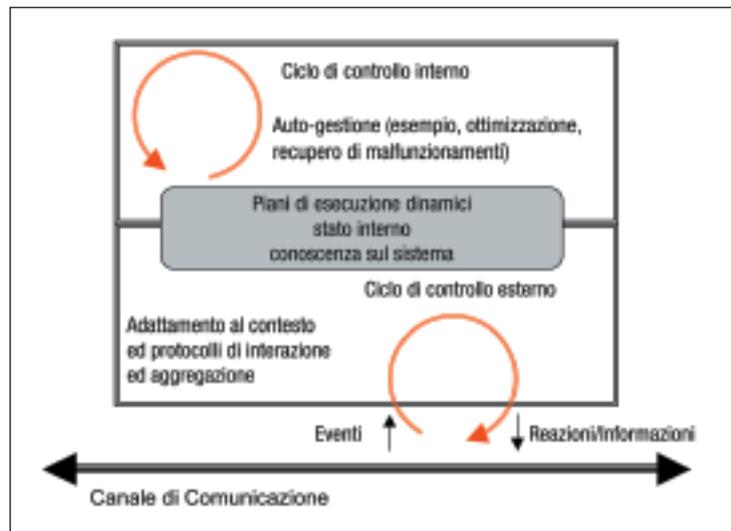


FIGURA 2
Vista logica di un componente autonomo

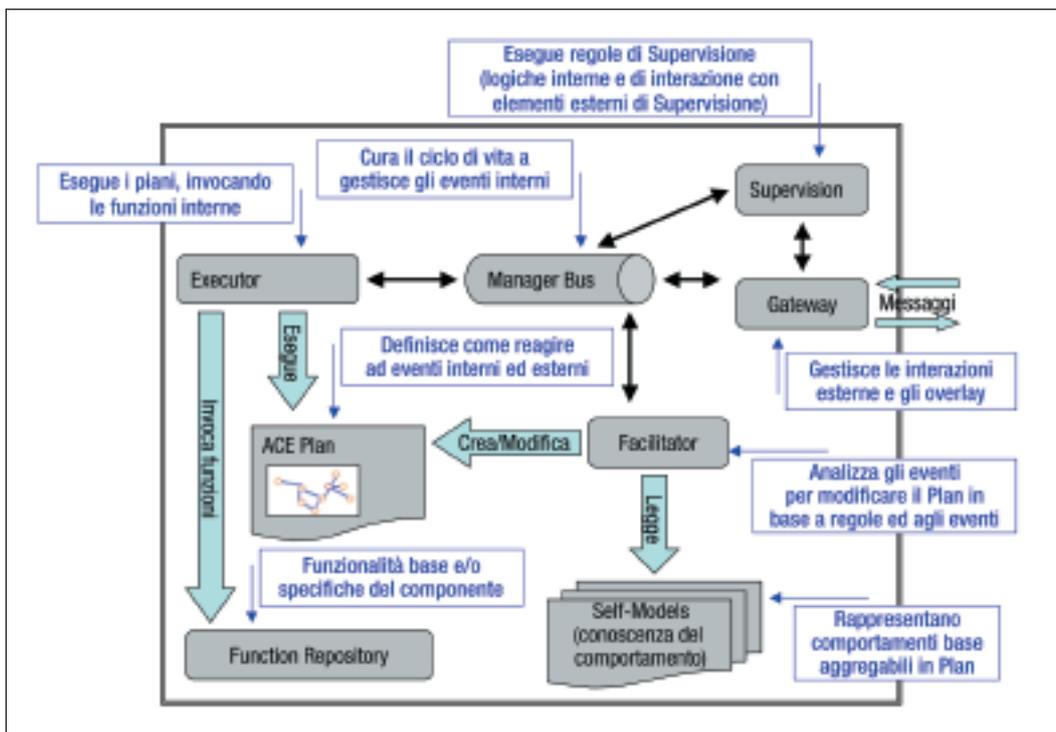


FIGURA 3
Struttura interna e funzioni di un ACE - Autonomic Communication Element [3]

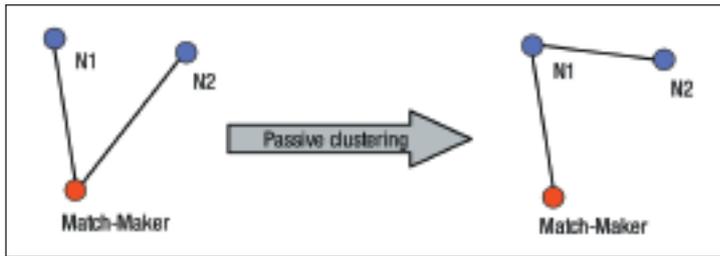


FIGURA 4

Modifica della rete overlay operata dall'algoritmo di "passive clustering"

In generale, gli AC implementano al loro interno semplici funzioni; l'aggregazione, possibilmente in modo autonomico, con altri AC permette lo sviluppo di funzioni più complesse e consente l'adattamento dinamico alle variazioni delle condizioni di esecuzione.

Al fine di soddisfare requisiti derivati dalla dinamicità e dalla scalabilità dei sistemi di componenti autonomici, le interazioni tra questi avvengono generalmente su reti *overlay*, in cui ogni componente interagisce con un numero limitato di componenti vicini. Sebbene i dettagli del modello d'interazioni tra componenti possano variare, si assume che la massima "distanza" (per esempio, in termini di nodi intermedi nella rete *overlay*) a cui due AC possono comunicare sia piccola rispetto alla dimensione dell'intero sistema. Questo garantisce la "località" della comunicazione tra i componenti, e quindi evita o limita il rischio di una generazione eccessiva di messaggi.

Sono stati anche definiti dei protocolli, detti di *gossiping*, ispirati alle tipiche interazioni tra entità biologiche, al fine di garantire la diffusione delle informazioni in un sistema distribuito autonomico, pur rispettando la località delle singole interazioni su una rete *overlay*. Analogamente a quanto succede nei sistemi biologici, ad ogni passo del protocollo di *gossiping*, un componente comunica una limitata quantità di dati con un limitato sottoinsieme di nodi nel sistema, in generale, tra i vicini in una rete *overlay*. Le informazioni ricevute dai vicini possono essere combinate dai componenti così da incrementare la conoscenza dello stato del sistema globale. Tramite tale soluzione si può, per esempio, calcolare in maniera completamente distribuita la media di un valore dello stato dei componenti [4].

Le reti *overlay* tramite cui avvengono le interazioni tra componenti possono essere crea-

te, mantenute, ed ottimizzate, per mezzo di algoritmi di auto-organizzazione [5], anche di ispirazione biologica.

La creazione di una rete *overlay* tra componenti può utilizzare protocolli di ricerca di servizi, basati sulla propagazione di richieste di servizi; un esempio è fornito dal protocollo *Goal Needed/Goal Achievable*, il quale permette di creare legami stabili tra un ACE che ha bisogno di un servizio e l'ACE che lo fornisce [3].

Un altro meccanismo di auto-organizzazione delle reti *overlay* è quello realizzato tramite gli algoritmi di aggregazione (*clustering*), i quali hanno l'obiettivo di favorire la creazione di aggregati tra componenti in base alle loro caratteristiche. Durante un processo di auto-aggregazione ogni nodo, le cui proprietà sono astratte da un concetto di tipo, rafforza i legami con altri nodi in base alle caratteristiche di questi ultimi (per esempio, se sono dello stesso tipo). Un esempio è fornito dall'algoritmo denominato *passive clustering*: due nodi dello stesso tipo sono invitati a creare un legame tra loro da un terzo nodo (detto *match-maker*) che ha legami con entrambi. In dettaglio, ad ogni ciclo di attività, ogni nodo ha una certa probabilità di assumere il ruolo di *match-maker* e di iniziare la procedura di modifica dei legami: il *match-maker* seleziona a caso due nodi vicini nella rete *overlay* (N1 e N2, nella Figura 4) e, nel caso in cui i loro tipi soddisfino la proprietà di aggregazione adottata li invita a creare un legame diretto; al fine di evitare una crescita delle relazioni tra i nodi, il *match-maker* può scegliere di rimuovere il legame con uno dei due nodi.

Un'altra classe di algoritmi basati sulle interazioni tramite una rete *overlay* sono gli algoritmi detti di auto-differenziazione, mediante cui è possibile sviluppare algoritmi decisionali completamente decentralizzati, soltanto tramite l'elaborazione e la modifica dello stato dei nodi vicini in una rete *overlay*. Gli algoritmi, per esempio, operano come una variante dell'algoritmo di *passive clustering*; ogni nodo ha una certa possibilità di attivarsi come *match-maker*; seleziona a caso due suoi vicini, reperisce i loro stati, li elabora e, in base al risultato, invia ai vicini selezionati istruzioni su come modificare il loro stato o come adattare il loro comportamento.

Grazie alla loro completa distribuzione, questi

meccanismi di auto-organizzazione, auto-differenziazione e di diffusione e combinazione di dati sono naturalmente scalabili, mostrano un'elevata affidabilità e capacità ad adattarsi ai cambiamenti che avvengono a livello del sistema globale.

La caratteristica fondamentale di queste reti di componenti è, infatti, il cosiddetto comportamento emergente: il termine indica l'emergere di un comportamento globale a livello di sistema, a partire dalle semplici funzioni che tali componenti implementano e dalle loro interazioni collettive; quanto sorprende di più è che non risulta immediato dare un modello di tale comportamento collettivo a partire da azioni individuali dei singoli componenti e dalle loro interazioni. È un po' come affermare che il sistema, nella sua totalità, risulta essere "qualcosa in più" della somma delle sue parti (ovvero vale il concetto della non linearità).

3. SCENARI APPLICATIVI

Le prime applicazioni delle tecnologie autonome hanno riguardato la gestione di applicazioni informatiche, al fine di rendere più efficiente il controllo del funzionamento di sistemi software sempre più sofisticati (e composti da differenti componenti distribuiti interagenti tra di loro), riducendo il coinvolgimento delle persone e velocizzando l'identificazione di eventuali situazioni anomale e dei corrispondenti correttivi.

Come menzionato in precedenza, secondo questo modello tutta l'"intelligenza" risiede nel sistema di supervisione, disaccoppiato dal sistema gestito. La piena potenza dell'approccio autonomo si ha, però, quando le prestazioni di tipo *self*-* sono realizzate all'interno dello stesso sistema e s'instaura una collaborazione tra tutti gli elementi autonomi.

Da questo punto di vista, l'Internet delle Cose rappresenta uno degli scenari applicativi più interessanti. Si sta osservando, infatti, un consenso generale nell'immaginare l'Internet del futuro come un'infrastruttura altamente pervasiva in grado di fornire servizi di comunicazione e scambio dati, contenuti per utenti, macchine e cose. La connettività, fornita da reti mobili e fisse, diventerà una risorsa di base abilitante l'aggregazione di ecosistemi. La comunicazione e l'interazione non avverranno più solamen-

te tra le persone ma si estenderanno anche verso l'ambiente. In particolare, lo sviluppo delle capacità di memorizzazione e di elaborazione dei terminali determinerà una significativa evoluzione delle architetture di rete.

La sfida principale, molto probabilmente, sarà capire le modalità secondo le quali gli attori (umani e non) della futura Internet delle Cose comunicheranno e interagiranno. Il forte sviluppo delle reti sociali sarà certamente confermato; anche le aziende utilizzeranno sempre più questa modalità di aggregazione per produzione, condivisione e gestione di conoscenza e innovazione, per interagire con i clienti e per creare e sviluppare nuovi ecosistemi economici. In questo contesto, l'Internet delle Cose, non potrà essere che un'infrastruttura neutrale e pervasiva a disposizione di tutti. Altro discorso per servizi e informazioni: diventerà necessario, e strategico, sviluppare adeguate piattaforme per la creazione e sviluppo di servizi e per la fruizione dell'informazione. Tali piattaforme dovranno avere requisiti di robustezza e protezione sia ai guasti sia alle frodi, garantendo al tempo stesso adeguati livelli di garanzia della riservatezza. È immediato intuire che questo futuro scenario di Internet delle Cose non solo richiederà una trasformazione delle attuali architetture di rete ma anche di strumenti adeguati per gestire la crescente complessità che lo caratterizza. In questo senso le tecnologie autonome sembra possano offrire efficaci soluzioni per la gestione e l'ottimizzazione delle risorse (inclusa l'energia).

In particolare, dal punto di vista della gestione, si potrebbe prevedere che ad ogni entità, risorsa o cosa sia associato un componente autonomo, capace di dotarla di prestazioni di auto-adattamento (esempio, *self-CHOP*): tale componente, tramite i cicli di controllo può intercettare eventi interni alla risorsa associata e/o provenienti da altri componenti autonomi, al fine di riconoscere situazioni che richiedono un intervento ed elaborare eventuali azioni correttive.

Le soluzioni autonome possono gestire, secondo il traffico o le situazioni di guasto, la configurazione di servizi e degli utenti, allocando le risorse necessarie, replicando logiche di servizio e contenuti, e distribuendo i carichi di lavoro. L'elaborazione autonoma di

tali azioni risulta particolarmente utile nel caso dei nuovi servizi di telecomunicazioni, ICT e Internet, i quali prevedono tra gli elementi di servizio e gli utenti un'elevata interattività e negoziazioni dinamiche, anche tra componenti gestiti da diversi provider fornitori di servizi. Per esempio, si potrebbero adottare soluzioni basate su aste e teoria dei giochi, per realizzare politiche di negoziazione ed allocazione delle risorse [6, 7].

Un secondo ambito d'applicazione è quello della gestione dei cosiddetti *cloud*, gruppi condivisi di risorse elaborative, applicative, comunicative o di memorizzazione. Esempi di tali *cloud* possono essere le soluzioni di *GRID* o *Cloud Computing*, oppure le reti overlay create da applicazioni *Peer-to-Peer*. Tali sistemi, soprattutto quelli che coinvolgono risorse dispiegate sui terminali d'utente, possono essere caratterizzati da un'elevata distribuzione e dinamicità delle risorse che partecipano alla condivisione, così da impedire la realizzazione di sistemi di supervisione (logicamente) centralizzati. Le risorse possono essere arricchite con componenti autonomici, i quali, tramite l'interazione con le risorse "vicine" (per esempio, implementando algoritmi di auto-differenziazione) possono identificare azioni adattative e/o correttive o attuare criteri di ottimizzazione. Tramite tali meccanismi si possono realizzare bilanciamento di carico, recupero da situazioni d'errore, oppure politiche di ottimizzazione nell'uso di risorse, per esempio, dal punto di vista del risparmio energetico [8].

Un altro campo d'utilizzo di soluzioni autonome è quello della supervisione delle cosiddette *Home Network* [9], infrastrutture che stanno diventando sempre più presenti nelle nostre case, non solo per collegare i sistemi che richiedono di essere interconnessi (esempio, PC, stampanti, *access gateway*, e *set-top box* per TV), ma anche nell'ottica di una sempre maggiore automatizzazione dei tradizionali elettrodomestici e di nuovi dispositivi (esempio, sensori). Di solito gli utenti hanno un limitato supporto nella realizzazione e nella gestione di una *Home Network*, per esempio per integrare e configurare un nuovo dispositivo oppure per riconoscere e risolvere un malfunzionamento. I dispositivi della *Home Network* potrebbero essere dotati di capacità autonome, attraverso le quali possono adattare il

loro comportamento al contesto di utilizzo (per esempio, considerando la connettività disponibile, la disponibilità o il costo dell'energia elettrica) oppure auto-configurarsi (adattando i parametri, aggiornando il *firmware* ecc.) quando sono introdotti in una rete. Sistemi di supervisione autonomici dispiegati, per esempio sugli *access gateway*, potrebbero riconoscere e risolvere eventuali malfunzionamenti realizzare interventi di ottimizzazione, oppure contattare logiche di gestione superiori fornite da un *Service Provider*, o facilitare l'interazione dell'Utente con un manutentore esterno fornendo tutti i dati necessari.

4. PRIMI PRODOTTI COMMERCIALI E STATO DI AVANZAMENTO DELLA RICERCA

Prerequisito indispensabile per la realizzazione di questi scenari è la disponibilità di ambienti per lo sviluppo di sistemi autonomici. Alcuni prodotti sono già disponibili, altri, sviluppati in contesti di progetti di ricerca, sono ancora allo stato di prototipo.

Un esempio di prodotti industriali è fornito dalle applicazioni ideate da IBM, tra cui la tecnologia alla base di Blue Gene/L [10], o gli strumenti per la realizzazione di sistemi autonomici per la gestione di ambienti informatici [1], dalle singole applicazioni ai complessi centri di servizio. Le caratteristiche autonome di tali sistemi sono principalmente orientate a facilitare gli operatori nel determinare situazioni di errore ed identificarne le cause o a semplificare aggiornamenti e configurazioni del software. In generale, questi sistemi di gestione sono strutturati a due livelli: un insieme d'elementi con proprietà di auto-gestione, per esempio secondo il modello MAPE-K descritto in precedenza, controllano uno o più risorse informatiche simili (esempio, stampanti, server, o applicazioni), e sono a loro volta supervisionati da uno o più sistemi, che realizzano prestazioni di *self-CHOP* a livello dell'intero sistema informatico. Tali sistemi sono realizzati conformemente a standard emergenti definiti anche considerando requisiti per i sistemi autonomici, quali, per esempio, *Web Service Distributed Management* specificato dall'ente di standardizzazione OASIS.

Analoghe motivazioni hanno ispirato molte

attività nel campo dei *Multi-Agent Systems* (MAS): un esempio è costituito dalle piattaforme di sviluppo ad agenti autonomici di *Whitestein Technologies* [11].

I sistemi MAS si basano sulla modellazione del comportamento di un sistema attraverso una collezione di agenti autonomi, eventualmente eterogenei, che mirano a risolvere un dato problema e/o che competono per massimizzare il vantaggio individuale. Il classico modello cognitivo su cui si basano gli agenti è il BDI (*Beliefs-Desires-Intentions*, cioè Convinzioni-Desideri-Intenzioni). Esso considera da una parte l'insieme delle credenze (*Beliefs*) dell'agente sull'ambiente in cui si trova ad operare, che sono il risultato delle sue conoscenze e delle sue percezioni, e dall'altra parte un insieme di obiettivi (*Desires*). Intersecando questi due insiemi, si ottiene un nuovo insieme di intenzioni (*Intentions*), che in seguito possono tradursi direttamente in azioni.

Queste soluzioni tuttavia sono state tipicamente concepite per architetture server centralizzate (o basate su *cluster*) e quindi non manifestano le caratteristiche di decentralizzazione indispensabili per garantire un'elevata penetrazione e scalabilità, come per esempio per l'Internet delle Cose.

Esempi di applicazioni distribuite dove si registra la prima introduzione di prestazioni autonome sono quelle strutturate secondo il paradigma *Peer-to-Peer* (P2P). Tali applicazioni si basano sul principio che ogni nodo che partecipa ad un servizio contribuisce, in maniera paritetica con gli altri nodi, alla fornitura del servizio stesso, o di alcune sue funzionalità; i nodi interagiscono con gli altri nodi tramite una rete *overlay*, che virtualizza la sottostante infrastruttura di comunicazione, per esempio Internet. L'approccio P2P è molto flessibile ed è utilizzato per realizzare differenti tipologie di servizi [12], quali *Gia* per la condivisione di contenuti, *VidTorrent* per la distribuzione di video in streaming, *SETI* per la condivisione di capacità elaborativa, e *Skype* per i servizi di telefonia. Oltre a specifici servizi, inoltre, l'approccio P2P è stato adottato per realizzare, in maniera completamente distribuita, prestazioni elaborative di base come, per esempio, le soluzioni *DHT* (*Distributed Hash Table*), le quali organizzano in maniera decentralizzate basi di dati di grandi dimensioni, o *JXTA* che realizza un'infra-

struttura completamente decentralizzata per lo scambio di messaggi.

In generale i nodi che partecipano ad un'applicazione P2P con caratteristiche autonome che realizzano prestazioni di tipo *self**, tramite cui si auto-gestiscono ed auto-configurano la rete *overlay* [13]. In particolare, tali prestazioni sono attivate quando i nodi entrano ed escono dalla rete. La rete *overlay* deve essere opportunamente riconfigurata, al fine di garantire una connettività adeguata ai nodi rimanenti, e i nodi devono modificare il loro comportamento e stato interno per adattarsi alla nuova configurazione o per recuperare eventuali situazioni critiche determinate dall'uscita di qualche nodo: per esempio, all'interno di una *DHT*, i nodi devono recuperare e ridistribuire i dati in carico ai nodi usciti. Inoltre, il comportamento auto-adattativo dei nodi può permettere di ottimizzare le prestazioni complessive applicando, per esempio, politiche di bilanciamento del carico, oppure riconfigurando le connessioni della rete *overlay* secondo le caratteristiche di Qualità di Servizio dei sottostanti canali di comunicazione. Queste proprietà rendono le soluzioni di tipo P2P maggiormente affidabili e scalabili rispetto alle soluzioni *client-server*, dove la logica di servizio è localizzata in pochi centri di servizio. Le prestazioni autonome possono altresì essere adottate per implementare prestazioni aggiuntive all'interno di un'applicazione P2P. Per esempio possono essere utilizzate per realizzare controlli di protezione, cercando di identificare possibili attacchi da nodi "maliziosi", oppure politiche per incentivare la cooperazione, scoraggiando i comportamenti parassitari dei nodi [14].

Nell'ambito della ricerca, diversi progetti internazionali stanno studiando dagli ambienti di servizi altamente decentralizzati e dotati di capacità di adattamento ed auto-organizzazione [15, 16]. Si tratta, tuttavia, per lo più di studi teorici, di simulazioni o di prototipi ancora lontani dallo sviluppo di reali applicazioni industriali. I progetti *Service Clouds* e *SwarmlingNets* costituiscono due esempi che cercano di applicare i principi di auto-organizzazione in piattaforme applicative distribuite. *Service Clouds* [17] è un ambiente per la prototipazione rapida e il dispiegamento di servizi distribuiti anche in ambienti mobili; *Service*

Clouds applica i principi di auto-organizzazione a livello di controllo di rete, al fine di ottimizzare la creazione di *overlay* e la trasmissione di dati, considerando contemporaneamente sia aspetti protocollari sia applicativi.

SwarmingNets [18], invece, si focalizza principalmente sugli aspetti di gestione di servizi distribuiti dispiegati su dispositivi mobili interconnessi tramite reti configurate dinamicamente: per ottenere soluzioni che offrono i livelli di adattamento, scalabilità e mobilità richiesti da tale contesto, *SwarmingNets* adotta meccanismi ispirati alla biologia, in particolare al mondo delle colonie di insetti, quali per esempio una strutturazione gerarchica auto-organizzata dei servizi e l'auto-adattamento dei servizi alle risorse disponibili nell'ambiente.

Un'altra interessante area di studi complementare è quella che investiga l'adozione di algoritmi di auto-organizzazione per realizzare piattaforme decentralizzate di servizi dotate di caratteristiche di autonome. Per esempio nel progetto "SESAM" [19] è stata sviluppata un'architettura che offre meccanismi d'interazione e coordinamento tra un insieme dinamico di nodi interconnessi tramite una rete *overlay*. SESAM in questo modo cerca di estendere il tipico approccio delle "Service Oriented Architecture", distribuendo l'implementazione di un servizio su tali reti di nodi, così da ottenere soluzioni maggiormente robuste e con un elevato grado di adattabilità ai requisiti prestazionali ed alle variazioni di traffico.

Invece, le attività condotte al fine di introdurre capacità autonome nelle piattaforme di *GRID Computing* (tra cui NextGRID, ASG, Accord) sono molto focalizzate sulla composizione dinamica e sulla gestione delle risorse di elaborazione, memoria e comunicazione. Tali soluzioni, le quali adottano un'architettura piatta (non *client-server*) basata sulla decentralizzazione delle risorse, non considerano però le implicazioni dei livelli superiori di servizio, contesto e gestione dei dati. Un esempio è fornito dal progetto AutoMate, il cui principale obiettivo è stato sviluppare una soluzione per applicazioni GRID autonome. L'architettura proposta [20] è basata su tre livelli (sistemi, componenti e applicazioni) sviluppati come insiemi di agenti. In particolare, i servizi e le applicazioni dell'ambiente Au-

toMate sono strutturati in componenti autonomi, dispiegati sulle risorse (livello di sistema) e gestiti da altri agenti (con motori basati su regole logiche di primo ordine).

L'ambiente Autonomia [21], invece, fornisce un insieme di servizi programmabili per realizzare, secondo principi autonomi, la gestione e la configurazione di applicazioni, come la registrazione, il reperimento delle risorse e dei servizi necessari, l'ottimizzazione delle prestazioni e il riconoscimento e la risoluzione di situazioni di errore. L'ambiente permette di associare ad ogni funzionalità o proprietà di un'applicazione un comportamento autonomo, strutturato in tre moduli: "Controllo", che ha riconosce gli eventi che evidenziano una potenziale deviazione del sistema da una situazione normale, "Analisi e Verifica", che identifica il tipo di deviazione, ed "Adattamento", che ha il compito di costruire un opportuno piano di intervento.

Anche i risultati del progetto IST CASCADAS (*Component-ware for Autonomic, Situation-aware Communications And Dynamically Adaptable Services*) [2], di cui Telecom Italia è stato Primo Contraente, si collocano in questo contesto di ricerca. L'obiettivo principale di CASCADAS è stato sviluppare un ecosistema decentralizzato di rete e servizi basato su componenti autonomi che virtualizzano le risorse di rete e che interagiscono secondo algoritmi di auto-organizzazione per espletare determinate funzionalità (tra cui esecuzione distribuita, gestione e controllo della piattaforma, gestione delle basi di conoscenza ecc.). Ogni risorsa ed "oggetto" dell'ecosistema è potenzialmente in grado di collegarsi alla rete, per erogare e/o fruire di qualunque servizio e/o informazione, interagendo con altre risorse ed "oggetti" (conformemente alla visione della "Internet delle Cose"). Questa visione del progetto risulta innovativa rispetto allo stato di avanzamento delle soluzioni commerciali e della ricerca, in quanto propone di integrare le caratteristiche dei Sistemi Complessi Adattativi (*Complex Adaptive Systems - CAS*) e dell'*Autonomic Computing e Networking*; questo approccio offrirebbe l'opportunità di abilitare nuovi modelli di business e di facilitare la convergenza dei paradigmi di gestione di rete e servizi, presupposti necessari allo sviluppo dell'Internet delle Co-

se. CASCADAS ha realizzato un ambiente di sviluppo e di esecuzione per componenti autonomi distribuiti, disponibile come open source [22]; tale ambiente è stato utilizzato per dimostrare sperimentalmente fattibilità di ecosistemi di servizi. Un'altra caratteristica distintiva dell'ambiente CASCADAS è costituita dal fatto che il componente autonomo non è un agente, nella classica accezione MAS, ma un aggregato di organi (Figura 3), seppur sviluppati a partire da agenti di minime dimensioni, in grado di svolgere determinate funzioni. La praticità e la semplicità di esecuzione dell'ambiente ne consente l'utilizzo anche su cellulari, ed eventualmente anche su micro dispositivi: anche questo rappresenta un elemento distintivo rispetto alle altre soluzioni precedentemente illustrate.

Anche nell'ambito delle attività di standardizzazione si sta inoltre osservando un crescente interesse per le tecnologie autonome. In particolare, l'*Autonomic Communication Forum* (ACF) si pone l'obiettivo di definire un modello architetturale di riferimento ed un insieme di specifiche per l'interoperabilità; inoltre, la recente definizione del gruppo ETSI *Autonomic Network Engineering for the Self-Managing Future Internet* mira ad un'architettura di rete autonoma per l'Internet del Futuro, che abbia spiccate capacità di auto-gestione.

5. PROSPETTIVE FUTURE

Nel lungo termine, l'applicazione delle tecnologie autonome potrebbe costituire uno dei fattori abilitanti di nuovi scenari applicativi e modelli di business. Gli ecosistemi, che si creano intorno ad un'innovazione tecnologica, possono rappresentare potenziali opportunità di sviluppo per gli operatori, nella misura in cui questi riescano a ritagliarsi un ruolo e inserirsi in un modello di business vantaggioso. In questo senso ci sono diverse aree nelle quali sono stati raggiunti dei rapporti prestazione/costo tali da rendere possibile nuovi modelli di business. Per esempio, le offerte gratuite di Amazon e Google di memoria in rete sono la conseguenza del costo di memorizzazione che ha subito una forte riduzione, e del minimo costo per trasmettere le informazioni in rete. La migrazione di questi stessi fattori verso il bordo della rete, favoriti dall'au-

mento della capacità di memoria e di esecuzione dei terminali d'utente, potrebbe abilitare nuovi Eco-Sistemi di Servizi (ESS) con la partecipazione attiva degli Utenti [23, 24].

Un ESS è una piattaforma aperta per servizi, per esempio per l'Internet delle Cose, basata su un modello di servizi a componenti, usati, ad esempio, per fornire astrazioni di servizi e funzioni o accessi a dati e contenuti (Figura 5), anche a livello di terminali di Utenti. In questo senso un ESS avrebbe l'obiettivo di creare complesse relazioni di utilizzo-fornitura di servizi tra differenti tipologie di attori: gli attori coinvolti non sono solo i tradizionali fornitori di servizi (operatori di telecomunicazione oppure fornitori di servizi su Internet), o PMI, ma anche, e in un certo senso, soprattutto Utenti che assumono contemporaneamente il ruolo di utilizzatori, creatori e fornitori di (componenti di) servizio e di contenuti (i cosiddetti *prosumer*). ESS deve permettere a questi attori di interagire, in modo sicuro ed affidabile, per vendere, comprare, negoziare l'uso di contenuti, servizi e componenti di servizio, inclusi quelli forniti da infrastrutture di Telecomunicazione e ICT. I servizi possono essere creati e offerti da tutti gli attori, compresi gli Utenti/*prosumer*, aggregando componenti di servizio atomici (esempio, funzioni elementari, servizi generici di Telecomunicazione o ICT, quali messaggistica, localizzazione, controllo di chiamate multimediali) e servizi base della piattaforma, quali quelli per dotare i servizi di prestazioni di auto-adattamento e auto-gestione.

La piattaforma dovrebbe fornire prestazioni base per controllarne l'evoluzione e la sopravvivenza (analogamente alle leggi necessarie per proteggere gli ecosistemi del mondo reale); esempi sono servizi per:

- garantire l'affidabilità del mercato, proteggere la differenziazione delle offerte, evitando posizioni dominanti;
- assicurare i fornitori di servizi sulla protezione dei loro contenuti e servizi, esempio, riservatezza dei dati sensibili, diritti d'uso, *accounting* affidabile sull'uso delle risorse necessari per abilitare politiche di remunerazione;
- proteggere le risorse in comune, cioè le risorse "ambientali" condivise che abilitano il mercato e sono usate da tutti gli attori (esempio: banda, potere elaborativo, memoria condivisa).

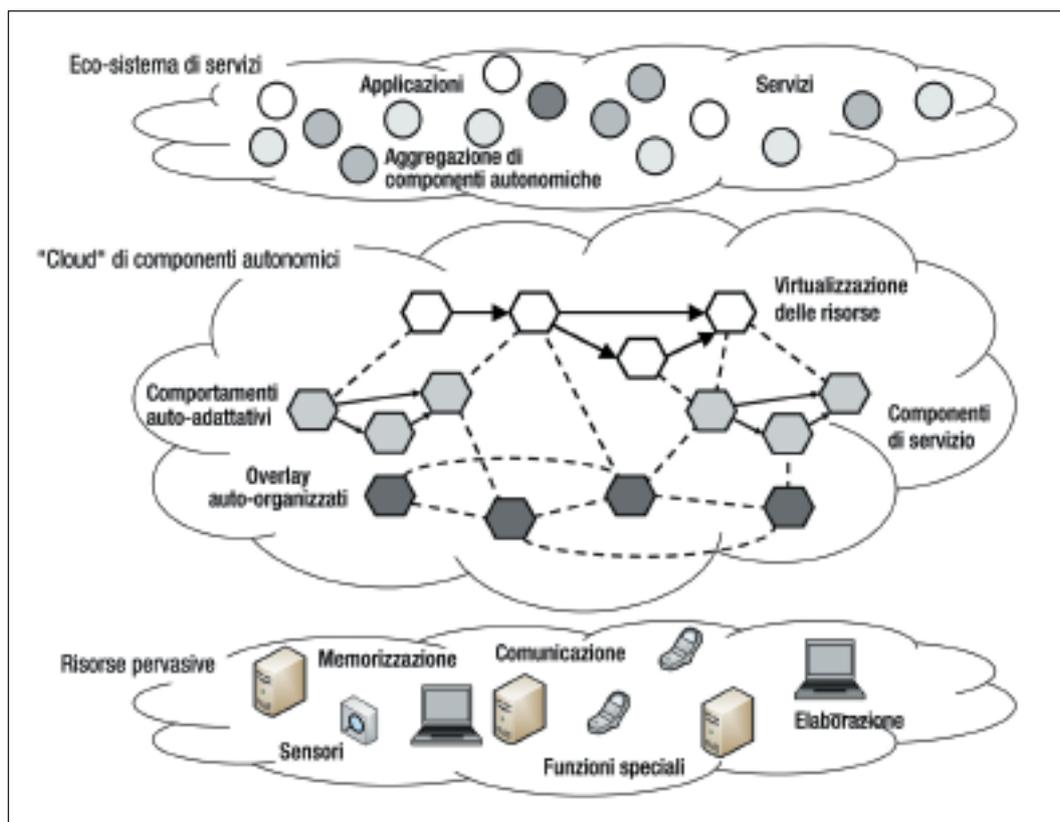


FIGURA 5
Eco-sistema di servizi

Un tale ESS è un ambiente altamente dinamico, distribuito geograficamente con un'elevata evoluzione in termini di attori e servizi, pertanto risulta difficile pensare di poterlo governare utilizzando sistemi tradizionali di gestione di servizi ed infrastrutture ed adottando soluzioni (logicamente) centralizzate. D'altra parte, è facile notare che il modello a componenti autonomi distribuiti è un elemento abilitante la realizzazione delle prestazioni richieste ad un ESS. Per esempio:

- i servizi e i componenti di servizio possono essere definiti in termini di raggruppamenti di componenti autonomi;
- l'autonomia della piattaforma e del modello a componenti permette di realizzare le prestazioni di auto-adattamento e di auto-gestione richieste ai servizi per potere operare nel contesto dinamico di un ESS;
- ambienti di sviluppo possono essere proposti ed adottati dai diversi attori, anche in maniera competitiva, per agevolare lo sviluppo dei servizi;
- servizi di supervisione più complessi, compresi servizi di accounting e di tariffazione, possono essere realizzati come aggregazio-

ne di servizi base e possono, quindi, essere soggetti a "commercializzazione" come tutti gli altri.

Inoltre, le prestazioni autonome permettono di realizzare politiche globali per la protezione delle risorse condivise e per evitare comportamenti dominanti di singoli attori. Per esempio, esse possono contribuire al mantenimento delle prestazioni dell'eco-sistema, che possono essere definite dai seguenti 4 fattori:

- creazione del valore (creazione di "nicchie"): la creazione di nicchie è fondamentale per la creazione del valore di un ESS e per la sua differenziazione; in particolare permette di realizzare servizi specializzati, fondamentali per affrontare la cosiddetta *long-tail*; la creazione di nicchie è favorita da un basso livello di entrata nell'ESS garantita dalla capacità della piattaforma di offrire e condividere risorse di elaborazione e di memoria; inoltre, la sopravvivenza delle nicchie è garantita da politiche di protezione realizzate a livello di piattaforma;
- capacità di gestire la crescita dell'infrastruttura (robustezza e scalabilità): le prestazioni autonome della piattaforma possono realizzare politiche di riscontro e recupero

degli errori, bilanciamento del carico e politiche adattative dinamiche; la robustezza e la scalabilità sono, inoltre, garantite dalla architettura decentralizzata e dai comportamenti auto-adattativi dei singoli componenti;

□ miglioramento della produttività: la produttività nello sviluppo dei servizi è favorita da modelli a componenti che offrono potenti livelli di astrazioni e dalla facilità di aggregazione ed uso dei servizi base; le soluzioni autonome forniscono un supporto, in quanto hanno l'obiettivo di spostare il peso della realizzazione delle funzioni di gestione dai programmatori alla piattaforma; inoltre, protocolli di auto-organizzazione aiutano nell'aggregazione dei componenti e nel gestire le relazioni/contratti tra questi a tempo di esecuzione;

□ evoluzione ed ottimizzazione: come indicato nel paragrafo precedente per la gestione delle *cloud* di risorse, le prestazioni autonome dei componenti e la realizzazione di algoritmi di controllo distribuiti basati su protocolli di auto-organizzazione permettono di realizzare politiche di controllo ed ottimizzazione che combinano i vantaggi degli approcci *top-down* (esempio, guidati da regole) con quelli degli approcci *bottom-up* (esempio, guidati da logiche di auto-adattamento dei componenti); esempi di tali ottimizzazioni riguardano l'allocazione e la negoziazione delle risorse, l'applicazione di politiche di ottimizzazione energetica, oppure di ottimizzazione delle prestazioni tramite distribuzione del carico.

6. CONCLUSIONI

La progressiva penetrazione della banda larga, l'aumento delle capacità di calcolo e di memoria dei dispositivi di Utente (per esempio i cellulari, o i PDA) e l'evoluzione verso "Internet delle Cose" determineranno lo sviluppo di "reti di reti" di crescente complessità e pervasività. Questo sviluppo richiederà sistemi e soluzioni capaci di auto-adattarsi alla forte dinamicità del contesto e al tempo stesso di auto-gestirsi (con un limitato intervento umano). La sfida tecnologica dell'*Autonomic Computing & Networking* ha l'obiettivo di soddisfare questi requisiti emergenti.

In particolare questo articolo ha proposto l'utilizzo di tecnologie e soluzioni autonome proprio con l'obiettivo di sviluppare soluzioni

di elaborazione distribuita in grado di rispondere ai requisiti della crescente complessità ed eterogeneità delle reti del futuro in grado di spingere la condivisione delle risorse fino all'estrema periferia.

In particolare, ha proposto una visione secondo la quale l'integrazione e sinergia dei terminali d'utente e della sensoristica diffusa con i sistemi di rete porteranno alla creazione di "cloud" di risorse (di elaborazione, di memoria, di comunicazione, di dati ecc.) distribuite in maniera pervasiva e senza soluzione di continuità dai terminali ai nodi della rete ed ai centri di servizio.

In sintesi, l'analisi fin qui condotta suggerisce che le tecnologie autonome permettono lo sviluppo di interessanti soluzioni che soddisfano, in linea di principio, i principali requisiti di questa evoluzione delle Telecomunicazioni e di Internet.

A fronte di alcune prime soluzioni commerciali proprietarie, si osserva sia un intenso sforzo di ricerca e sviluppo (per esempio, nell'ambito delle collaborazioni europee finanziate nel Settimo Programma Quadro) ma anche un crescente interesse nelle attività di standardizzazione (per esempio, in ACF ed ETSI).

Nel lungo termine, l'applicazione delle tecnologie autonome potrebbe costituire uno dei fattori abilitanti per la realizzazione di nuovi ecosistemi di servizi aperti alla partecipazione attiva di un'ampia gamma di attori, dagli operatori ai fornitori di servizi, alle imprese e agli utenti.

Bibliografia

- [1] IBM: *An architectural blueprint for autonomic computing*. http://www01.ibm.com/software/tivoli/autonomic/pdfs/AC_Blueprint_White_Paper_4th.pdf (giugno 2006).
- [2] CASCADAS Project, CASCADAS Web site, www.cascadas-project.org.
- [3] Manzalini A., Zambonelli F., Baresi L., Di Ferdinando A.: *The CASCADAS Framework for Autonomic Communications*. In: A. Vasilakos, et al. (Eds.): "Autonomic Communication", Springer book, 2009.
- [4] Jelasity M., Montresor A., Babaoglu O.: Gossip-Based Aggregation in Large Dynamic Networks. *ACM Transactions on Computer Systems*, Vol. 23, n. 3, agosto 2005, p. 219-252.
- [5] Saffre F., Tateson R., Halloy J., Shackleton M., De-neubourg J.L.: Aggregation Dynamics in Overlay Networks and Their Implications for Self-Organized

zed Distributed Applications. *The Computer Journal*, febbraio 2008.

- [6] Magrath S., Chiang F., Markovits S., Braun R., Cuervo F.: *Autonomics in telecommunications service activation*. In: Proc. Autonomous Decentralized Systems, ISADS 2005, aprile 2005, p. 731-737.
- [7] Zheng Y., Feng Z.: Evolutionary game and resources competition in the Internet. *Modern Communication Technologies*, 2001, p. 51-54.
- [8] Deussen P., Ferrari L., Manzalini A., Moiso C.: *Highly Distributed Supervision for Autonomic Networks and Services*. In: Proc. 5-th Advanced International Conference on Telecommunications, AICT2009, maggio 2009.
- [9] Manzalini A., Minerva R., Moiso C.: *Autonomics in the home*. In: Proc. ICIN2009, ottobre 2009.
- [10] Chiu G.L.-T., Gupta M., Royyuru A.K. (Eds.): *Blue Gene*. *IBM Journal of Research and Development*, Vol. 49, n. 2/3, 2005.
- [11] Calisti M., Greenwood D.: *Goal-Oriented Autonomic Process Modeling and Execution for Next Generation Networks*. In: Proc. 3-rd IEEE International Workshop on Modelling Autonomic Communications Environments, LNCS 5276, settembre 2008, p. 38-49.
- [12] Milojicic D.S.: *P2P Computing*. HP Laboratories Palo Alto Technical Report n. HPL-2002-57, http://www.hpl.hp.com/techreports/2002/HP_L-2002-57R1.pdf.
- [13] Leitao J., Rosa L., Rodrigues L.: *Large-Scale Peer-to-Peer Autonomic Monitoring*. In: Proc. IEEE Globecom Workshops, 2008, p. 1-5.
- [14] Kwok Y.-K.: *Autonomic Peer-to-Peer Systems: Incentive and Security Issues*. In: M.K. Denko et al. (eds.): "Autonomic Computing and Networking", Springer, 2009, p. 205-236.
- [15] Babaoglu O., Canright G., Deutsch A., Caro G.A.D., et al.: Design Patterns from Biology to Distributed Computing. *ACM Transaction on Autonomous and Adaptive Systems*, Vol. 1, n. 1, 2007, p. 22-66.
- [16] Mamei M., Menezes R., Tolksdorf R., Zambonelli F.: Case Studies for Self-organization in Computer Science. *Journal of Systems Architecture*, Vol. 52, Issues 8-9, agosto 2006, p. 443-460.
- [17] Samimi F.A., McKinley P.K., Sadjadi S.M., Tang C., Shapiro J.K., Zhou Z.: *Service Clouds: Distributed Infrastructure for Adaptive Communication Services*. *IEEE Transactions on Network and System Management*, Special Issue on Self-Managed Networks, Systems and Services, settembre 2007, p. 84-95.
- [18] Chiang F., Braun R.: *A Nature Inspired Multi-Agent Framework for Autonomic Service Management in Pervasive Computing Environments*. In: Proc. 10-th IEEE/IFIP Network Operations and Management Symposium, NOMS2006, aprile 2006, p. 1-4.
- [19] SESAM Project: *Self Organization and Spontaneity in Liberalized and Harmonized Markets*. <http://dsn.tm.uni-karlsruhe.de/english/sesam-project.php>.
- [20] Agarwal M., Bhat V., Matossian V., Putty V., Schmidt C., Zhang G., Zhen L., Parashar M., Khargharia M., Hariri S.: *AutoMate: enabling autonomic applications on the grid*. In: Proc. Workshop on Autonomic Computing, 2003.
- [21] Dong X., Hariri S., Xue L., Chen H., Zhang M., Pavuluri S., Rao S.: *Autonomia: an autonomic computing environment*. In: Proc. IEEE International Conference on Performance, Computing, and Communications, 2003.
- [22] CASCADAS Project: *ACE Toolkit open source*. <http://sourceforge.net/projects/acetoolkit/> gennaio 2009.
- [23] Alfano R., Manzalini A., Moiso C.: *Distributed Service Framework: an innovative open eco-system for ICT/Telecommunications*. In: Proc. Autonomics 2007, ottobre 2007.
- [24] Deussen P., Höfig E., Manzalini A.: *An Ecological Perspective on Future Service Environments*. In: Proc. 2-nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, 2008, p. 37-42.

ANTONIO MANZALINI, laureato in Ingegneria Elettronica al Politecnico di Torino, è entrato in Telecom Italia nel 1990. Ha partecipato a diversi progetti internazionali di ricerca finanziati dalla Comunità Europea, riguardanti architetture di rete e piattaforme servizio, occupando varie posizioni di responsabilità. Ha partecipato in ITU-T ed ETSI a molte attività di standardizzazione nelle telecomunicazioni. Attualmente fa parte del Future Centre di Telecom Italia dove si occupa di architetture di rete e servizi altamente distribuite (quali Autonomic Networking e Cloud Computing), abilitanti ecosistemi di TLC e l'Internet delle Cose. Nel 2008 ha conseguito la certificazione internazionale PMI come Project Manager. È autore di molte pubblicazioni, di un libro e di cinque brevetti su soluzioni di reti e servizi. E-mail: antonio.manzalini@telecomitalia.it

CORRADO MOISO, si è laureato in Scienze dell'Informazione all'Università di Torino nel 1984. Dallo stesso anno lavora in TILAB (ex CSELT). Inizialmente ha studiato linguaggi logici e funzionali, l'elaborazione distribuita ad oggetti ed il loro uso in TMN. Dal 1994, ha investigato l'introduzione di tecnologie informatiche nell'Intelligenza di Rete, contribuendo alla definizione di TINA, allo standard Parlay ed all'introduzione di SOA in piattaforme di servizio. Attualmente opera presso Future Centre di Telecom Italia, dove investiga l'adozione di architetture decentralizzate e tecnologie autonomiche nelle infrastrutture ICT. Ha collaborato a progetti finanziati da EC ed Eurescom. È autore di numerosi articoli in conferenze e riviste e di sette brevetti su sistemi e metodi per servizi. E-mail: corrado.moiso@telecomitalia.it