

Tecnologie XML: un elemento essenziale per l'interoperabilità

Oreste Signore

W3C Office in Italy e ISTI-CNR
Area della Ricerca CNR di Pisa – via G. Moruzzi, 1 – 56124 Pisa (Italy)
e.mail: oreste@w3.org
personal home page: <http://www.weblab.isti.cnr.it/people/oreste/>

Abstract. Nel contesto applicativo attuale, che deve tener conto della natura totalmente decentralizzata del web, l'interoperabilità costituisce un fattore chiave di successo e una caratteristica di qualità. Nel Web di oggi un ruolo fondamentale viene giocato dalla famiglia di tecnologie XML, il cui uso coerente consente di realizzare applicazioni con un alto livello di accessibilità, portabilità e flessibilità, contribuendo anche ad una potenziale riduzione dei costi. Inoltre, la coerenza con i principi di base del web e con gli obiettivi a lungo termine del World Wide Web Consortium (W3C) garantisce l'evoluzione delle applicazioni verso l'ambizioso scenario del Semantic Web. In questo intervento verrà discussa l'interoperabilità tecnologica e semantica, mostrando come le tecnologie W3C possano contribuire alla realizzazione di applicazioni efficaci, salvaguardando i costi, e supportano l'evoluzione dalla interoperabilità tra applicazioni verso il “web of meaning”.

Introduzione

Il contesto applicativo attuale è sempre più basato sul web, che, per sua natura, è un mondo aperto, in cui interagiscono realtà diverse per organizzazione e cultura. Ne consegue l'importanza di disporre di strumenti e tecnologie che consentano una reale *interoperabilità* delle applicazioni. In questo breve lavoro vengono illustrate le caratteristiche del Web, facendo riferimento alla sua evoluzione nel tempo, e viene evidenziata l'importanza di una reale interoperabilità, che non è un aspetto meramente tecnologico, ma anche semantico, e la fondamentale importanza di poter garantire l'accessibilità dei siti.

Per quanto concerne l'*interoperabilità tecnologica*, viene fatto un breve riferimento a XML e XML Schema, discutendo il ruolo essenziale giocato da XML in alcuni aspetti relativi alla presentazione dell'informazione. Come scenario di interazione uomo-macchina, viene data una certa rilevanza alla Recommendation XForms.

Riguardo all'*interoperabilità semantica*, viene fatto un breve accenno ai metadati e al Resource Description Framework, richiamando anche il contesto del Semantic Web. Come scenario di interoperabilità anche semantica, viene fatto un breve richiamo ai Web Service.

Il Web

Il World Wide Web Consortium (W3C)

Il World Wide Web Consortium (<http://www.w3.org>), è un consorzio di imprese che regola l'evoluzione del web, sviluppando tecnologie e definendo protocolli comuni che ne favoriscano l'evoluzione e assicurino l'interoperabilità. Gli obiettivi a lungo termine del W3C possono essere espressi sinteticamente come:

- *Universal Access*: Rendere il Web accessibile a tutti, promuovendo tecnologie che tengono conto delle notevoli differenze in termini di cultura, formazione, capacità, risorse materiali, e limitazioni fisiche degli utenti in tutti i continenti.
- *Semantic Web*: Sviluppare un ambiente software che consenta ad ogni utente di fare il miglior uso possibile delle risorse disponibili sul Web.
- *Web of Trust*: guidare lo sviluppo del Web tenendo in attenta considerazione gli aspetti innovativi che questa tecnologia solleva in campo legale, commerciale e sociale.

Le tecnologie W3C costituiscono un insieme di strumenti che permettono di realizzare applicazioni realmente interoperabili, e di strutturare, condividere e utilizzare, la conoscenza, in un ambiente aperto e in costante evoluzione.

I principi informatori

Il Web è una applicazione costruita su Internet, e quindi ne ha ereditato i principi fondamentali, che, molto sinteticamente, sono:

- *Interoperabilità*: Le specifiche dei linguaggi e dei protocolli del Web devono essere compatibili tra di loro, e consentire a qualunque hardware e software di operare tra di loro.
- *Evoluzione*: Il Web deve essere in grado di accogliere le nuove tecnologie. Principi di progettazione quali la semplicità, la modularità e l'estensibilità aumentano le possibilità che il Web sia in grado di funzionare con le tecnologie emergenti, quali i dispositivi mobili e la televisione digitale, o con altre tecnologie che compariranno.
- *Decentralizzazione*: La decentralizzazione è senza dubbio il principio più nuovo e difficile da applicare. Per consentire che il Web si diffonda realmente su scala mondiale senza rischiare errori o interruzioni, l'architettura (e Internet) devono limitare o eliminare le dipendenze da nodi centrali.

L'interoperabilità

Due applicazioni sono interoperabili se si possono scambiare dati e servizi in modo efficace e consistente, permettendo la comunicazione tra piattaforme hardware e software eterogenee. Nel contesto attuale, l'interoperabilità costituisce un fattore chiave di successo. I vantaggi sono ben noti, e comprendono la possibilità di salvaguardare gli investimenti fatti, grazie alla possibilità di adeguarsi all'evoluzione degli ambienti operativi, e quella di allargare il numero di applicazioni con cui interagire. La coerenza con un contesto tecnologico solido è un elemento fondamentale per raggiungere un buon livello di interoperabilità. Tuttavia, l'interoperabilità non è un aspetto meramente tecnologico. Bisogna tener presenti le differenti culture e il diverso modo di percepire i concetti, quindi occorre considerare non solo l'*interoperabilità tecnologica*, ma anche quella *semantica*.

L'obiettivo dell'Universal Access, in particolare, pone l'accento sul superamento delle differenze di cultura, lingua, formazione, capacità, risorse materiali, limitazioni fisiche e cognitive degli utenti in tutti i continenti. Quindi, in un web che voglia essere davvero universale, vanno considerate, e superate, anche le potenziali *barriere culturali* determinate dalle differenze di tradizione e storia degli utenti.

Espressioni, colori, immagini, classificazioni di concetti, possono essere totalmente diversi per persone di culture diverse. Anche l'aspetto esteriore può determinare un diverso modo di percepire il messaggio, che è costituito da:

- *Contenuto*: il contenuto reale del messaggio, che l'autore intende comunicare;
- *Struttura*: il modo in cui è organizzata l'informazione (es. titolo, autore, corpo del testo, firma);
- *Presentazione*: il modo in cui l'informazione viene presentata all'utente (tipo di carattere, colore, organizzazione della pagina, etc.).

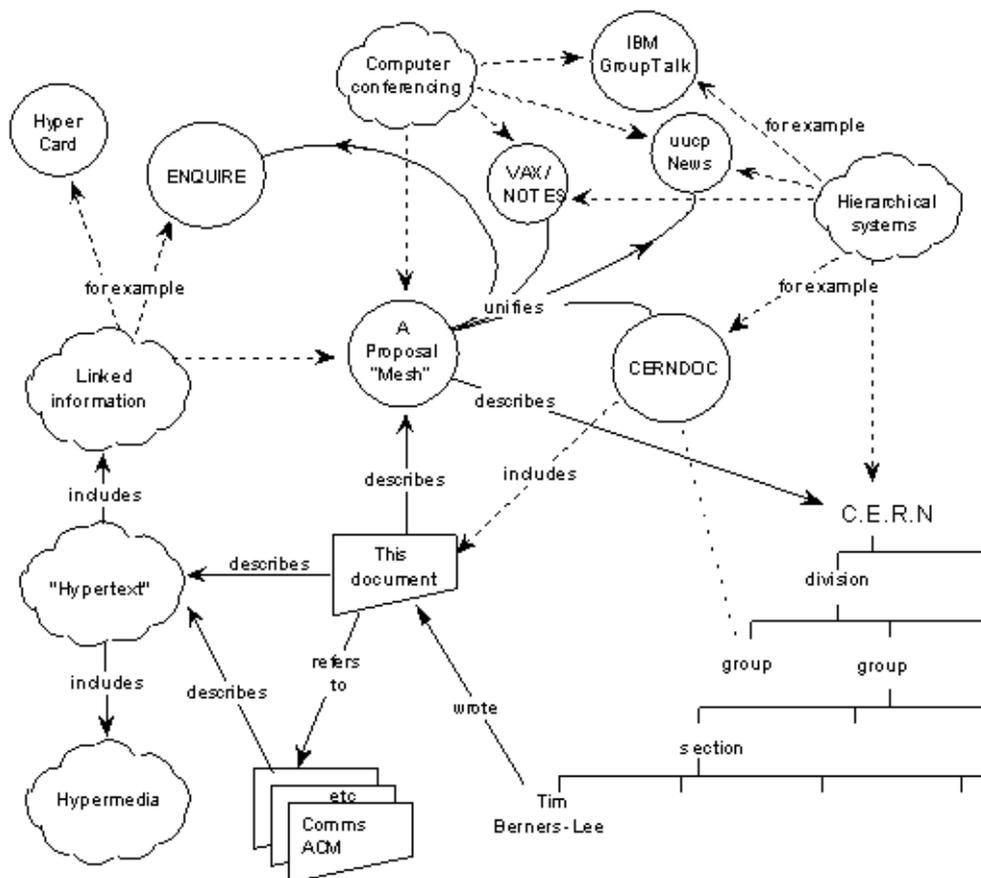


Figura 1 - La visione originaria del World Wide Web

Ciascuno di questi componenti ha una *valenza semantica*, e veicola una *conoscenza esplicita o tacita*. Condividere la conoscenza sul web significa poter disporre di strumenti e

tecnologie che consentano di *esprimere i contenuti, strutturarli e presentarli* in modo adeguato, rendendone esplicita la *semantica* e consentendo la fruizione dell'informazione a tutti, indipendentemente dal particolare retroterra culturale e dal contesto tecnologico.

L'evoluzione

Il World Wide Web, nato da un'idea di Tim Berners Lee (Figura 1), ha avuto un impatto incredibile sulla nostra realtà, modificando radicalmente il modo in cui le persone accedono all'informazione, e cambiando la vita di milioni di persone. Nel giro di pochi anni, vi è stata una notevole evoluzione del web (Figura 2), con uno spostamento dell'enfasi dall'*interazione uomo-macchina* all'*interazione macchina-macchina*. Si noti, in Figura 1, come gli archi rechino una esplicita identificazione del loro significato (*semantica dei link*).

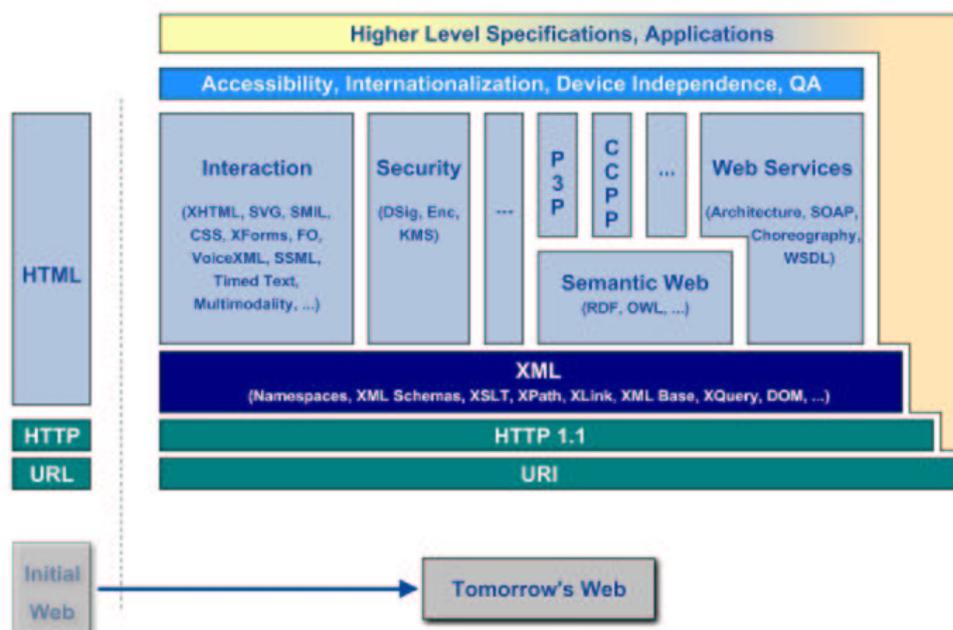


Figura 2 - L'evoluzione del Web

Decentralizzazione e architettura peer-to-peer

Internet, progettata originariamente come un sistema *peer-to-peer*, si è successivamente trasformata in un ambiente sempre più client-server, in cui milioni di utenti comunicano con un insieme relativamente limitato di server privilegiati. Le applicazioni peer-to-peer attualmente presenti in Internet utilizzano la rete essenzialmente per come era stata concepita: un *mezzo di comunicazione tra macchine che condividono risorse in modo paritetico*. I sistemi peer-to-peer sono del tutto coerenti con una architettura decentralizzata. Nei sistemi completamente decentralizzati non solo ogni host ha la stessa dignità, ma non esistono host con specifici ruoli di gestione o di supporto.

La realizzazione di sistemi completamente peer-to-peer può risultare complessa, e viene spesso adottata una architettura ibrida (un esempio tipico è la gestione del DNS, basata su

un protocollo peer-to-peer, ma con una struttura gerarchica intrinseca). In pratica, poi, alcune applicazioni operano meglio in un'architettura centralizzata, senza far ricorso a nessuna tecnologia peer-to-peer. La ricerca su grossi database, relativamente statici, è un caso in cui l'adozione di tecnologie e architetture peer-to-peer non reca vantaggi. La definizione centralizzata di concetti, dizionari, thesauri, e l'identificazione di siti privilegiati, depositari di questa conoscenza, è, invece, un processo che comporta spesso tempi lunghi, anche per superare differenze culturali o tradizioni ben consolidate. Un uso coerente delle tecnologie web consente di superare questi ostacoli.

Design for all: l'accessibilità dei siti Web

Il Web è la tecnologia che si è diffusa più rapidamente nella storia dell'uomo, e sta diventando una risorsa chiave per il *reperimento dell'informazione* (notizie, commercio, entertainment), la *formazione* (classroom education, distance learning), *lavoro* (ricerca d'impiego, interazione sul posto di lavoro), *partecipazione civica* (leggi, elezioni, informazioni governative, servizi). Eppure il Web è talvolta **non accessibile** ai disabili. Ma l'accesso universale è, secondo Tim Berners-Lee, Direttore del W3C e inventore del World Wide Web uno dei requisiti essenziali del web ("*The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.*").

La *Web accessibility* deve considerare vari tipi di disabilità, poiché il Web può presentare ostacoli a persone che abbiano limitazioni visive, uditive, fisiche, cognitive o neurologiche. Anche se non tutte le disabilità hanno impatto sulle possibilità di accesso al Web, va tenuto presente che talvolta anche l'avanzamento nell'età può comportare una combinazione di problemi (diminuzione della vista o dell'udito, riduzione della destrezza, difficoltà di memoria). La *Web accessibility* è importante anche perché milioni di persone hanno difficoltà nell'accesso al Web, e molti Governi, tra cui quello Italiano, hanno emanato linee guida per garantire l'accessibilità dei siti. La Web accessibility ha una valenza non solo *sociale*, ma anche *economica* (costituisce un mercato rilevante, dato l'elevato numero di portatori di handicap e l'aumento dell'età media) e *tecnologica*, dato che la progettazione che tiene conto dei potenziali handicap porta dei benefici a tutti gli utenti, quando si trovano in condizioni ambientali difficili (dispositivi mobili, eccessiva illuminazione, elevato rumore di fondo, banda limitata, mani e occhi impegnati). Quindi, l'accessibilità contribuisce ad una migliore progettazione per tutti gli utenti, coerentemente con uno dei principi fondamentali del Web: l'Universal Access.

La *Web Accessibility Initiative (WAI)* del W3C ha operato in modo efficace per assicurare che le tecnologie Web supportino l'accessibilità, e ha sviluppato alcune Guideline che giocano un ruolo critico nel rendere accessibile il Web: ([WCAG], [ATAG], ([UAAG])).

L'interoperabilità tecnologica

Strutturazione dell'informazione

XML. - Extensible Markup Language (XML) è nato per supportare la realizzazione delle nuove applicazioni Web, in cui i dati costituiscono un elemento essenziale (data-centric Web applications) ed è stato quindi il primo passo per assegnare una semantica ai tag e supportare le transazioni sul Web, permettendo lo scambio di informazioni tra database diversi. XML, che consente di avere viste diverse degli stessi dati e di personalizzare le

informazioni mediante opportuni agenti, è indipendente dalla piattaforma e dal linguaggio, e non a caso è stato definito "ASCII del 2000".

La sintassi XML è semplice, la sua elaborazione automatica è poco complessa, e un documento XML resta comprensibile alla lettura diretta. XML, mezzo espandibile e flessibile per modellare il Web, costituisce attualmente la tecnologia chiave di W3C, e ricopre un ruolo centrale nell'architettura del Web. Ogni nuovo linguaggio utilizzato per definire un nuovo standard deve essere descritto in XML.

La struttura formale del documento viene espressa nella DTD (*Document Type Definition*), che può essere inclusa nel documento o referenziata come risorsa esterna. Un documento XML si dice "well formed" quando rispetta le regole di scrittura; viene detto "validato" quando è coerente con la struttura definita nella DTD.

XML Namespaces. - La sempre crescente popolarità di XML determina un proliferare di applicazioni, e la conseguente possibilità che esse debbano essere utilizzate insieme. Ne consegue che è necessario poter distinguere tra i vari namespace. Un XML namespace è un *insieme di nomi, caratterizzati da un URI¹ di riferimento, utilizzati come element type e attribute name.*

Ogni *nameset* ha associato un prefisso che lo identifica, e quindi i tag sono individuati univocamente specificando il prefisso e il nome "locale". Il Namespace URL non necessariamente punta a una DTD, l'unico requisito è che punti a qualcosa che identifichi univocamente il namespace, eventualmente, quindi, anche a una descrizione informale.

XML Schema Definition. - La DTD viene espressa con una sintassi sua propria, e quindi richiede editor, parser e processor ad hoc. Inoltre, è difficile estenderla, non contempla datatype e deve supportare tutti gli elementi e attributi descritti dai namespace inclusi. Queste limitazioni sono superate da XMLSchema, che ha le stesse funzionalità delle DTD, ma offre alcuni significativi vantaggi, in quanto è espresso con la sintassi XML e include datatype, inheritance, regole di combinazione degli schema. XMLSchema fornisce anche un miglior supporto dei namespace e offre la possibilità di agganciare documentazione e informazioni semantiche. XMLSchema permette di rappresentare vincoli sui possibili valori, tipi complessi e gerarchie di tipi. In definitiva, gli XMLSchema sono strumenti molto più potenti delle DTD, e sul sito W3C sono disponibili parser, validatori, e altri strumenti utili. Utilizzare questa specifica nella realizzazione di nuove applicazioni costituisce un indubbio *investimento per il futuro.*

Presentazione dell'informazione

L'architettura di riferimento. - Le applicazioni XML si basano sulla comprensione del principio chiave che ha portato alla sua definizione: la *separazione tra contenuto e forma.*

¹ **URI** (*Uniform Resource Identifier*). è il generico insieme di tutti i nomi/indirizzi che costituiscono le brevi sequenze di caratteri che fanno riferimento ad una risorsa.

URL (*Uniform Resource Locator*) è un termine informale, non più utilizzato nelle specifiche tecniche, associato con gli schemi URI più noti e diffusi (http, ftp, mailto, etc.).

URN (*Uniform Resource Name*) può essere:

- un URI che ha una valenza istituzionale per quanto riguarda la persistenza, la disponibilità, etc. Un URN può anche essere un URL (p. es. si veda <http://purl.oclc.org/>).
- Un particolare schema, urn, specificato dalla RFC2141 (<http://www.ietf.org/rfc/rfc2141.txt>) e documenti correlati, inteso per essere utilizzato come identificatore di risorsa persistente e indipendente dalla locazione.

Per ulteriori dettagli si veda [Naming]

Nell' architettura comune di riferimento (Figura 3) che ne consegue, le informazioni, estratte dalla base dati aziendale, vengono strutturate in un documento XML, che viene successivamente trasformato nel formato più adatto per l' utente finale, mediante una trasformazione di stile ([XSL], [XSLT]). Si noti che la presentazione dell' informazione non riveste un ruolo semplicemente formale ed estetico, ma investe aspetti di distribuzione dell' informazione (per es. consentendo l' accesso all' informazione in maniera indipendente dal dispositivo utilizzato) e semantici. In particolare, non si può ignorare che la forma in cui viene presentata l' informazione porta intrinsecamente un messaggio preciso, legato alle specifiche tradizioni culturali.

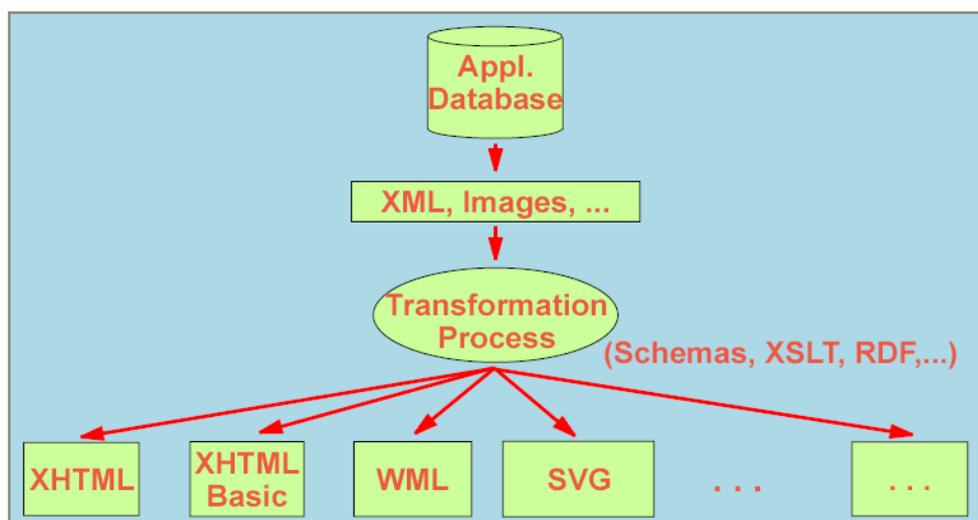


Figura 3 - L' architettura di riferimento

Multimedialità, matematica, grafica. - L' arricchimento dell' informazione presentata all' utente è anche legata alla possibilità di corredarla di componenti multimediali o grafiche (si pensi, banalmente, ad un thesaurus corredato di informazioni grafiche, al video che illustra un certo tipo di processo, etc.).

Synchronized Multimedia Integration Language (SMIL, pronunciato come la parola inglese "smile") permette in modo semplice la creazione ("authoring") di presentazioni interattive audiovisive. SMIL viene utilizzato tipicamente per presentazioni "rich media"/multimedia, che integrano streaming audio e video con immagini, testi, o altri tipi di media. In termini molto semplici, permette agli autori di specificare cosa deve essere presentato e quando. L' idea di base è quella di dare un nome, con un URI, ai vari componenti, qualunque sia il medium utilizzato (testo, immagine, video, audio) e programmare la loro presentazione in sequenza o in parallelo. La presentazione è costituita da vari componenti di tipo diverso (audio, video, immagine, testo), memorizzati su un Web server, quindi accessibili mediante un URI, e l' utente può seguire eventuali hyperlink inclusi nella presentazione.

Sono ovviamente disponibili tutte le funzioni tipiche degli ambienti multimediali. Il linguaggio SMIL è stato progettato in modo che una presentazione possa essere preparata utilizzando un banale editore di testi, raccogliendo così l' eredità di HTML, che permette di

creare pagine ipertestuali senza far ricorso a strumenti sofisticati. Esempi classici di applicazioni SMIL sono: archivi di foto digitalizzate coordinate con una loro presentazione, corsi di formazione con integrazione di voce e immagini, etc.

Formule matematiche o grafici contengono molta conoscenza che è difficilmente indicizzabile e reperibile. Formule semplici possono essere facilmente inserite nella pagine HTML, utilizzando apici e pedici (per esempio, a_i verrebbe espresso come: `a_i`). Per formule più complesse (anche la banale formula risolutiva dell'equazione di secondo grado) bisogna però ricorrere ad artifici: tipicamente, rappresentare la formula come immagine, e inserirla poi nel testo. Evidenti, e noti, i problemi di posizionamento delle formule. Inoltre, la formula viene vista come un elemento singolo, nessuna sua parte è individuabile separatamente, la sua presentazione ad un portatore di handicap visivo potrebbe essere problematica. Infine, non è trascurabile il costo di una eventuale trasposizione del materiale esistente in un nuovo formato utilizzabile sul Web.

```

<mrow>
  <mi>x</mi> <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow><mo>-</mo><mi>b</mi></mrow>
      <mo>&PlusMinus;</mo>
      <msqrt>
        <mrow>
          <msup><mi>b</mi><mn>2</mn></msup>
          <mo>-</mo>
          <mrow>
            <mn>4</mn><mo>&InvisibleTimes;</mo>
            <mi>a</mi><mo>&InvisibleTimes;</mo>
            <mi>c</mi>
          </mrow>
        </mrow>
      </msqrt>
    </mrow>
  <mrow>
    <mn>2</mn><mo>&InvisibleTimes;</mo>
    <mi>a</mi>
  </mrow>
</mfrac>
</mrow>

```

Figura 4 - Una formula espressa inMathML

MathML ([MathML]) è un linguaggio di marcatura che permette di scrivere formule matematiche anche molto complesse (la Figura 4 è un esempio di scrittura di una formula semplice). Le formule sono espresse in XML, e quindi è possibile ricercarne i singoli elementi, si possono mescolare le formule con altri markup, un voice browser potrebbe essere in grado di leggere la formula. In altri termini, le *informazioni semantiche* presenti nella formula possono essere rese esplicite e condivise con altri.

Per la grafica vettoriale esiste una specifica, SVG (**S**calable **V**ector **G**raphics). SVG è un linguaggio per descrivere grafici bidimensionali in XML. SVG gestisce tre tipi di oggetti grafici: forme in grafica vettoriale (per es. cammini, o path, costituiti da linee rette e curve),

immagini e testi. Gli oggetti grafici SVG hanno la proprietà di essere scalabili, con componenti identificabili singolarmente, che possono essere corredati di *descrizioni semantiche (metadati)*, ed essere origine o destinazione di link. Il linguaggio SVG è molto ricco, e consente anche animazioni.

Uno scenario di interazione uomo-macchina: XForms

A 10 anni dalla sua nascita, il disegno originale dei form HTML sta mostrando i suoi limiti. Oggi gli utenti vogliono accedere al Web con i telefoni cellulari, con i palmari e tecnologie assistive, come gli screen reader, mentre gli autori hanno bisogno di maggiori funzionalità e sono interessati a minimizzare lo scripting, a massimizzare il riuso delle componenti di un form, e a separare in maniera chiara lo *scopo*, la *presentazione* e *risultati* di un form. Inoltre, le aziende che sono passate a XML, sicuramente sono interessate ad integrare i form nei loro processi d'impresa.

L' XForms Working Group ha fornito un modello che rende semplice agli implementatori sviluppare e riusare componenti di form, integrarli all' interno di Web Service e dare delle funzionalità precedentemente impossibili agli utenti e ai dispositivi.

A differenza dei form HTML, nei quali il markup funzionale e di presentazione erano mescolati, XForms permette agli autori di form di distinguere la descrizione dello scopo del form, la presentazione del form e il modo in cui i risultati (i dati dell' istanza) sono scritti in XML. Tale separazione porta nuovi vantaggi:

- *Riuso*: i moduli di XForms possono essere riusati indipendentemente dall' informazione che raccolgono
- *Indipendenza dal dispositivo*: i controlli dell' interfaccia utente sono astratti, cioè sono indicate solamente caratteristiche generiche, in modo da poter essere inviati facilmente a dispositivi differenti con caratteristiche diverse.
- *Accessibilità*: la separazione della presentazione dal contenuto permette una maggiore leggibilità per utenti di tecnologie assistive; inoltre, i controlli dell' interfaccia utente incapsulano tutti i metadati rilevanti, migliorando quindi l' accessibilità dell'applicazione nel caso in cui si usino diverse modalità di interazione.

In pratica, le tecnologie XForms rendono possibile l' invio dello stesso form ad un PDA, ad un telefono cellulare, ad uno screen reader o ad un PC convenzionale, senza perdita di funzionalità per l' utente finale.

XForms permette agli autori di specificare proprietà di, e relazioni tra, valori che devono esser raccolti. Ad esempio, XForms permette di specificare che un particolare campo deve essere un indirizzo email, che il campo quantità totale è la somma dei singoli elementi nelle righe, oppure che un numero di carta di credito non deve essere richiesto se la modalità di pagamento è per contante. Queste cose sono specificate usando semplici proprietà come l' affermazione che un campo è “required”, oppure fornendo il tipo del campo, anziché usando in maniera estensiva lo scripting che era necessario nei tradizionali form HTML.

Questo comporta un netto miglioramento dell' interazione utente, spostando parte dell' elaborazione sul client.

XML è il cuore del modello di XForms, alla cui tecnologia dà vantaggi significativi:

- I dati ricevuti da un XForm sono sempre fortemente tipati, ben formati, facili da validare e da elaborare, in altre parole, sono XML.
- Usando XML 1.0 per la descrizione dei risultati, detti “dati dell' istanza”, si assicura che i dati inviati possano essere facilmente internazionalizzati.

- XForms può essere usato per “compilare” qualsiasi documento XML.
- Gli XML schema di processi d' impresa già esistenti possono essere usati facilmente per la validazione dei dati dell' istanza.
- Gli XML schema possono essere riusati attraverso XForms, aiutando a mantenere insiemi di form aggiornati e consistenti.
- XForms può parlare con i Web service, integrando infine l' utente nel processo dei Web service.

Inoltre, XForms, progettato inizialmente per essere integrato in XHTML, può essere adottato da ogni linguaggio di markup adatto, come lo Scalable Vector Graphics (SVG). XForms utilizza XML Events, un' altra tecnologia W3C divenuta una Recommendation, per definire *event handler* dichiarativi basati su XML che coprono casi d' uso comuni, in modo che la maggior parte dei documenti XForms possa essere analizzata staticamente, riducendo la necessità di scripting complicato per gli event handler.

Esistono già molte implementazioni di Xforms. Le implementazioni attuali possono consegnare lo stesso form a diversi dispositivi, includendo telefoni cellulari, PDA, voice browser, PC, ed anche usando client di instant messenger. Stanno emergendo alcune grandi comunità di utenti, in particolare lo “United Kingdom e-government interoperability framework” richiama espressamente Xforms.

XForms Basic, il profilo mobile di XForms che permette a XForms di essere implementato in maniera nativa su dispositivi mobili, è attualmente una W3C Candidate Recommendation. Ci si attende che diventi una Recommendation quando una implementazione mobile supererà la suite di test di XForms.

L' interoperabilità semantica

Un problema tipico

La ricerca di informazioni è uno dei principali punti deboli del web, nonostante il gran numero di motori di ricerca esistenti, che sono poveri di semantica sia in fase di indicizzazione che in fase di ricerca. In fase di indicizzazione, essi utilizzano o moduli compilati dai fornitori di informazioni, che spesso non consentono di specificare metainformazioni come l' autore, le parole chiave, etc., o strumenti automatici (*spider*) per accedere alle pagine ed estrarre semantica. Talvolta, anche le informazioni contenute nei tag <meta> vengono di fatto ignorate. In fase di ricerca, viene consentito di combinare le parole con operatori di contesto (“tutte le parole”, “una parola qualunque”, “nel titolo”), ma in definitiva il risultato scaturisce sempre da una ricerca sulla presenza di parole chiave e dall' identificazione dei documenti più affini alla domanda posta.

L' esistenza di proposizioni più ricche dal punto di vista espressivo permette invece agli utenti di ritrovare in maniera più facile ed efficace le informazioni necessarie per risolvere i problemi, e consente di indicizzare le risorse esistenti sul web in maniera più ricca rispetto al metodo tradizionale di associare alle risorse parole chiave o concetti. Diventa allora possibile formulare richieste più sofisticate, migliorando sia la precisione delle risposte ottenute, che il richiamo dei documenti pertinenti.

I metadati

Nel navigare sul web, si seguono dei link, che portano a quella che formalmente viene detta *risorsa* (resource) identificata univocamente da un URI. Nel linguaggio corrente una risorsa viene anche detta “documento”, per mettere in evidenza il fatto che sia leggibile da un

essere umano, o “oggetto”, per mettere in evidenza che è leggibile da una macchina. Qualunque sia il termine utilizzato, la risorsa non è una entità a sé, ma è accompagnata da informazioni che la descrivono. Le informazioni sulla risorsa vengono generalmente dette Metadati.

Si può quindi dire che *i metadati sono informazioni, comprensibili dalla macchina, relative a una risorsa web o a qualche altra cosa*. Il punto chiave è costituito dal fatto che i metadati sono comprensibili dalla macchina (*machine understandable*), e quindi costituiscono un tipo di informazione che può essere utilizzata dai *software agent*, per fare un uso appropriato delle risorse. Il Web, come insieme di risorse e di informazioni sulle risorse (cioè metadati) è già una realtà alla quale siamo abituati.

Va tenuto presente che *i metadati sono dati*, e perciò possono essere *memorizzati come dati*, (nella risorsa stessa o in un’altra risorsa), e possono essere *descritti da altri metadati*, e così via. Il numero di metalivelli da specificare dipende dalle caratteristiche delle applicazioni e dalle specifiche esigenze.

Il Resource Description Framework

I *metadati* sembrano offrire una soluzione al problema di automatizzare il Web restando ancorati alla sua architettura originaria, in cui tutte le informazioni erano *machine-readable*, ma non *machine-understandable*. L’uso efficace dei metadati, tuttavia, richiede che vengano stabilite delle convenzioni per la *semantica*, la *sintassi* e la *struttura*. Coerentemente con l’assunzione di fondo di una architettura peer-to-peer, le *single comunità* interessate alla descrizione delle loro risorse specifiche definiscono la semantica dei metadati pertinenti alle loro esigenze. La sintassi, cioè l’organizzazione sistematica dei data element per l’elaborazione automatica, facilita lo scambio e l’utilizzo dei metadati tra applicazioni diverse. La struttura può essere vista come un vincolo formale sulla sintassi, per una rappresentazione consistente della semantica.

Resource Description Framework (RDF) è lo strumento base per la codifica, lo scambio e il riutilizzo di metadati strutturati, e consente l’interoperabilità tra applicazioni che si scambiano sul Web informazioni machine-understandable. RDF consente l’elaborazione automatica delle risorse reperibili sul web, e può essere utilizzato e portare vantaggi in molti settori, quali, ad esempio, la *descrizione e classificazione del contenuto* di un sito Web, l’implementazione di *intelligent software agent*, stabilire i criteri di *proprietà intellettuale* delle singole pagine, esprimere criteri di *privacy preference* degli utenti e le *privacy policies* di un sito Web, tanto per citarne qualcuno.

RDF fornisce una base comune per poter esprimere la semantica, permettendo di definire la semantica dei tag XML, ed è costituito da due componenti: *RDF Model and Syntax* ([RDFMSS]), che definisce il *data model* RDF e la sua codifica XML, e *RDF Schema* ([RDFSS]) che permette di definire specifici *vocabolari* per i metadati.

RDF fornisce quindi un modello per descrivere le risorse, che hanno delle proprietà (o anche attributi o caratteristiche). RDF definisce una risorsa come un qualsiasi oggetto che sia identificabile univocamente mediante un Uniform Resource Identifier (URI).

Il data model RDF, che consente di rappresentare statement RDF in modo sintatticamente neutro, è molto semplice ed è basato su tre tipi di oggetti:

Resources Qualunque cosa descritta da una espressione RDF viene detta risorsa (resource).

Properties Una *property* (proprietà) è un aspetto specifico, una caratteristica, un attributo, o una relazione utilizzata per descrivere una risorsa. Le

proprietà associate alle risorse sono identificate da un *nome*, e assumono dei *valori*.

Statements Una risorsa, con una proprietà distinta da un nome, e un valore della proprietà per la specifica risorsa, costituisce un RDF *statement*. Uno *statement* è quindi una tupla composta da un *soggetto* (risorsa), un *predicato* (proprietà) e un *oggetto* (valore). L'oggetto di uno *statement* (cioè il *property value*) può essere un'espressione (sequenza di caratteri o qualche altro tipo primitivo definito da XML) oppure un'altra risorsa.

Graficamente, le relazioni tra Resource, Property e Value vengono rappresentate mediante *grafi etichettati orientati*, in cui le risorse vengono identificate come nodi (graficamente delle ellissi), le proprietà come archi orientati etichettati, e i valori corrispondenti a sequenze di caratteri come rettangoli. Un insieme di proprietà che fanno riferimento alla stessa risorsa viene detto *descrizione* (*description*).

Namespace: un elemento chiave nell'architettura peer-to-peer

RDF supporta l'utilizzo di convenzioni che rendono più agevole l'*interoperabilità tra insiemi separati di metadati*. Queste convenzioni includono l'utilizzo di meccanismi standard per rappresentare la semantica, basati sul modello, semplice ma potente, illustrato precedentemente. Inoltre, RDF consente di pubblicare vocabolari *machine readable*, ma anche leggibili da utenti umani. I vocabolari sono un insieme di proprietà (o *metadata elements*) definiti dalle singole comunità disciplinari. Uno degli esempi più noti è la Dublin Core Initiative ([DC]). La capacità di standardizzare le definizioni dei vocabolari potrebbe favorire enormemente il riuso e l'estensione della semantica tra comunità diverse.

RDF identifica univocamente le proprietà mediante il meccanismo dei namespace XML ([XMLns]), che forniscono un metodo per identificare in maniera non ambigua la semantica e le convenzioni che regolano l'utilizzo delle proprietà identificando l'*authority* che gestisce il vocabolario.

Si può utilizzare un namespace XML per identificare in maniera non ambigua lo schema per il vocabolario Dublin Core puntando alla risorsa Dublin Core che ne definisce la semantica. Un sito Web potrebbe quindi essere descritto mediante tutte o parte delle proprietà definite nel vocabolario Dublin Core e le proprietà definite in altri vocabolari, compresi quelli eventualmente definiti ad hoc.

Lo scenario del Semantic Web

La sfida attuale, su cui l'intera comunità scientifica sta investendo molte energie ([SemWeb]), è il Semantic Web, la cui architettura a livelli è riportata in Figura 5.

Va ricordato che la filosofia di base del Web è quella di uno spazio informativo universale, navigabile, con un mapping da *URI* (*Uniform Resource Identifier*) alle risorse. Nel contesto del Semantic Web, il termine semantico assume la valenza di "*elaborabile dalla macchina*" e non intende fare riferimento alla semantica del linguaggio naturale e alle tecniche di intelligenza artificiale. Il Semantic Web è, come l'XML, un *ambiente dichiarativo*, in cui si specifica il significato dei dati, e non il modo in cui si intende utilizzarli. La semantica dei dati consiste nelle informazioni utili perché la macchina possa utilizzarli nel modo corretto, eventualmente convertendoli.

Il Semantic Web ([TBL2001]) potrà funzionare solo se le macchine potranno accedere ad un *insieme strutturato di informazioni* e a un insieme di *regole di inferenza* da utilizzare per

il ragionamento automatico. La sfida del semantic web, quindi, è fornire un linguaggio per esprimere *dati* e *regole* per ragionare sui dati, che consenta l' *esportazione sul web* delle regole da qualunque sistema di rappresentazione della conoscenza.

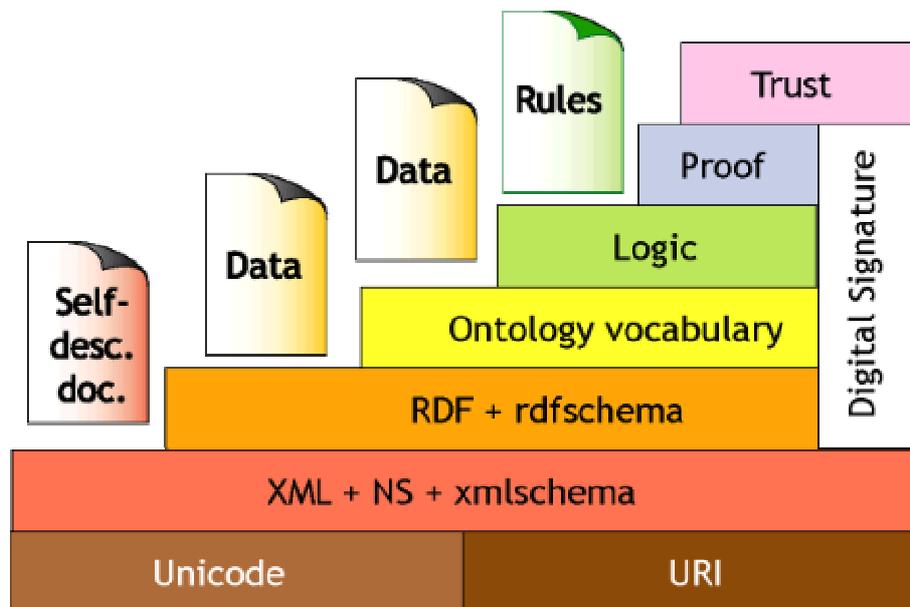


Figura 5 - L' architettura del Semantic Web
<http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>

Lo schema in Figura 5, evidenzia il ruolo di base giocato da **XML** (con *Name Space* e *xmlschema*), che consente di dare ai documenti una *struttura* arbitraria, mentre **RDF** può essere usato per esprimere il *significato*, asserendo che alcuni particolari elementi hanno delle proprietà (p.es. *autore-di*).

Un terzo componente è l' *Ontology Vocabulary* (*livello ontologico*), inteso come il contenitore che definisce in modo formale le relazioni fra i termini. Una ontologia permette di descrivere le relazioni tra i tipi di elementi (per es. "questa è una proprietà transitiva") senza però fornire informazioni su come utilizzare queste relazioni dal punto di vista computazionale. Le ontologie possono svolgere un ruolo fondamentale nel migliorare il funzionamento del Web (ricerca di concetti, collegamento delle informazioni contenute in una pagina alle strutture di conoscenza associate, etc.).

Il linguaggio definito dal W3C per definire ontologie strutturate, in architettura web, per consentire una migliore integrazione dei dati tra applicazioni in settori diversi è **OWL** (*Ontology Web Language*).

Il *livello logico* è il livello immediatamente superiore al livello ontologico. A questo livello le asserzioni esistenti sul Web possono essere utilizzate per derivare nuova conoscenza.

La firma digitale (*digital signature*) è di significativa importanza in diversi strati nel modello astratto del Semantic Web. È necessaria infatti una infrastruttura in cui le parti possano essere riconosciute e accettate come credibili in specifici domini. Con una

granularità fine come questa, la firma digitale potrebbe essere utilizzata per stabilire la provenienza delle ontologie e delle deduzioni, oltre che dei dati.

Uno scenario di interoperabilità: i Web Service

Definizione e architettura di un Web Service

Una possibile definizione di WEB Service è la seguente:

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

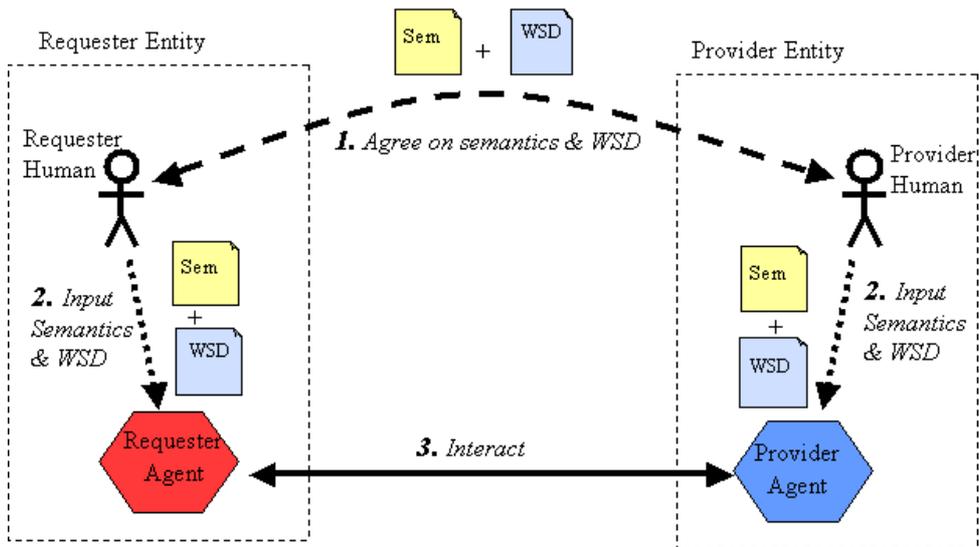


Figura 6 - I vari ruoli nell' architettura dei Web Service (da [WSA])

Agenti e Servizi. - Un Web Service è una nozione astratta che deve essere implementata da un agent reale. L' *agent* è quindi l' *entità concreta* (il software) che spedisce e riceve messaggi, mentre il *service* è l' *insieme astratto di funzionalità fornite*. L' obiettivo di un Web Service è fornire alcune funzionalità per conto del suo proprietario (una persona o un' organizzazione). La *provider entity* è la persona o organizzazione che fornisce un idoneo agent per implementare un particolare servizio. (vedi Figura 6).

Una *requester entity* è una persona o organizzazione che intende utilizzare il web Service di una provider entity. A tale scopo, utilizzerà un *requester agent* per scambiare messaggi con il *provider agent* della provider entity. Questo scambio di messaggi potrà avere successo solo se provider e requester entity si saranno accordate sulla *semantica* e la *meccanica* dello scambio di messaggi, documentati in una Web Service Description

(WSD). La WSD è una specifica *machine processable* dell'interfaccia, e definisce formato, datatype, protocollo di trasporto, locazioni di rete ove può essere invocato il servizio, etc.

La semantica dello scambio di messaggi rappresenta il "contratto" tra requester e provider entity riguardo allo scopo e conseguenze dell'interazione

In altri termini, la descrizione del servizio (WSD) rappresenta un contratto che governa l'*aspetto meccanico* dell'interazione con un particolare servizio, la semantica rappresenta un contratto che regola il *significato e lo scopo* dell'interazione.

Il ruolo degli operatori umani. – Sebbene uno degli obiettivi principali dei Web Service sia l'automazione di processi che altrimenti sarebbero manuali, la componente umana gioca ancora un ruolo fondamentale, principalmente in due modi:

1. Gli esseri umani devono concordare sulla semantica e sulla descrizione dei servizi. Considerato che alla fin fine è sempre un essere umano (o una organizzazione) a detenere il possesso di un Web Service, le persone devono raggiungere, implicitamente o esplicitamente, un *accordo sulla semantica e sulla descrizione del servizio che governa l'interazione*.
2. Gli esseri umani direttamente o indirettamente creano i requester e provider agent, e, in definitiva, devono garantire che questi agenti *implementino la descrizione e la semantica del servizio concordato*.

Indipendentemente dall'approccio adottato, dal punto di vista informativo è necessario che la semantica e la descrizione del servizio vengano in qualche modo forniti o incorporati nei due agenti prima che essi possano interagire.

SOAP. - In estrema sintesi, SOAP 1.2 ([SOAP12]) fornisce la definizione di informazione XML-based che può essere utilizzata per scambiare informazione strutturata e tipata tra componenti paritarie in un ambiente decentralizzato e distribuito. SOAP è fondamentalmente un paradigma di scambio messaggi stateless e unidirezionale, ma le applicazioni possono creare modelli di interazione più complessi.

Un messaggio SOAP è composto come in Figura 7: una *envelope* che contiene due elementi: un Header (opzionale) e un Body.

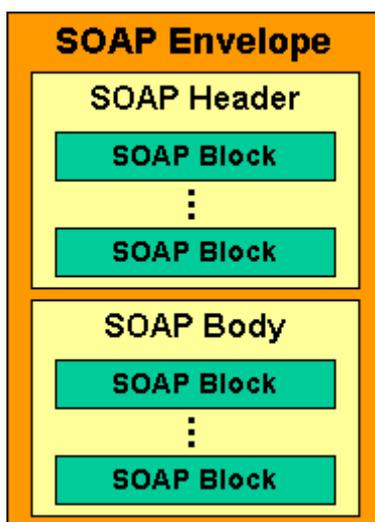


Figura 7 – Struttura di un messaggio SOAP

L' *Header* è opzionale, ed è stato progettato in previsione dei vari utilizzi di SOAP, molti dei quali potrebbero coinvolgere altri nodi (gli intermediari SOAP) che elaborino il messaggio SOAP nel suo cammino dal mittente al destinatario finale. I figli di primo livello del SOAP Header sono gli *header block*, che rappresentano raggruppamenti logici di dati che possono essere indirizzati a intermediari previsti nel cammino del messaggio. Il *Body*, con i suoi figli, è obbligatorio, e contiene l' informazione principale da trasmettere tra il mittente e il destinatario, e quindi si suppone implicitamente che il contenuto dei suoi figli (i *body block*) sia comprensibile ed elaborabile dal destinatario finale.

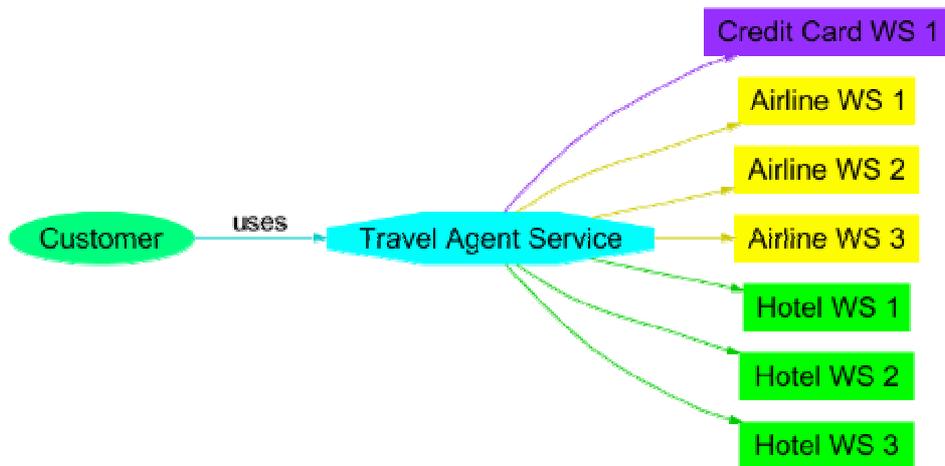


Figura 8 - Uno scenario d'uso (da [WSA-US])

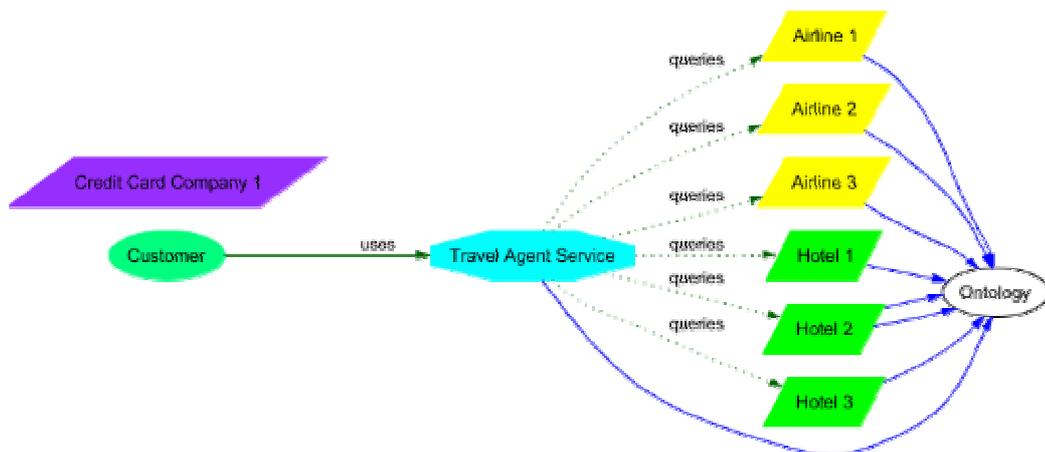


Figura 9 – Il ruolo di un'ontologia in uno Scenario d' uso (da [WSA-US])

Un esempio

Lo scenario d' uso di Figura 8 descrive come un utente potrebbe prenotare un pacchetto vacanze (volo e pernottamento).

Questo scenario d'uso presuppone che tutti i servizi facciano riferimento a concetti comuni (volo, classe economica, pernottamento, etc.). Perché il servizio dell'agenzia di viaggi possa comprendere i servizi della compagnia aerea e inviare l'informazione corretta, è necessario che sia stata definita, e venga usata sul Web, una *ontologia per il turismo* (Figura 9). Ricordiamo che un'ontologia è una descrizione formale di un insieme di concetti e delle loro relazioni, quindi, associando un nome ad ogni concetto, un'ontologia definisce il vocabolario standard da utilizzare per comunicare questi concetti.

Conclusioni

Nell'evoluzione del web verso il Semantic Web va assumendo sempre maggiore importanza l'interoperabilità, sia tecnologica che semantica. La famiglia di tecnologie XML gioca un ruolo essenziale nei vari livelli architetturali.

A livello di *markup*, XML consente l'*interoperabilità nel contesto delle applicazioni*.

A livello dei *dati*, RDF è l'elemento chiave per l'*interoperabilità tra applicazioni*.

Infine, al livello *ontologico*, linguaggi come OWL consentono di perseguire l'obiettivo del *web of meaning*.

Ringraziamenti

Un doveroso ringraziamento va a tutti quelli che mi hanno aiutato nella preparazione di questo documento, con utili suggerimenti e discussioni chiarificatrici (in particolare a Jeremy J. Carroll e Silvia Martelli). Un particolare ringraziamento va al W3C Team, che mantiene sul sito documentazione aggiornata. Parte del contenuto di questo lavoro proviene direttamente dal materiale reperibile sul sito.

Bibliografia

- [ATAG] *Authoring Tool Accessibility Guidelines 1.0*,
<http://www.w3.org/TR/WAI-AUTOOLS/>
- [DC] The Dublin Core Home Page, URL: <http://dublincore.org/>
- [IRDF] Introduction to RDF Metadata, W3C NOTE 1997-11-13, Ora Lassila,
URL: <http://www.w3.org/TR/NOTE-rdf-simple-intro>
- [MathML] W3C's Math Home Page, <http://www.w3.org/Math/>
- [Miller1998] Miller E.: An Introduction to the Resource Description Framework, D-Lib Magazine, May 1998, <http://www.dlib.org/dlib/may98/miller/05miller.html>
- [Naming] Naming and Addressing: URIs, URLs, ..., <http://www.w3.org/Addressing/>
- [RDFMSS] O.Lassila, R.Swick: *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation 22 February 1999,
<http://www.w3.org/TR/REC-rdf-syntax>
- [RDFSS] *Resource Description Framework (RDF) Schema Specification*, W3C Recommendation 03 March 1999, <http://www.w3.org/TR/1999/PR-rdf-schema-19990303>
- [SemWeb] <http://www.semanticweb.org/>

- [Signore2001] Signore O.: *Il ruolo centrale di XML nell'evoluzione del Web*, XML Day Milan, Conference proceedings, Milan, September 21 (find this and other similar papers at: <http://www.w3c.it/papers/>)
- [TBL1997] Tim Berners-Lee: *Metadata architecture*, (1997), <http://www.w3.org/DesignIssues/Metadata.html>
- [TBL1998] Tim Berners-Lee: *Semantic Web Road Map*, (1998), <http://www.w3.org/DesignIssues/Semantic.html>
- [TBL1999] Tim Berners-Lee: *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*, HarperSanFrancisco (1999), ISBN 0-06-251587-X
- [TBL2001] Berners-Lee T., Hendler J., Lassila O.: *The Semantic Web*, Scientific American, May 2001, <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>
- [UAAG] *User Agent Accessibility Guidelines 1.0*, <http://www.w3.org/TR/WAI-USERAGENT/>
- [W3C] *World Wide Web Consortium Home Page*, <http://www.w3.org>
- [WA-Policies] *Policies Relating to Web Accessibility*, <http://www.w3.org/WAI/Policy/>
- [WCAG] *Web Content Accessibility Guidelines 1.0*, <http://www.w3.org/TR/WAI-WEBCONTENT/>
- [WSA] *Web Services Architecture - W3C Working Draft 8 August 2003*, <http://www.w3.org/TR/ws-arch/>
- [WSA-US] *Web Services Architecture Usage Scenarios - W3C Working Draft 14 May 2003*, <http://www.w3.org/TR/ws-arch-scenarios/>
- [XML] *Extensible Markup Language (XML)*, <http://www.w3.org/XML/>
- [XML1.0] *Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation 6 October 2000*, <http://www.w3.org/TR/2000/REC-xml-20001006>
- [XML-Events] <http://www.w3.org/TR/2003/REC-xml-events-20031014/>
- [XMLns] *Namespaces in XML - World Wide Web Consortium 14-January-1999* <http://www.w3.org/TR/REC-xml-names/>
- [XMLschema0] *XML Schema Part 0: Primer - W3C Recommendation - 2 May 2001* <http://www.w3.org/TR/xmlschema-0/>
- [XMLschema1] *XML Schema Part 1: Structures - W3C Recommendation - 2 May 2001* <http://www.w3.org/TR/xmlschema-1/>
- [XMLschema2] *XML Schema Part 2: Datatypes - W3C Recommendation - 2 May 2001* <http://www.w3.org/TR/xmlschema-2/>
- [XSL] *Extensible Stylesheet Language (XSL), Version 1.0 W3C Recommendation 15 October 2001* <http://www.w3.org/TR/2001/REC-xsl-20011015/xslspecRX.pdf>
- [XSLT] *XSL Transformations (XSLT) Version 1.0, W3C Recommendation 16 November 1999*, <http://www.w3.org/TR/xslt>