

Self-Organization & MAS: Autonomic Computing

Approfondimento per il corso di:
Sistemi Multiagente LS

Massimo Bartolini

Ingegneria Due
ALMA MATER STUDIORUM - Università di Bologna a Cesena

1 Introduzione

- Auto-organizzazione

2 Autonomic Computing

- Introduzione
- Principi fondamentali
- Architettura

3 Autonomic Computing e Agenti

- Integrazione tra Autonomic Computing e MAS
- Architetture agent-based per Autonomic Computing

Definizione

- Self-Organization
 - *Organization*: si riferisce alla relazione fra le parti del sistema in termini di struttura e interazioni.
 - *Self*: effettuata da una o più parti del sistema e non imposta da agenti esterni.
- Sistemi che sfruttano tale principio migliorano spontaneamente la loro organizzazione interna come risultato dell'interazione fra le parti del sistema e la mantengono anche in presenza di perturbazioni da parte dell'ambiente.

Caratteristiche di un sistema auto-organizzante

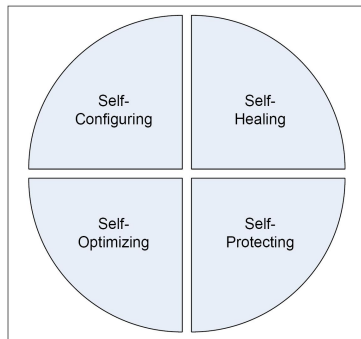
- *Autonomia*: le relazioni e interazioni che definiscono il sistema sono determinate solamente dal sistema stesso.
- *Complessità*: le parti del sistema si intrecciano l'una all'altra tramite relazioni mutuali in permanente cambiamento rendendo più difficile la descrizione e la previsione del comportamento del sistema nella sua interezza.
- *Dinamicità*: l'auto-organizzazione deve essere intesa come un processo e non come uno stato finale.
- *Adattività*: risposte adeguate a cambiamenti esterni.

Che cos' è l'Autonomic Computing

- Paradigma proposto nel 2001 da IBM.
- Idea: Creare un sistema simile al sistema nervoso autonomo umano.
- Un sistema computazionale autonomico deve essere in grado di gestire se stesso e ottimizzare il suo rendimento senza alcun intervento umano.

Self-management

- Auto-configurazione (*Self-Configuring*)
- Auto-riparazione (*Self-Healing*)
- Auto-ottimizzazione (*Self-Optimizing*)
- Auto-protezione (*Self-Protecting*)



Auto-configurazione

- Il sistema deve essere in grado di individuare nuovi componenti e di integrarli nel sistema scoprendo ed imparando le loro caratteristiche.
- L'auto-configurazione permette di adattare il sistema ai veloci cambiamenti dell'ambiente esterno, aumentando così la sua velocità di risposta.

Auto-riparazione

- Il sistema deve monitorare costantemente il proprio stato, eventualmente localizzare i problemi presenti e trovare una soluzione cercandola in un database di rimedi.
- L'auto-riparazione permette di abilitare la resilienza del business. I sistemi autonomici rilevano, diagnosticano e agiscono per prevenire la loro distruzione.

Auto-ottimizzazione

- Il sistema deve essere in grado di tenere sotto controllo il carico di lavoro agendo, se necessario, sulla struttura del sistema stesso aggiungendo o togliendo risorse.
- L'auto-ottimizzazione permette di ottenere un'efficienza operativa mediante il tuning delle risorse e del carico di lavoro per massimizzare l'utilizzo dell'infrastruttura IT.

Auto-protezione

- Il sistema deve monitorare che le politiche di sicurezza non vengano infrante sia da parti del sistema stesso che da parte di sistemi esterni.
- L'auto-protezione permette di mettere al sicuro informazioni e risorse anticipando, rilevando, identificando e proteggendo attivamente il sistema contro attacchi esterni.

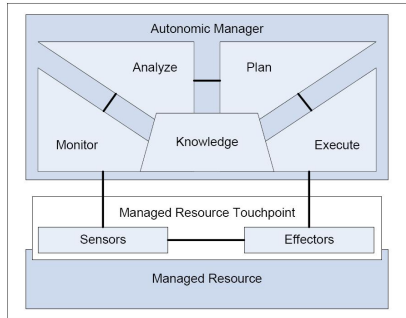
Livelli di automazione

IBM definisce cinque livelli di automazione [IBM, 2006] :

- *Basic*: i componenti del sistema sono gestiti manualmente e singolarmente.
- *Managed*: sono introdotti dei sistemi centralizzati di raccolta delle informazioni.
- *Predictive*: il sistema è in grado di analizzare i dati raccolti, correlando i pattern con le azioni raccomandate dagli amministratori.
- *Adaptive*: il sistema è in grado di prendere le decisioni sulla base dei dati raccolti, riducendo al minimo l'intervento umano.
- *Autonomic*: i componenti del sistema sono totalmente integrati e gestiti dinamicamente sulla base di regole e politiche di alto livello.

Architettura di un Autonomic Element

- Managed Resource
- Autonomic Manager
- Touchpoint



Managed Resource

Componente di un'infrastruttura IT come ad esempio:

- server
- database
- file
- ...

Autonomic Manager

- Componente che si occupa della gestione delle managed resource.
- All'interno di ogni autonomic manager è presente un loop di controllo, il cosiddetto MAPE:
 - *Monitor*: ricezione degli eventi individuati dai sensori che sono connessi alla managed resource.
 - *Analyze*: individuazione dell'obiettivo da raggiungere, utilizzando la knowledge base.
 - *Plan*: pianificazione delle azioni da svolgere per raggiungere l'obiettivo.
 - *Execute*: esecuzione concreta dell'azione pianificata tramite gli attuatori.

Touchpoint

- Componente del sistema che espone lo stato e le operazioni di gestione per una risorsa del sistema.
- All'interno un touchpoint è costituito da una *manageability interface*, che include una *sensor interface* e una *effector interface*, e permette di interagire con la risorsa gestita (managed resource).
- Gli attuatori e i sensori permettono diverse di tipologie di comunicazione tra autonomic manager e managed resource:
 - *Request-Response*
 - *Send notification*
 - *Perform-Operation*
 - *Solicit-Response*

Interazione tra autonomic element

Visione peer-to-peer [Kephart and Chess, 2003]

- Ogni autonomic element è un agente che fornisce e consuma servizi.
- Gli agenti sono: (i) *autonomi* e (ii) *goal-oriented*, perchè eseguono le richieste degli altri agenti solo se queste sono consistenti con i propri goal; (iii) *pro-attivi* perchè prendono iniziativa e senza il controllo umano decidono cosa fare a run-time.

Visione gerarchica [IBM, 2006]

- Ogni autonomic manager può gestire come managed resource un altro autonomic element.

Autonomic element $\equiv$ Agente

Autonomic element - Definizione

Un sistema capace di auto-gestirsi in accordo a determinate policy impostate dall'amministratore. [Kephart and Chess, 2003]

Agente - Definizione

Un sistema informatico situato in un certo ambiente e capace di eseguire azioni autonomamente, all'interno di questo ambiente, al fine di raggiungere i propri obiettivi. [Jennings et al., 1998]

- Si possono quindi facilmente identificare i singoli agenti con i singoli autonomic element.

Autonomic system $\equiv$ Sistema multiagente

Autonomic system - Definizione

Un insieme di autonomic element dalla cui interazione emerge il *self-management*

Sistema multiagente - Definizione

Un insieme di agenti situati in un certo ambiente e interagenti tra loro mediante una opportuna organizzazione.

- Si possono quindi facilmente identificare gli autonomic system con i sistemi multiagente.

Architetture agent-oriented

- Approccio decentralizzato basato su sistemi multiagente.
- Ogni autonomic element è analogo a un agente.
- Il comportamento autonomo dell'intero sistema deriva dalle interazioni tra gli autonomic element.

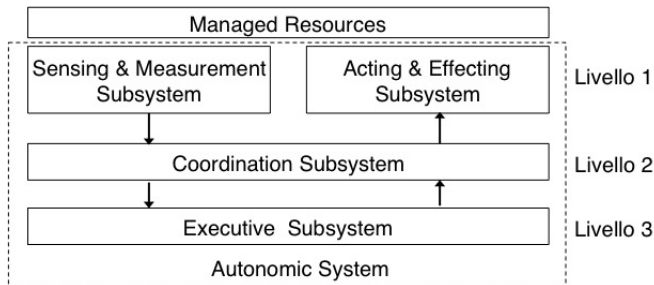
Possibili modelli

Tre articoli con differenti approcci:

- *Expanding the Possibilities for Enterprise Computing: Multi-Agent Autonomic Architectures* - G.Pour.
- *A multiagent systems approach to autonomic computing* - G. Tesauro, D.M. Chess, W.E. Walsh et al.
- *A Self-Organized Model of Agent-Enabling Autonomic Computing for Grid Environment* - H.Guo, J.Gao, P.Zhu, F.Zhang.

Primo modello - Architettura di alto-livello I

Architettura basata su tre livelli principali [Pour, 2006]



Primo modello - Architettura di alto-livello II

Knowledge Vs. Information

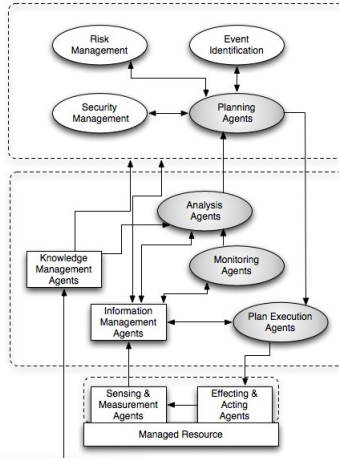
- Knowledge: regole utilizzate dall'autonomic system (*input*).
- Information: dati generati dai sottosistemi (*output*).

Agenti nei tre livelli

- Livello 1: gli agenti agiscono localmente, sono reattivi, autonomi, svolgono azioni preprogrammate.
- Livello 2: gli agenti dispongono di una base di conoscenza euristica per svolgere differenti attività di coordinazione.
- Livello 3: gli agenti hanno goal e piani espliciti per raggiungere i loro obiettivi.

Primo modello - Sottosistemi

Ciascun livello è costituito da differenti sottosistemi / agenti:



Standards, Procedures & Regulations

Secondo modello - Introduzione

Unity [Tesauro et al., 2004]

- Primo prototipo di autonomic system proposto da IBM.
- Architettura decentralizzata.
- Basata su un insieme di agenti che interagiscono tra di loro.

Obiettivi

- Auto-configurazione durante l'inizializzazione.
- Sviluppo di pattern per l'autoriparazione.
- Auto-ottimizzazione dell'uso delle risorse.

Secondo modello - Struttura

Ogni componente è un autonomic element.

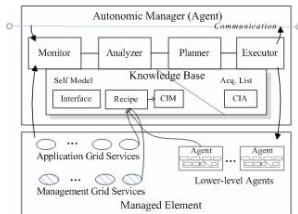
- *Application manager*: responsabile della gestione dell'ambiente, della ricerca delle risorse e della comunicazione con gli altri elementi.
- *Resource arbiter*: responsabili dell'allocazione delle risorse.
- *Registry*: consentono ai diversi elementi di localizzarne altri con cui essi hanno bisogno di interagire.
- *Policy repository*: supportano delle interfacce che permettono agli amministratori di inserire delle policy di alto livello che guidano le operazioni del sistema.
- *Sentinel*: permettono di monitorare il funzionamento di un elemento.

Secondo modello - “Goal-driven Self-assembly”

- 1 Ogni autonomic element, quando inizia a eseguire per la prima volta, conosce solo una descrizione di alto livello di ciò che deve essere svolto e il registry.
- 2 L'autonomic element contatta il registry per localizzare gli elementi esistenti che possono fornire i servizi di cui ha bisogno.
- 3 Una volta che l'elemento ha ottenuto tutte le risorse necessarie si registra a sua volta nel registry per fornire agli altri i suoi servizi.

Terzo modello - Introduzione

- Gli autonomic manager vengono rappresentati come degli agenti, che gestiscono servizi e agenti di più basso livello [Guo et al., 2006]

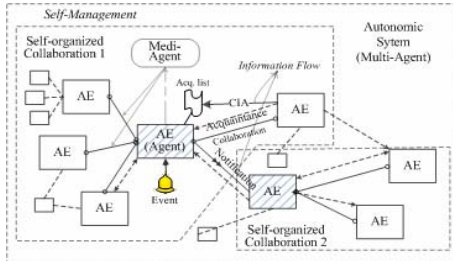


- Ogni agente divulga le proprie *capability* e i propri *interest*.
 - *Capability*: ciò che l'agente è capace di fare.
 - *Interest*: una qualche informazione che l'agente considera importante e necessaria per iniziare ed eseguire un'azione.

Terzo modello - Architettura I

Il sistema può essere definito da tre componenti:

- un insieme di autonomic element
- un insieme di relazioni tra autonomic element
- un meccanismo che permetta al sistema di auto-gestirsi



Terzo modello - Architettura II

Autonomic element - Composizione

- AM (*Autonomic Manager*): un agente.
- ME (*Managed Elements*): elementi come applicazioni, servizi o agenti di più basso livello.
- IR (*Internal Relationship*): relazioni interne tra AM e ME, definite nel *Self-Model* (SM).

Self-Model - Composizione

- *Recipe*: definisce le linee guida per realizzare servizi composti.
- *Interface*: descrive come utilizzare i managed element.
- *CIM* (*Capability and Interest Model*): definisce ciò che gli autonomic element sanno fare e ciò cui sono interessati.

Terzo modello - Architettura III

Relazioni tra autonomic element

- *Conoscenza*: relazione che si verifica quando un autonomic element riceve i *CIA* (*Capability and Interest Advertisement*) resi noti dagli altri; tali informazioni vengono memorizzate unitamente agli autonomic element che le hanno diffuse.
- *Cooperazione*: relazione stabilita per realizzare obiettivi comuni. Ogni elemento realizza un sub-task parziale e lavora insieme agli altri membri per soddisfare il goal collettivo.
- *Notifica*: relazione che si verifica quando un elemento possiede informazioni richieste da un altro elemento; in questo caso viene inviato un messaggio contenente tali informazioni all'elemento interessato.

Bibliografia I



Parashar, M., Hariri, S.

Autonomic Computing - Concepts, Infrastructure, and Applications



IBM (2006)

An architectural blueprint for autonomic computing
IBM Autonomic Computing White paper



Kephart, J.O., Chess, D.M. (2003)

The Vision of Autonomic Computing
IEEE Computer 36(1): 41-50

Bibliografia II



Horn, P. (2001)

Autonomic Computing: IBM's Perspective on the State of
Information Technology

Technical Report, IBM Corporation



Salehie, M., Tahvildari, L. (2005)

Autonomic computing: emerging trends and open problems

*Proceedings of the 2005 workshop on Design and evolution of
autonomic application software, pages 1 - 7*

Bibliografia III



White, S.R., Hanson, J.R., Whalley, I., Chess, D.M., Kephart, J.O. (2004)

An Architectural Approach to Autonomic Computing
Proceedings of the 1st International Conference on Autonomic Computing (ICAC'04), IEEE Computer Society Press, pp 2-9



Tesauro, G., Chess, D.M., Walsh, W.E., Das, R., Whalley, I., Kephart, J.O., White, S.R. (2004)

A multiagent systems approach to autonomic computing
Autonomous Agents and Multi-Agent Systems

Bibliografia IV



Pour, G. (2006)

Expanding the Possibilities for Enterprise Computing:
Multi-Agent Autonomic Architectures

*Proceedings of the 10th IEEE on International Enterprise
Distributed Object Computing Conference Workshops*



Guo, H., Gao, J., Zhu, P., Zhang, F. (2006)

A Self-Organized Model of Agent-Enabling Autonomic
Computing for Grid Environment

Intelligent Control and Automation

Bibliografia V



Brazier, F.M.T., Kephart, J.O., Van Dyke Parunak, H., Huhns, M.N. (2009)

Agents and Service-Oriented Computing for Autonomic Computing: A Research Agenda

IEEE Internet Computing, vol. 13, no. 3, pp. 82-87



Jennings, N.R., Sycara, K., Wooldridge, M. (1998)

A Roadmap of Agent Research and Development

Autonomous Agents and Multi-Agent Systems, vol 1, pp 7-38



Gardelli, L. (2008)

Ingegneria di Sistemi Auto-organizzanti con il Paradigma Multiagente

Tesi di dottorato