

PCD Module 5.3

Ingegneria dei sistemi distribuiti: EDA e ESB

Event-Driven Architectures (EDA)

Da Publish/Subscribe agli Event Services

Il modello Publish/Subscribe visto nel modulo MOM è la base per impostare gli **Event Services** e le **Architetture guidate dagli eventi (EDA)**.

- I messaggi rappresentano eventi come cose che succedono dentro o fuori il sistema o l'azienda (problemi, opportunità, soglie, deviazioni...)
- Il significato è a livello dominio/azienda (non applicativo) (es. inventario di un rivenditore)
- Compongono/correlano eventi per semantiche di alto livello (congestione del traffico, inquinamento e traffico...)

Modello di comunicazione effettiva in diversi domini applicativi (es. sensori che generano dati, notificato come eventi)

Architettura guidata dagli eventi

Architettura software adottata nel modello P/S per disseminare **eventi** immediatamente o a tutte le parti interessate. Promuove la produzione, la ricezione, il consumo e la reazione agli eventi. Le parti interessate valutano l'evento e opzionalmente compiono azioni.

Un evento è un **cambiamento di stato**, es. quando si acquista una macchina, il suo stato passa da "in vendita" a "venduta". L'architettura software deve trattare questo cambio come evento la cui occorrenza può essere far saputa ad altre applicazioni.

Gli eventi sono notificati tramite **notifiche degli eventi**, ovvero messaggi asincroni prodotti, pubblicati, propagati, trovati e consumati quando accade un evento. Gli eventi non viaggiano, capitano solamente.

Concetti principali di EDA

- **emettitori di eventi** (agenti): hanno la responsabilità di catturare, gestire e trasferire eventi. Un emettitore non conosce i consumatori degli eventi.
- **consumatori di eventi** (assorbitori): hanno la responsabilità di applicare una reazione appena un evento si presenta. La reazione può o non può essere completamente fornita dal consumatore.

- **canali degli eventi:** condotti in cui gli eventi sono trasmessi dagli emettitori ai consumatori. La conoscenza della corretta distribuzione degli eventi è esclusivamente presentata all'interno del canale. L'implementazione fisica di tali canali può essere basata su componenti tradizionali, e.s. middleware orientati ai messaggi.

Accoppiamento estremamente leggero

Architettura software effettiva per progettare sistemi i cui componenti/servizi sono necessari per essere estremamente poco accoppiati. I componenti che sono sorgenti di un evento non hanno bisogno di essere accoppiate o conoscere il consumatore di una notifica dell'evento.

Proprietà: apertura, dinamismo, flessibilità.

Stili di processazione degli eventi

- Simple Event Processing, su eventi notabili
- Stream Event Processing, eventi notabili e ordinari
- Complex Event Processing, confluenza di eventi

Integrazione con SOA

Una EDA può essere usata a complemento di **architetture orientate ai servizi (SOA)**.

Mentre SOA generalmente si adatta meglio a uno scambio di richieste/risposte, EDA introduce capacità di **processi asincroni a lunga durata**. I servizi possono essere attivati tramite **trigger** eseguiti su eventi in arrivo. EDA a volte è chiamato *SOA guidato dagli eventi*.

Richiamo sulle funzionalità di SOA

SOA: tutte le funzioni o servizi sono definiti tramite un linguaggio descrittivo e dove le loro interfacce sono ritrovabili in una rete.

Permette di muoversi fuori dall'*approccio a silo*, dove ogni dipartimento costruisce il suo sistema senza nessuna conoscenza di cosa sia stato fatto da altri. L'approccio a silo porta a situazioni inefficienti e costose dove la stessa funzionalità è sviluppata, prodotta e mantenuta molte volte.

SOA è basato su un portfolio di servizi condivisi tra l'organizzazione: fornisce un modo per riusare e integrare efficientemente asset già esistenti.

Caratteristiche di SOA

- **Interazioni poco accoppiate:** i servizi sono invocati indipendentemente dalla loro tecnologia e posizione

- **Comunicazioni 1:1**: un servizio specifico è invocato da un consumatore per volta. Le comunicazioni sono bidirezionali.
- **Trigger basati sui consumatori**: il flusso di controllo è inizializzato dal client (il consumatore)
- **Sincrono**: SOA è a volte basato su un protocollo convenzionale request/reply basato sui messaggi, a volte implementato utilizzando un approccio sincrono.

Aggiungere caratteristiche di EDA

- **Interazioni disaccoppiate**: i pubblicatori non sono a conoscenza dell'esistenza di sottoscrittori di eventi
- **Comunicazioni multi-a-molti**: messaggi P/S dove un evento specifico può impattare molti sottoscrittori
- **Trigger basati sugli eventi**: flusso di controllo che è determinato dal destinatario, basato su un evento postato
- **Asincrono**: supporta operazioni asincrone tramite event messaging, usato per comunicare tra due o più processi applicativi. La comunicazione è iniziata da una notifica di evento.

Enterprise Service Bus (ESB)

- **ESB** combina approcci event-driven + service-oriented per semplificare l'integrazione di BU, unendo piattaforme e ambienti eterogenei.
- Agisce come livello intermediario per abilitare la comunicazione tra diversi processi applicativi.
 - un servizio prodotto su un ESB può essere eseguito da un consumatore o un evento
 - supporta sincrono e asincrono, facilitando le interazioni tra uno o più stakeholders
- Diverse implementazioni da diverse aziende propongono ESB come la chiave per risolvere l'integrazione aziendale. Commerciale (IBM WebSphere, SAP Process Integration, Windows Azure Service Bus) o open source (Apache Camel, JBoss ESB, Mule ESB...)

Servizi di ESB

Non ci sono specifiche ufficiali su cosa debba essere un'implementazione ESB, ma è comunemente accettato che debba fornire servizi di **trasporto**, **eventi** e **mediazione**.

- **Servizi di trasporto**: devono assicurare l'invio del messaggio tra processi aziendali interconnessi tramite un bus aziendale. Include l'instradamento basato sui contenuti (msg). Parte di un ambiente mission-critical, questi servizi sono transazionali, sicuri e monitorati.
- **Servizi di eventi**: forniscono capacità di cattura, esecuzione e distribuzione degli eventi. Relativi alla nozione di processo degli eventi.
- **Servizi di mediazione**: abilita l'interoperabilità tra servizi sul bus.

Servizi di mediazione

- Assicura il protocollo necessario per integrare sistemi eterogenei di corrispondenza
 - come due diversi servizi non è necessario utilizzare lo stesso protocollo di trasporto, il servizio di mediazione si prende cura della trasformazione da un protocollo a altro, in modo che la comunicazione è possibile.
 - l'interruttore protocol è trasparente per tutti i partecipanti servizi di una transazione di affari
- Permette di trasformare il contenuto di qualsiasi messaggio allo scopo di integrazione aziendale
 - assicura che i dati che transita attraverso il bus siano comprensibili da qualsiasi processo
 - consente l'aumento del contenuto di arricchire un messaggio con informazioni aggiuntive

Esempi su slide

Imprese come sistemi reattivi

L'integrazione di EDA con SOA consente di aumentare SOA tempi di risposta. Aggiunta la capacità o reagendo agli ambienti imprevedibili e asincroni

Caratteristiche di sistemi reattivi

- tempi di risposta
- resilienza
- elasticità (scalabilità)
- basato sui messaggi