
CAPITOLO 1

Introduzione

1.1 Ambito Tesi

Nel corso degli anni sono avvenuti profondi cambiamenti nell'ambito delle tecnologie web: si è passati dalle risorse statiche, quali pagine HTML, alle risorse dinamiche, ad esempio le pagine PHP, per arrivare alle applicazioni Web. Con l'evoluzione del Web abbiamo assistito ad una migrazione delle applicazioni: il software installato sulle macchine si sta spostando verso il Web, fino a diventare servizi web (*Web Services*), i quali non forniscono un'interfaccia grafica all'utente (*GUI*), come nel caso delle pagine web o di un programma eseguibile, ma soltanto delle funzionalità. Gli sviluppatori potranno poi utilizzare i Web Services e quindi le funzioni messe a disposizione, per realizzare delle applicazioni che interagiscono con questi servizi; in questo modo è possibile riutilizzare il codice sviluppato da altri, per realizzare nuove applicazioni, utilizzando anche diversi linguaggi di programmazione. Tali applicazioni potranno essere eseguite su piattaforme differenti, grazie al fatto che i Web Services utilizzano la rete Internet come strumento per veicolare le informazioni scambiate.

Una definizione di web service è quella fornita dal W3C: “*un servizio web (Web Service) è un sistema software progettato per supportare l’interoperabilità tra diversi elaboratori su di una rete. Esso ha un’interfaccia descritta in un formato elaborabile da una macchina. Altri sistemi possono interagire con il Web Service, attivando le operazioni descritte nell’interfaccia tramite appositi messaggi SOAP, tipicamente trasportati attraverso il protocollo HTTP, utilizzando una serializzazione XML in aggiunta ad altri standards relativi al Web*”. [W3C WS]

Mentre il *servizio* è una risorsa caratterizzata “*dall’insieme astratto di funzionalità che esso fornisce*” attraverso la sua interfaccia (il Web Service), concretamente deve esistere una sua implementazione pratica, rappresentata da un modulo software o hardware detto **agente**, che invia e riceve messaggi. Quindi un servizio “*è una risorsa astratta che rappresenta la possibilità di eseguire dei compiti rappresentanti una funzionalità coerente dal punto di vista delle entità fornitrici (provider entities) e delle entità richiedenti (requestor entities). Per essere utilizzato, un servizio deve essere realizzato attraverso un agente fornitore (provider agent)*” (Figura 1.1). [W3C WS]

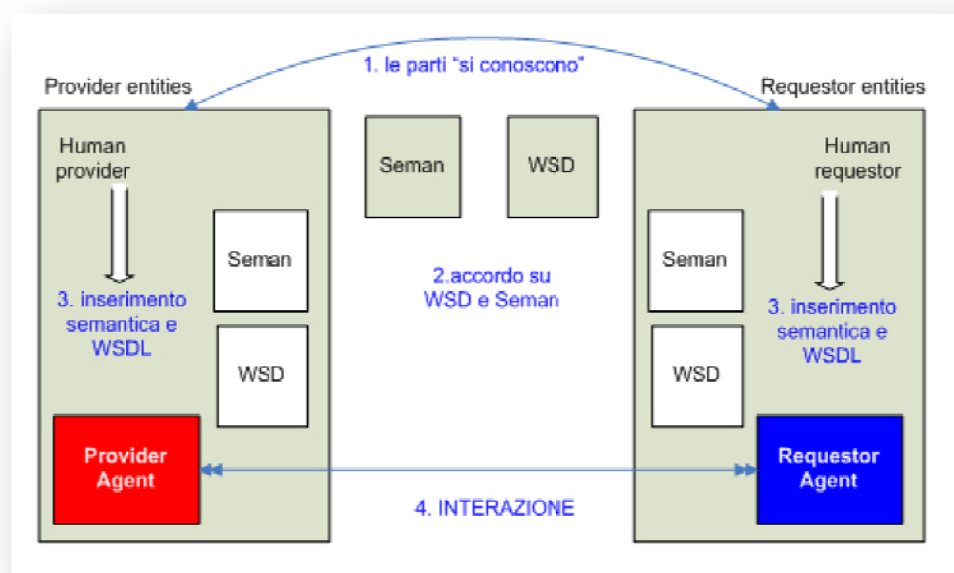


FIGURA 1.1

Una descrizione completa del servizio è realizzata attraverso i documenti WSDL, dove sono definiti tutti i dettagli dell’interfaccia, i tipi di dati utilizzati dal servizio, le operazioni, le informazioni relative al protocollo di trasporto utilizzato e l’indirizzo fisico dove si trova la risorsa.

1.1.1 Architettura dei Web Services

L'architettura di funzionamento dei Web Services è nota con il nome di SOA (*Service Oriented Architecture*), si basa sulle tecnologie SOAP, WSDL e UDDI. Concettualmente consiste di tre attori che eseguono altrettanti ruoli ben definiti (Figura 1.2). [JWS]

Il *Service* è l'implementazione del Web Service, ospitata nel provider; il *Service Description* è l'interfaccia del Web Service, espressa in XML rispettando uno o più standards.

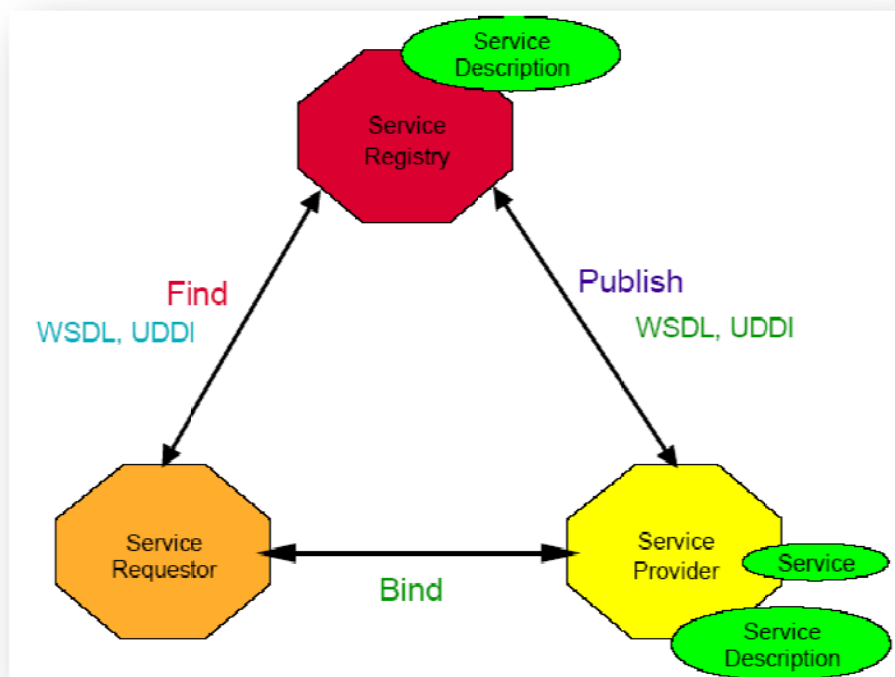


FIGURA 1.2

I tre ruoli dell'architettura sono:

- **Service Provider:** sono le entità fornitrici del servizio web; nell'ottica dell'architettura pura, rappresenta la piattaforma software che ospita il servizio e ne gestisce l'accesso attraverso la rete.
- **Service Registry:** si tratta di un registro che contiene i file descrittivi (WSDL) dei servizi web pubblicati dai provider. L'utente (*requestor*) che vuole connettersi ad un servizio web, può ottenere da questo registro le informazioni necessarie per il collegamento (*binding*) al servizio, dinamicamente in fase d'esecuzione o staticamente in fase di sviluppo.
- **Service Requestor:** l'applicazione che richiede di invocare o inizializzare l'interazione con il provider dove è ospitato il servizio. Questo ruolo può essere svolto da un qualsiasi utente che usa un browser oppure da un altro servizio web.

Dopo aver definito i ruoli che i partecipanti possono svolgere nei Web Services, si osserva come questi interagiscono tra loro; ci sono tre tipi fondamentali di interazioni: pubblicazione (*publish*), ricerca (*find*) e collegamento (*bind*).

- **La pubblicazione:** affinché un web service sia accessibile, è necessario che la sua descrizione formale sia pubblicata in modo tale da essere reperibile per il richiedente.
- **La ricerca:** attraverso questa operazione il richiedente ottiene la descrizione del servizio web, attraverso una query fatta sul “registro dei servizi”. Può svolgersi durante lo sviluppo oppure al momento dell’esecuzione; nel primo caso sono recuperati i documenti che descrivono il servizio; nel secondo, le informazioni di binding.
- **Il collegamento:** è ciò che un’applicazione esegue quando usa la descrizione del servizio per creare un messaggio da inviare al service provider, al fine di individuare e contattare il servizio desiderato.

In un tipico scenario di funzionamento, esiste un *server provider* che ospita l’implementazione del servizio web, rendendo accessibile attraverso la rete il file WSDL che descrive il servizio agli utenti e al *service registry* in cui è reso disponibile per la ricerca. Il richiedente ottiene la descrizione del servizio attraverso l’operazione di ricerca, e quindi utilizza le informazioni ricevute per collegarsi al provider e interagire con il servizio.

1.1.2 Pila dei Protocolli

Dopo aver analizzato l'architettura che sta alla base del funzionamento dei Web Services è interessante osservare i protocolli e le tecnologie utilizzate per eseguire le tre operazioni viste nel precedente paragrafo. Nella pila dei protocolli in Figura 1.3, sulla destra si trovano i requisiti che i vari livelli devono rispettare, sulla sinistra invece sono indicati gli standard tecnologici applicati ai livelli. [WSPA]

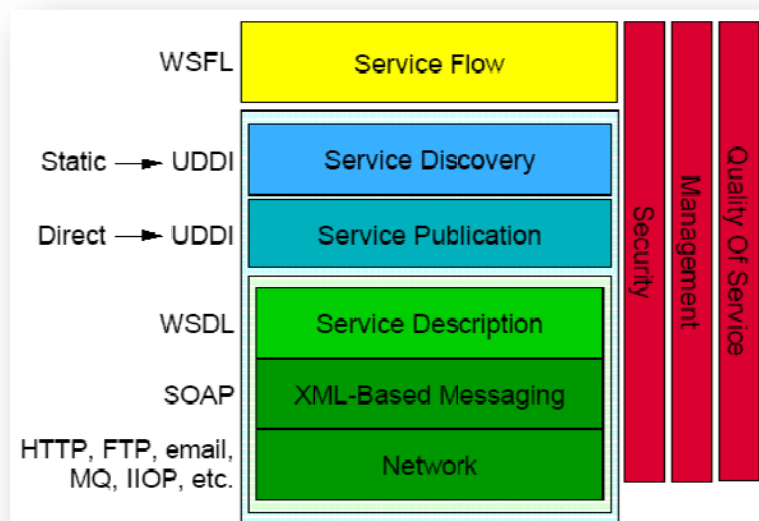


FIGURA 1.3

Alla base si trova il livello di rete (*Network*), contiene le procedure che si occupano del trasporto delle informazioni. Il protocollo utilizzato è HTTP, ma sono supportati altri protocolli come FTP, SMTP.

Ad un livello superiore si trova *XML-Based Messaging*, dove viene costruito il messaggio, cioè l'unità di dati scambiata; questo è basato sull'utilizzo di XML. Il protocollo utilizzato per lo scambio di messaggi è il SOAP (Simple Object Access Protocol).

Il livello successivo, di *service description*, si occupa della descrizione dei servizi, utilizzando il WSDL come standard.

Ai livelli superiori, indicati con *service publication* e *service discovery*, si trovano altri documenti atti a descrivere in maniera più dettagliata il contesto e la qualità del servizio; per queste attività, lo standard è UDDI (Universal Description, Discovery and Integration).

In cima alla pila ci sono i documenti che si occupano di descrivere il flusso e la composizione dei servizi, utilizzando grammatiche particolari come WSFL (Web Services Flow Language).

1.1.2.1 Il protocollo SOAP

Si tratta di un protocollo semplice, basato su XML, per lo scambio d'informazioni in un ambiente distribuito. Il suo compito principale è quello di garantire l'interoperabilità tra sistemi diversi che cercano di dialogare in ambienti disomogenei dal punto di vista software e hardware. [W3C SOAP]

Questo standard permette di eseguire chiamate di procedure remote (**RPC**: *Remote Procedure Call*, tecnologia attraverso la quale sono stati realizzati i primi sistemi distribuiti.) o l'invio di messaggi XML attraverso il protocollo HTTP.

La struttura del messaggio SOAP (Figura 1.4) comprende tre parti principali:

- **Envelope**: è la busta, nodo contenitore del messaggio SOAP.
- **Header**: contiene l'intestazione del messaggio, è opzionale.
- **Body**: è il corpo del messaggio SOAP, contiene una sequenza di blocchi SOAP.

Nel listato di Figura 1.4 è riportato un esempio di messaggio SOAP trasportato mediante HTTP e la relativa risposta.

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

FIGURA 1.4

1.1.2.2 Il registro UDDI

Nei precedenti paragrafi si è parlato del registro UDDI, si tratta di un meccanismo di registrazione e consultazione di un archivio di servizi, pensato per la ricerca di Web Service in rete. Si tratta quindi di una base di dati distribuita, contenente informazioni sulle aziende, suddivise per tipologia, sui servizi da queste offerti, con i dettagli tecnici sugli stessi e su come raggiungerli. UDDI fa parte di un progetto sponsorizzato dall'OASIS (*Organization for the Advancement of Structured Information Standards*, un consorzio internazionale per lo sviluppo e l'adozione di standard nel campo dell'e-business e dei Web), volto a creare una tecnologia standard per consentire alle aziende e alle applicazioni di trovare e utilizzare i Web Service in modo rapido, semplice e dinamico. [UDDI]

Una "registrazione" UDDI consiste, di tre diverse componenti:

- **Pagine bianche (White Pages):** contengono indirizzo, contatti e identificativi dell'azienda che offre uno o più servizi;
- **Pagine gialle (Yellow Pages):** contengono l'elenco dei servizi, suddivisi per categorie;
- **Pagine verdi (Green Pages):** informazioni tecniche dei servizi fornite dall'azienda

L'UDDI è stato progettato per essere interrogato da messaggi SOAP e per fornire il collegamento ai documenti WSDL che descrivono i vincoli protocollari ed i formati dei messaggi necessari per l'interazione con i Web Service elencati nella propria directory.

1.1.2.3 Il linguaggio BPEL

In cima alla pila dei protocolli (Figura 1.3) si trovano i linguaggi di WSFL, uno di questi è il BPEL (*Business Process Execution Language*). Questo linguaggio fornisce una grammatica basata sull'XML, utilizzata per descrivere la logica e controllare e coordinare la partecipazione dei servizi web nel flusso di un processo. Il BPEL viene utilizzato per comporre vari servizi web, all'interno di un *business process*, in questo modo è possibile creare delle nuove applicazioni completamente nuove, con una loro interfaccia pubblica.

Il termine *business process*, fa riferimento ad un generico tipo di processo che viene eseguito all'interno di un'organizzazione per adempiere ad uno o più servizi, al fine di raggiungere un determinato scopo.

La grammatica del BPEL può essere interpretata ed eseguita da un *orchestration engine BPEL*, controllato da uno dei partecipanti. L'*engine* è un modulo software, generalmente pubblicato in un Web Server, che si occupa di coordinare l'esecuzione di tutte le attività del flusso del processo, e gestire l'esecuzione di attività di correzione quando si verificano delle eccezioni nel corso dell'esecuzione del processo.

Il BPEL fornisce quindi una descrizione più accurata rispetto a quanto viene fatto dal WSDL, ma soprattutto, permette l'integrazione di diversi servizi web al fine di ottenerne di nuovi, in questo modo è possibile modificare facilmente il funzionamento di un *business process*, sulla base delle necessità degli utenti.

Usando il BPEL per definire i *business process*, le aziende hanno il potere di scegliere i processi e i servizi da incorporare nelle loro operazioni, tra quelli che meglio si adattano alle loro esigenze. In questo modo è possibile sostituire o aggiornare alcuni aspetti di un *business process*, senza influenzare i sistemi che funzionano già correttamente. Per esempio un'azienda può cambiare il servizio di gestione del magazzino senza influenzare altri servizi come quello per la gestione degli ordini. Il BPEL, fornendo una descrizione del flusso di un processo può essere utilizzato anche per effettuare delle ricerche basate sul funzionamento del processo stesso.

1.1.3 Servizi Web Semantici

Idealmente, la descrizione di un servizio dovrebbe contenere informazioni sufficienti tali da permettere ad un agente automatico non solo di utilizzare il servizio, ma di capire anche se quel servizio risulta appropriato per gli scopi previsti. Attraverso la descrizione effettuata con il WSDL non è possibile raggiungere questo obiettivo, in quanto viene fornito soltanto un insieme di metodi e operazioni; è necessaria invece, una descrizione in termini di aspetti procedurali e flusso di operazioni, il BPEL offre questo tipo di descrizione.

I servizi web semantici sono il punto di convergenza tra i servizi web e il web semantico, quest'ultimo rappresenta l'evoluzione di Internet, da semplice sistema di comunicazione e recupero di documenti, ad un sistema "intelligente" in cui l'informazione sarà compresa da specifici software in grado di cooperare e assistere l'utente in complicati task, al fine di ottenere una conoscenza riutilizzabile, accessibile e condivisibile. Elemento fondamentale per raggiungere questo obiettivo è **l'ontologia**, una "definizione formale di un dominio di conoscenza", dove *definizione formale* significa enumerare e definire in modo più o meno formale i concetti e le relazioni che tra essi sussistono, mentre il *dominio di conoscenza* è una parte del mondo isolata considerata con i suoi concetti fondamentali. L'introduzione e l'utilizzo delle ontologie servirà principalmente per risolvere le ambiguità dei termini, definendo il contesto in cui il termine è usato il suo significato diventa chiaro; inoltre sarà possibile superare la notevole diversità di formati presenti nella rappresentazione delle informazioni, l'ontologia può descrivere in modo formale i contenuti di ogni sorgente di informazione e quindi aiutare l'utente nella ricerca.

In questo ambito, il termine *semantico* ha il significato di "elaborabile dalla macchina" e non intende fare riferimento alla semantica del linguaggio naturale e alle tecniche dell'intelligenza artificiale; la semantica dei dati consiste nelle informazioni utili alla macchina per utilizzare, ed eventualmente convertire, i dati in maniera corretta.

I servizi web semantici aggiungono ai servizi convenzionali dei meccanismi di auto-descrizione; tramite i servizi semantici sarà possibile capire qual è la necessità dell'utente, quindi formulare l'obiettivo (il "goal") e realizzarlo tramite interazioni con altri servizi semantici. Tra le caratteristiche dei servizi web semantici, vi è anche la capacità di adattare le richieste scambiate tra i servizi, tramite i mediatori; ad esempio in un caso semplice, convertire valute o misure, oppure in un caso più complesso collezionare in un unico messaggio i dati relativi ad una prenotazione o un ordine, che devono essere visti unitariamente da un servizio e sono viceversa prodotti separatamente da un altro servizio.

Senza l'adozione di una semantica comprensibile da un elaboratore, lasciando ad esso la ricerca di servizi web in ambiti eterogenei si rischia di ottenere come risultato soltanto quei servizi che soddisfano in maniera totale certi criteri di identità (la classica parola chiave) e si perderebbero quelli che implementano servizi simili o più ampi, ma non identici rispetto a quello cercato. Per superare questo limite, la soluzione sta nel formalizzare delle relazioni fra simboli (le ontologie) che anche un elaboratore possa capire e utilizzare nelle sue attività di ricerca.

Portare tutte queste classificazioni ai Web Services è la strada per arrivare ai cosiddetti servizi Web semantici. Il consorzio OASIS (Organization for the Advancement of Structured Information Standards) e il W3C sono i principali responsabili dello sviluppo e della certificazione degli standard legati a un linguaggio che esprima le ontologie e a uno che permetta di descrivere un Web Service in maniera non ambigua comprensibile anche a una macchina. Nel primo caso ci sono *WSML* (Web Service Modeling Language) e *OWL* (Ontology Web Language, l'evoluzione del DARPA Agent Markup Language o DAML); nel secondo *WSMO* (Web Service Modeling Ontology) e *OWL-S*.

1.2 Obiettivo della Tesi

Questa tesi ha come obiettivo la presentazione delle tecnologie, e del lavoro svolto durante il periodo di tirocinio, per annotare semanticamente i servizi web.

L'annotazione semantica di un servizio web non può prescindere dalla conoscenza del servizio stesso, infatti il tirocinio è iniziato con l'implementazione di alcuni servizi web, la loro pubblicazione e quindi, dopo averne testato il corretto funzionamento, sono stati creati i file BPEL che descrivono il flusso delle operazioni dei servizi web; soltanto a questo punto si può procedere con l'inserimento della semantica nel processo, modificando manualmente il codice del file BPEL.

Al fine di evitare errori e facilitare la procedura di inserimento delle annotazioni semantiche, nella seconda parte del periodo di tirocinio è stato progettato e quindi realizzato un software che, attraverso una procedura grafica, effettua le modifiche necessarie per aggiungere la semantica al codice del file BPEL 2.0, rispettando le regole definite nella specifica BPEL4SWS.

1.3 Struttura della Tesi

Dopo aver fatto una panoramica, in questo capitolo introduttivo, circa l'ambito dei servizi web semantici; nel capitolo 2 verranno presentate le specifiche e quindi le tecnologie che sono disponibili per annotare semanticamente i servizi web: iniziando con il WSDL, si parlerà poi del WS-BPEL 2.0, verrà poi analizzato e discusso il BPEL4SWS e poi il WSML. Infine si effettuerà una panoramica sullo stato dell'arte circa gli strumenti disponibili per la creazione, la modifica e l'utilizzo dei file basati su questi linguaggi.

Nel capitolo 3 verrà descritto il progetto dello strumento software che deve permettere l'annotazione dei file BPEL, attraverso una procedura grafica, rispettando la specifica BPEL4SWS: dopo aver analizzato i requisiti del software, verrà effettuata una panoramica circa l'architettura del software e una descrizione delle librerie utilizzate, si analizzeranno quindi i vari moduli che compongono l'applicazione.

Nel capitolo 4 si andrà a fare una panoramica sul lavoro svolto durante la prima parte del tirocinio, dedicata alla realizzazione dei servizi web: si partirà descrivendo l'ambiente di lavoro in cui verranno sviluppati i servizi, per poi passare alla fase della pubblicazione degli stessi, quindi all'installazione dei database di supporto. Successivamente verrà descritta la procedura seguita per l'invocazione dei servizi tramite l'utilizzo di client, infine si parlerà della fase di creazione e test dei file BPEL.

Nel capitolo 5 verrà presentata una sessione di lavoro del software di annotazione realizzato nella seconda parte del tirocinio, con il supporto di varie immagini saranno mostrate le varie funzionalità dell'applicazione.

Infine nel capitolo 6 si andranno a discutere i risultati ottenuti, effettuando considerazioni sui possibili sviluppi futuri e sulle difficoltà incontrate durante lo svolgimento del tirocinio.