

Il ruolo centrale di XML nell'evoluzione del Web

Oreste Signore

Ufficio Italiano W3C presso il C.N.R. - Istituto CNUCE
Area della Ricerca di Pisa San Cataldo - Via G. Moruzzi, 1 - 56124 Pisa
Email:oreste@w3.org

Sommario

Uno degli obiettivi a lungo termine del World Wide Web Consortium, che definisce regole e strategie di evoluzione del Web, è il Semantic Web. Anche se molto lavoro è ancora da compiere per raggiungere questo ambizioso obiettivo, alcuni elementi esistono già, e sono parte della famiglia di tecnologie XML. Questo lavoro contiene una descrizione sommaria del W3C, di alcune specifiche e di alcuni aspetti significativi in cui l'adozione di XML permette una migliore fruizione dell'informazione. Infine, vengono discusse le prospettive future, in particolar modo per quanto riguarda il Semantic Web.

Il World Wide Web Consortium

Il World Wide Web Consortium (W3C), è un consorzio che sviluppa tecnologie interoperabili (specifiche, linee guida, software, e strumenti) per portare il Web al massimo del suo potenziale come forum per l'accesso all'informazione, il commercio, le comunicazioni e la creazione di una cultura comune. Il W3C (<http://www.w3.org>), guidato da Tim Berners-Lee, Direttore e creatore del World Wide Web, e Jean-François Abramatic, Presidente, è stato costituito nell'ottobre 1994 con lo scopo di sviluppare al massimo il potenziale del World Wide Web, definendo protocolli comuni che ne favoriscano l'evoluzione e assicurino l'interoperabilità. È un consorzio internazionale di imprese, ospitato congiuntamente da Massachusetts Institute of Technology Laboratory for Computer Science (MIT/LCS) negli Stati Uniti, Institut National de Recherche en Informatique et en Automatique (INRIA) in Europa, Keio University Shonan Fujisawa Campus in Giappone.

Il W3C, finanziato dai membri, è *neutrale* rispetto ai venditori, e opera con tutta la comunità per produrre specifiche e software di riferimento reso disponibile gratuitamente in tutto il mondo. Il funzionamento del consorzio è regolato da un insieme di regole, contenute nel *Process Document* ([W3CPD]), che viene periodicamente verificato e adeguato, dietro accettazione da parte dei membri, alle esigenze emergenti. Un aspetto essenziale è che le decisioni vengono prese a seguito di un processo che prevede il raggiungimento del consenso dei partner. Questo significa che, anche se non sempre è possibile raggiungere l'unanimità, si ha comunque cura di non prendere decisioni su cui non ci sia accordo da parte di una vasta maggioranza. Tutte le osservazioni vengono valutate dal punto di vista tecnico.

Gli *obiettivi a lungo termine* del W3C sono coerenti con le motivazioni iniziali che hanno portato alla nascita del web. Essi 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 W3C Recommendation

Le W3C Recommendation sono il risultato di un processo lungo e cooperativo, regolato dal Process Document, che prevede una serie di passi e di documenti prodotti: Alcuni documenti sono riservati ai partecipanti ai gruppi di lavoro, altri sono disponibili per i membri, che votano per approvarli o

modificarli, altri sono pubblici.

Anche se *Network Computing* cita il W3C nella sua lista dei "Ten Most Significant Standards Groups", dal punto di vista formale il W3C non è un organo di standardizzazione. Tuttavia, va riconosciuto che il W3C è una comunità di membri che cooperano spontaneamente per definire le linee guida e le specifiche, verificando che esse siano realmente implementabili, e mantiene stretti contatti con gli organi di standardizzazione e con gli User Forum.

La partecipazione al W3C

I membri del W3C, ad oggi, sono più di 500 ([W3CML]), e comprendono organizzazioni di vario tipo, che collaborano allo sviluppo del Web¹. I vantaggi dell' associazione al W3C sono riconducibili a un guadagno in termini di maggiore informazione, miglioramento di immagine, coinvolgimento attivo e possibile influenza nell' evoluzione del Web, maggiore prontezza nel seguire l' evoluzione del mercato. Il punto essenziale dell' associazione al W3C consiste proprio nella partecipazione attiva allo sviluppo del Web, assicurando:

- Maggiore conoscenza dei processi evolutivi e possibilità di partecipare ai processi decisionali
- Conoscenza anticipata dell' evoluzione delle raccomandazioni, e quindi opportunità per svolgere attività tecnologica di punta, sviluppando prototipi in fase con lo sviluppo tecnologico e realizzando prima degli altri applicazioni e prodotti conformi alle raccomandazioni

Per migliorare la sua presenza e i rapporti con le singole comunità nazionali, il W3C ha creato un certo numero di W3C Offices (<http://www.w3.org/Consortium/Offices/>). Gli uffici al momento sono 10, e si trovano in Australia, Germania, Grecia, Hong Kong, Inghilterra, Israele, Italia, Olanda, Marocco, Svezia.

L' Ufficio Italiano W3C (<http://w3c.it>), che ha iniziato la sua attività nel 1999, sta dando vita ad una nuova iniziativa, denominata *WebLab*, che intende rafforzarne il ruolo tecnico scientifico, agendo da punto di incontro tra le attività di ricerca e l' integrazione e sviluppo di tecnologie Web.

I cinque "Domain"

Per una sua migliore organizzazione interna, il W3C ha individuato cinque² domini di attività, che hanno comunque una significativa interazione tra di loro.

I quattro domini sono:

- *Architecture*. Sviluppa le tecnologie di base del Web, potenziandone l' infrastruttura e migliorandone l' automazione.
- *Document Formats*. Opera sui formati e i linguaggi utilizzati per presentare le informazioni agli utenti in modo accurato, piacevole e con un adeguato livello di controllo.
- *Interaction*. Mira a migliorare l' interazione degli utenti con il Web. Questo comporta sviluppi sui formati e i linguaggi per presentare l' informazione con maggiore accuratezza e più elevato livello di controllo.
- *Technology and Society*. Cerca di sviluppare l' infrastruttura del Web per affrontare i problemi di tipo sociale e politico. La crescita esponenziale del Web ha costretto la comunità del Web a considerare gli aspetti etici e legali in una prospettiva internazionale. Il Technology and Society Domain cerca di comprendere questi problemi alla luce della nuova tecnologia, sia modificandola, sia formando gli utenti su benefici, costi e limiti della tecnologia.
- *Web Accessibility*. L' impegno del W3C per portare il Web al massimo del suo potenziale prevede la promozione di un elevato grado di usabilità per le persone portatrici di handicap. La Web Accessibility Initiative (WAI) persegue l' obiettivo di una maggiore accessibilità operando in cinque settori: tecnologia, linee guida, strumenti, formazione e promozione, ricerca e sviluppo.

I principi informatori del Web

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

¹ Al momento, i membri italiani sono dieci: C.N.R, CSP di Torino, Presidenza del Consiglio dei Ministri, Telecom Italia, Università di Bologna, CINECA, Dipartimento di Informatica dell' Università di Pisa, POSTECOM, Università Commerciale "Luigi Bocconi", Omnitel-Vodafone.

² La suddivisione in cinque domini è recentissima (giugno 2001) ed è originata dalla suddivisione del domain User Interface nei due domain: Document Formats e Interaction.

guidano il lavoro svolto nelle varie Activities.

Questi principi informatori 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 device 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.

XML: la pietra angolare?

Perché XML?

L'HTML era stato ottimizzato per un facile apprendimento, per cui aveva un unico tag set, con una semantica predefinita per ogni tag. Essendo le strutture dati predefinite, e mancando una validazione formale, offriva un supporto inadeguato per gestire strutture dati complesse, articolate su più livelli. Ovviamente, l'HTML era il risultato di un compromesso tra le potenzialità di utilizzo e la facilità di uso, e si è dimostrato molto utile e potente per la realizzazione di applicazioni ragionevolmente semplici. I limiti di HTML emergono quando si hanno di fronte applicazioni che presentano requisiti più stringenti, determinati dalla complessità delle strutture dati da gestire, dalla necessità di utilizzare gli stessi dati in maniera diversa e di gestire in modo adeguato dati con un ciclo di vita significativo.

Extensible Markup Language (XML) è nato per far fronte alle nuove applicazioni Web, in cui i dati costituiscono un elemento essenziale (*data-centric Web applications*).

XML 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. Ulteriori e significativi vantaggi sono costituiti dalla possibilità di avere viste diverse degli stessi dati, e la possibilità di personalizzare le informazioni mediante opportuni agenti. L'adozione di XML agevola la gestione di collezioni di documenti, e costituisce un supporto fondamentale per la pubblicazione di informazioni a livello internazionale, con il non piccolo vantaggio di essere indipendente dalla piattaforma e dal linguaggio. XML è un formato testo semplice e molto flessibile, derivato da SGML (ISO 8879). Progettato inizialmente per far fronte ai problemi dell'editoria elettronica a grande scala, XML sta ora giocando un ruolo sempre più significativo nello scambio di una grande varietà di dati sul Web.

Caratteristiche della struttura del linguaggio

Le caratteristiche di XML possono essere illustrate con un esempio semplice, relativo alla gestione degli Ordini.

In Figura 1 viene riportato un esempio di documento XML che descrive un Ordine. La sintassi XML usa tag di inizio e fine, come per esempio `<importo>` e `</importo>`, per marcare i campi informativi. Un campo informativo racchiuso tra due marcatori viene detto elemento (*element*) e può essere ulteriormente arricchito dalla presenza di coppie nome/valore (nell'esempio, `id="ord001"`) dette attributi (*attribute*). Come si può vedere, si tratta di una sintassi semplice, la cui elaborazione automatica è poco complessa, senza codifiche particolarmente criptiche, per cui resta comprensibile alla lettura diretta. XML, basato su SGML, riesce familiare a chi conosce HTML.

```
01 <?xml version="1.0"?>
11 <ordine id="ord001">
12   <cliente db="codcli123"/>
13   <prodotto db="prod345">
14     <importo>23.45</importo>
15   </prodotto>
16 </ordine>
```

Figura 1 - Un documento XML per descrivere un Ordine

```

01 <?xml version="1.0"?>
02 <!DOCTYPE ordine [
03 <!ELEMENT ordine ( cliente, prodotto+ )>
04 <!ATTLIST ordine id ID #REQUIRED>
05 <!ELEMENT cliente EMPTY>
06 <!ATTLIST cliente db CDATA #REQUIRED>
07 <!ELEMENT prodotto ( importo )>
08 <!ATTLIST prodotto db CDATA #REQUIRED>
09 <!ELEMENT importo ( #PCDATA )>
10 ]>
11 <ordine id="ord001">
12   <cliente db="codcli123"/>
13   <prodotto db="prod345">
14     <importo>23.45</importo>
15   </prodotto>
16 </ordine>

```

Figura 2 - Un documento XML con la sua DTD (in grassetto)

In XML i tag devono essere inseriti correttamente uno dentro l'altro, deve esistere una corrispondenza tra il tag di apertura e quello di chiusura, sono previsti elementi a campo informativo nullo e gli attributi dei tag devono essere racchiusi tra doppi apici.

La presenza di una struttura formale del documento, espressa nella **DTD** (*Document Type Definition*), non ha un impatto diretto sul modello strutturale implicito: nell'esempio di Figura 2, in cui la DTD è inclusa nel documento (ma potrebbe anche essere referenziata come risorsa esterna) la riga 6 specifica che l'attributo *db* è obbligatorio. 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.

Per cosa?

XML costituisce attualmente la tecnologia chiave di W3C, è infatti un mezzo espandibile e flessibile per modellare il Web, di cui costituisce la "lingua franca". Va comunque tenuto presente che in realtà W3C considera XML come una famiglia di tecnologie, su cui vengono investite le risorse del Consorzio, e non intende centralizzare il controllo di XML, preferendo lasciare agli utenti, coerentemente con la filosofia del Web, il compito di sviluppare applicazioni particolari. Il W3C vede l'XML come una Activity([XML]), che ha molte interazioni con tutte le altre. Vari gruppi di lavoro concorrono allo sviluppo della famiglia di tecnologie XML, definendo il suo minimo modello informativo (*infoset*), le sue proprietà strutturali (*schema*), la sua struttura ipertestuale (*link*), il modo di manipolare le strutture dati (*query*). La centralità di XML nell'architettura del Web è provata anche dal fatto che ogni nuovo linguaggio utilizzato per definire un nuovo standard deve essere descritto in XML.

XML è una sintassi a basso livello per rappresentare dati strutturati, e può essere utilizzato per una

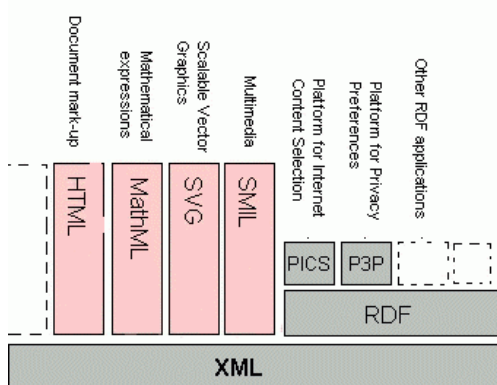


Figura 3 - XML: una sintassi a basso livello

larga varietà di applicazioni. Questa idea è rappresentata in maniera molto schematica in Figura 3, che descrive come già ora XML sia il supporto di un gran numero di applicazioni e di linguaggi di markup. Sia nel W3C che in altri contesti, molti gruppi stanno definendo nuovi formati per lo scambio delle informazioni, e questo sta portando ad una autentica esplosione di applicazioni XML, con un processo che appare inarrestabile. In molti contesti le applicazioni XML vengono utilizzate per rappresentare e organizzare i dati, con l'evidente convinzione che questo approccio possa rendere più facilmente gestibili i problemi derivanti dalla incompatibilità dei dati, riducendo il lavoro di reimmissione di informazioni che già sono in forma elettronica.

XML: storia e stato attuale di sviluppo

La *prima fase* dell' Activity XML iniziò nel giugno 1996, e si concluse nel febbraio 1998 con l' emanazione della XML 1.0 Recommendation, successivamente rivista nell' ottobre 2000 ([XML1.0]).

Nella *seconda fase*, il lavoro è stato svolto in parallelo da un certo numero di Working Group, che hanno prodotto le Recommendation *Namespaces in XML* nel gennaio 1999 e *Style Sheet linking* nel giugno 1999 ([XMLns], [XMLstylesheet]).

Nel settembre 1999 è iniziata una *terza fase*, con l' obiettivo di completare il lavoro della seconda e di introdurre un nuovo Working Group su XML Query. Al momento sono attivi diversi Working Group.

Alcuni componenti della famiglia di tecnologie XML

La famiglia di tecnologie basate su XML è estremamente vasta e in rapida evoluzione. Per semplicità e per completezza espositiva, nel seguito verranno sommariamente illustrate alcune di queste tecnologie, essenziali per poter comprendere le linee evolutive del Web. Si rimanda alla documentazione ufficiale, disponibile sul sito, per una illustrazione completa e più dettagliata.

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 URI³ 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.

```
<!DOCTYPE ordine SYSTEM "OrdDTD">
< ordine
  xmlns:cli="http://generalstore.it/anagcli"
  xmlns:ordcli="http://generalstore.it /ordcli"
  ordcli:nord="2001Set320">
  <cli:cliente>Il re dei golosi</cli:cliente>
  <rigaord>
    <prodotto>Tartufo d'Alba prima qualità</prodotto>
    <qta>1</qta>
    <umis>Kg</umis>
    - - -
  </rigaord>
</ ordine >
```

Figura 4 - Esempio di documento con riferimento a namespace

XML Schema Definition

La DTD presenta alcune limitazioni, riconducibili essenzialmente al fatto che 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. Infine, va anche ricordato che in molti casi è sufficiente che il documento sia well-formed, e quindi la DTD non è sempre necessaria.

Gli schema hanno le stesse funzionalità delle DTD, ma offrono alcuni significativi vantaggi: sono espressi con la sintassi XML e includono datatype, inheritance, regole di combinazione degli schema.

³ **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.

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. Per quanto si debba riconoscere che la XMLSchema Recommendation è piuttosto complessa, va tenuto presente che XMLSchema sostituirà nel futuro le DTD, e quindi utilizzare questa specifica costituisce un indubbio investimento per il futuro.

XLink, XPointer e XMLBase

Come è noto ([Signore1995a], [Signore1995b], [Signore1996]) i link sono un elemento fondamentale nell'architettura ipertestuale, e ne costituiscono l'autentica ricchezza.

Le proprietà avanzate del meccanismo di link sono supportate da tre specifiche:

- **XLink** per l'hyperlinking;
- **XPointer** per indirizzare porzioni di documento;
- **XMLBase** per consentire agli autori di specificare esplicitamente un URI base del documento per risolvere gli URI relativi nei link verso immagini esterne, applet, programmi per l'elaborazione di maschere, style sheet, etc.

XLink permette di creare sia semplici link unidirezionali che strutture di link più complesse, consentendo di specificare link multipli e di associare metadati ai link.

XPointer è il linguaggio da utilizzare per identificare porzioni di documento per qualunque risorsa riferita da un URI. Il linguaggio utilizzato è essenzialmente costituito da espressioni XPath⁴, con alcune estensioni.

Il Document Object Model (DOM)

Il Document Object Model (DOM) è una API (Application Programmer Interface) standard verso la struttura del documento, il cui obiettivo è semplificare l'accesso alle componenti dei documenti, per cancellare, aggiungere o modificare il loro contenuto, stile e attributi. Il DOM rende possibile lo sviluppo di applicazioni che funzionino correttamente su tutti i browser, server, e piattaforme.

Il W3C sta attualmente completando il *DOM Level 3*, che presenta alcune caratteristiche nuove. Tra i vari moduli del DOM, il DOM-XML si basa su una rappresentazione interna ad albero del documento, nella quale l'utente può navigare. Il modo standard di vedere il documento è una gerarchia di tag,

Canonical XML

La XML 1.0 Recommendation specifica la sintassi di una classe di risorse (XML Documents). I Namespace forniscono ulteriori informazioni di tipo sintattico e semantico. Tuttavia, può accadere che documenti equivalenti per alcune applicazioni abbiano una rappresentazione fisica diversa (struttura, ordine degli attributi, codifica dei caratteri). La specifica **Canonical XML** definisce un metodo per stabilire quando due documenti sono identici. Canonical XML permette quindi di definire una rappresentazione fisica del documento invariante rispetto alle differenze sintattiche del documento di input (per es. la presenza di spazi bianchi, l'ordine degli attributi, etc.).

XML Digital Signature

XML Digital Signature fornisce servizi di integrità, autenticazione di messaggi e autenticazione del firmatario per qualunque tipo di dati, sia inclusi nel documento che contiene la firma (*signature*) che altrove.

La XML signature può essere applicata al contenuto di una o più risorse, ed è un metodo per associare una chiave a dei dati, ma non specifica come le chiavi sono associate a persone o istituzioni, né il significato dei dati referenziati. Di conseguenza, per quanto costituisca un elemento importante per le applicazioni XML sicure, non è di per sé sufficiente per risolvere tutti i problemi legati alla sicurezza,

⁴ XPath è un linguaggio per indirizzare porzioni di documenti XML, ed è stato progettato per essere utilizzato da XSLT e XPointer. Adotta una sintassi compatta, non XML, per facilitarne l'uso all'interno di URI e come valore di attributi XML. XPath modella un documento XML come un albero di nodi, opera sulla struttura logica astratta di un documento XML, e deriva il suo nome dall'uso di una *path notation* come nelle URL, per navigare attraverso la struttura gerarchica dei documenti XML.

specialmente se confrontato con altri meccanismi di scambio di informazioni cifrate (signed XML, per esempio). Applicazioni sicure devono specificare ulteriori requisiti per quanto riguarda le chiavi, l'algoritmo, l'elaborazione e la restituzione.

Il processo di digital signature prevede cinque passi:

- identificare il documento da firmare;
- trasformarlo in forma "canonica";
- applicare le eventuali trasformazioni (XPath, XSLT);
- creare un sommario (*digest*) dei risultati della trasformazione;
- firmare il digest.

Metadati, RDF Model e RDFSchemata

I metadati, che rappresentano le informazioni relative ai dati, costituiscono una necessità per una varietà di esigenze, sostanzialmente riconducibili all'elaborazione semantica dell'informazione: esempi tipici sono la gestione di alcuni aspetti di privacy, sicurezza, accessibilità. Inoltre, indicizzazione e ricerca possono essere significativamente influenzati dalla gestione delle caratteristiche semantiche dell'informazione. Anche la realizzazione di agenti intelligenti presuppone la conoscenza delle caratteristiche semantiche delle informazioni. Il Resource Description Framework (RDF) non descrive la semantica, ma fornisce una base comune per poterla esprimere, permettendo di definire la semantica dei tag XML. RDF è costituito da due componenti:

- **RDF Model and Syntax:** definisce il *data model* RDF e la sua codifica XML;
- **RDF Schema:** permette di definire specifici *vocabolari* per i metadati.

Il data model RDF è molto semplice, ed è basato su tre elementi:

- *Resource* (es. pagina web, elemento HTML, articolo, libro, etc.);
- *Property:* attributo o relazione per descrivere la risorsa
- *Statement:* costituito da una tupla composta da una risorsa (*soggetto*), una proprietà (*predicato*) e un valore (*oggetto*) della proprietà per quella risorsa.

Per esempio, il fatto:

Oreste Signore, la cui e.mail è *oreste@w3.org* è l'*autore* di questo *articolo*

Potrebbe essere rappresentato dallo statement RDF espresso in XML come in Figura 5.

RDFSchemata permette definire significato, caratteristiche e relazioni di un insieme di proprietà, per esempio, permette di definire la caratteristica "creator" di Figura 5. Dato che è uno Schema Specification Language, permette di definire un insieme di predicati, i possibili soggetti e oggetti, i vincoli sull'uso dei predicati.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:appl="http://description.org/schema/"
  <rdf:Description about="http:articolo">
    <appl:creator>
      <rdf:Description>
        <appl:name>Oreste Signore</appl:name>
        <appl:email>oreste@w3.org</appl:email>
      </rdf:Description>
    </appl:creator>
  </rdf:Description>
</rdf:RDF>
```

Figura 5 - Un esempio di statement RDF espresso in XML

XML, XSL e XSLT nell'architettura di applicazioni

Al di là delle molte e spesso interessanti applicazioni specifiche di XML, un aspetto rilevante è costituito dalla possibilità di applicazioni di tipo più generale, che utilizzano XML per realizzare architetture applicative innovative. Tra queste, va segnalata la possibilità di *distribuire le funzioni tra client e server*, o realizzare un *ipertesto adattivo*. Molte di queste applicazioni si basano sulla comprensione di un principio chiave dell'XML, cioè la *separazione tra contenuto e forma*. Mediante le trasformazioni di stile è possibile presentare l'informazione in modo adeguato alle caratteristiche dell'utilizzatore. Considerato che frequentemente la medesima informazione può essere indirizzata a

utenti diversi, è spesso necessario adeguare la presentazione alle specifiche esigenze.

Va sempre tenuto presente ([Holman2000]) che ogni volta che viene modificato lo stile della presentazione dei dati, si ha in realtà un doppio processo:

- in prima battuta, va *trasformata* l'informazione, dalla forma utilizzata quando è stata creata nell'ambito dell'organizzazione, a quella indicata per la distribuzione delle informazioni;
- successivamente questa informazione, ulteriormente ristrutturata, deve essere *presentata* nella maniera più efficace per il fruitore.

Nel flusso delle informazioni dall'origine alla fase di presentazione (*rendering*), è conveniente legare l'informazione alla sua forma finale il più tardi possibile, mantenendo così la possibilità di utilizzare l'informazione anche per altri obiettivi durante l'intero processo. La pratica di modellare l'informazione sulla base di quello che sarà il suo utilizzo finale, per quanto molto diffusa, è da considerare poco accorta.

In generale, nella restituzione di un documento XML si hanno due passi distinti di trasformazione di stile:

- la *trasformazione* dell'istanza del vocabolario XML in una nuova istanza coerente con il vocabolario della semantica di restituzione
- la *formattazione* dell'istanza del vocabolario di restituzione nello user agent.

Per far fronte alle esigenze di questi due processi in modo indipendente, il Working Group del W3C responsabile dell'Extensible Stylesheet Language (XSL) ha messo a punto due Recommendation, una per la trasformazione dell'informazione, e l'altra per la restituzione (*rendering*).

La Recommendation *XSL Transformations* ([XSLT]) descrive un vocabolario riconosciuto da un XSLT processor per trasformare la struttura dell'informazione prodotta nell'ambito dell'organizzazione in un'altra struttura, adeguata per i successivi passi elaborativi.

La Proposed Recommendation *Extensible Stylesheet Language* ([XSL]) descrive un vocabolario riconosciuto dall'agente di rendering per trasformare i formati espressi in forma astratta in particolari mezzi di presentazione.

È importante tener presente che XSLT è stato progettato essenzialmente per *trasformare i vocabolari XML nel vocabolario XSL di formattazione*, per cui sono assenti alcune funzionalità che lo rendano uno strumento di uso affatto generale, anche se è utilizzabile per la maggior parte delle esigenze di trasformazione. Per questo motivo, i fogli di stile XSLT vengono spesso utilizzati in contesti diversi da quelli legati alla restituzione (*stylesheet rendering*) e sono detti *transformation script*. Un caso tipico è quello dell'interazione con sistemi preesistenti (*legacy system*), molto frequente anche nel settore dell'e-commerce, in cui un XSLT processor può trasformare una particolare istanza di transazione, basata sul suo vocabolario, nel vocabolario su cui è basato l'input previsto dall'applicazione preesistente. In altri termini, XSLT può rivelarsi utile in tutti quei casi in cui è rilevante non tanto la sintassi con cui viene rappresentato il contenuto, ma il risultato dell'analisi sintattica.

Il comportamento di uno stylesheet è guidato dalla presenza dei tag di markup, che definiscono un modello *implicito*, non dalla struttura *esplicita* del documento, descritta dalla DTD o da altro schema esterno al documento. Questo comportamento, indipendente dalla DTD, risulta molto utile nell'esaminare documenti XML *well formed*, che non fanno esplicito riferimento a un modello, anche se, ovviamente, la presenza di una DTD consente di operare sul documento con un livello di conoscenza più elevato.

È probabilmente superfluo rammentare che un singolo file può essere elaborato da più stylesheet, in quanto non c'è nulla che leghi una istanza di file ad uno o più specifici stylesheet.

Un XSLT processor prende in input uno o più stylesheet e uno o più file sorgente. L'input iniziale è costituito da un singolo stylesheet e un singolo file sorgente, gli altri stylesheet vengono incorporati prima dell'elaborazione completa del primo file sorgente. L'XML processor legge poi gli altri file sorgente sulla base del contenuto del primo file XML, quindi l'XSLT processor può accedere altri file sorgente in qualunque momento, sotto il controllo dello stylesheet.

Tutti gli input devono essere documenti XML *well formed*. L'ovvia conseguenza è che non è possibile elaborare file HTML che seguono convenzioni lessicali non XML, ma è possibile elaborare file XHTML (*Extensible Hypertext Markup Language*)⁵

In una architettura a tre livelli, è possibile inviare informazione strutturata al web browser utilizzando

⁵ File XHTML possono essere generati da file HTML utilizzando un tool disponibile gratuitamente sul sito W3C (<http://www.w3.org/People/Raggett/tidy/>), che corregge le codifiche HTML errate e segnala quelle che non riesce a correggere.

XSLT sull' *host*, sullo *user agent*, o su *entrambi*. Da un punto di vista tecnico, il server può distribuire il carico elaborativo ad agenti XML/XSLT inviando una combinazione di stylesheet e informazione sorgente da trasformare sul lato client; oppure può effettuare tutte le trasformazioni direttamente sul server, in modo da poter supportare anche user agent che supportano solo vocabolari HTML o HTML/CSS, o WML ([Lee2000]).

Indipendentemente dalle caratteristiche del browser, la scelta se trasferire documenti con markup XML o trasformarli in documenti HTML (semanticamente meno ricchi) può dipendere da considerazioni legate all' opportunità di rendere disponibili a tutti le informazioni contenute nel documento XML completo. Una scelta potrebbe essere quella di trasformare il contenuto del documento in base alle caratteristiche del potenziale fruitore. Infine, per diminuire l' utilizzo della banda di trasmissione e incrementare l' elaborazione distribuita, è anche possibile utilizzare due processi XSLT, uno per trasformare il markup delle informazioni in un generico formato di distribuzione, e un altro sul client agent, per la personalizzazione,

Va comunque tenuto presente che esistono due linee di pensiero: secondo alcuni è necessario porre un "semantic firewall" tra i dati aziendali e gli utilizzatori, che quindi potrebbero anche essere dei semplici browser HTML, destinati solo alla presentazione di informazioni spogliate di qualunque semantica. Altri invece ritengono il Web debba comunque evolvere verso un *semantic web* (di cui si parla nel seguito) in cui l' informazione trasmessa debba sempre essere semanticamente ricca, ed essere poi elaborata da user agent intelligenti e dal comportamento rispettoso delle regole.

L' accessibilità dei siti Web

Non è possibile non menzionare la Web accessibility⁶ importante per vari motivi:

- l' uso del Web si va diffondendo rapidamente in tutta la società, ed è la tecnologia che nella storia ha avuto la diffusione più rapida;
- il Web costituisce un ostacolo per varie forme di disabilità
- milioni di persone hanno difficoltà nell' accesso al Web;
- alcuni siti Web **devono** essere accessibili (molti Governi, tra cui quello Italiano, hanno emanato linee guida per garantire l' accessibilità dei siti).

La *Web accessibility* ha una valenza *sociale*, ma anche *economica* (costituisce un mercato rilevante) 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).

La *Web Accessibility Initiative (WAI)* del W3C ha operato in modo efficace per assicurare che le tecnologie Web supportino l' accessibilità, il cui supporto è già incluso in alcune specifiche ([HTML-AF], [CSS-AF], [SMIL-AF],[MathML]), e ha sviluppato alcune Guideline che giocano un ruolo critico nel rendere accessibile il Web:

- **Web Content Accessibility Guidelines 1.0** ([WCAG]) che spiega agli autori come creare contenuti Web accessibili.
- **Authoring Tool Accessibility Guidelines 1.0** ([ATAG]) che spiega agli sviluppatori come progettare authoring tool utilizzabili dai portatori di handicap;
- **User Agent Accessibility Guidelines 1.0**, ([UAAG]) che illustra il ruolo degli user agent nel soddisfare le esigenze dei disabili.

Uno sguardo al futuro del Web

Se già oggi è quindi possibile utilizzare la tecnologia *per distribuire le funzioni tra client e server*, o realizzare un *ipertesto adattivo*, la sfida del futuro, illustrata compiutamente da Tim Berners-Lee, Direttore del Wide Web Consortium, in varie sedi (XML2000, [WWW10]), è la *Semantic Web*. Nella visione di Berners-Lee, il Web ha una architettura a livelli (Figura 6), che verrà sviluppata completamente nel giro di una decina di anni.

Per chiarezza di terminologia, 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.

⁶ "The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect." -- Tim Berners-Lee, Direttore del W3C Director e inventore del World Wide Web.

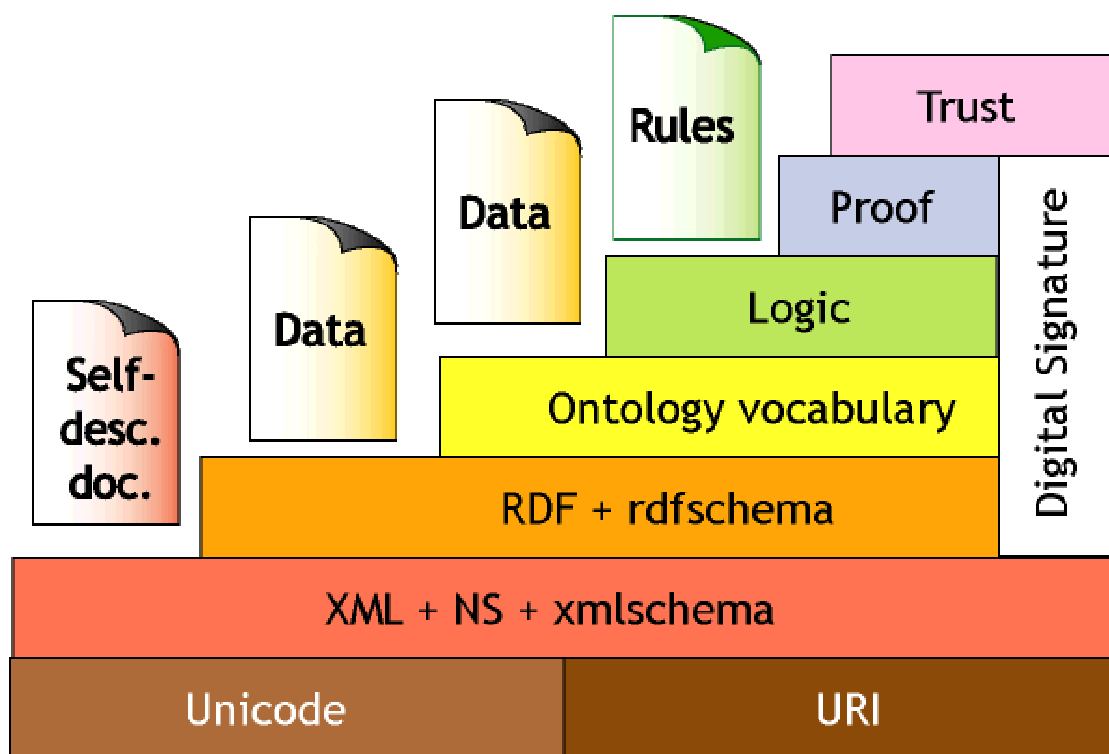


Figura 6 - L'architettura del Semantic Web (<http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>)

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. In questo senso, si potrebbe parlare del superamento di un "semantic test" quando una macchina, ricevendo dei dati, li utilizza nella maniera corretta.

Si riproduce quindi la situazione di una decina di anni fa, quando vi furono le prime dimostrazioni del Web: l'aspetto essenziale è il fattore di scala. Su piccola scala, un collegamento tra due informazioni semanticamente equivalenti, ma rappresentate in maniera diversa su due database, non rappresenta qualcosa di significativo, ma ampliare queste possibilità alla comunità mondiale, senza imporre vincoli stringenti, assume tutta un'altra rilevanza⁷.

Esaminando più in dettaglio gli elementi del Semantic Web, appare evidente, oltre al ruolo di base giocato da XML (con *Name Space* e *xmlschema*), la centralità di RDF e RDF Schema, che costituiscono il linguaggio per descrivere le risorse e i loro tipi.

Al di sopra di questo strato si pone il *livello ontologico*. 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.

La firma digitale (*digital signature*) è di significativa importanza in diversi strati nel modello astratto del Semantic Web. La crittografia a chiave pubblica è una tecnica nota da qualche anno, ma non si è ancora diffusa su larga scala come ci si poteva attendere. Nella visione di Berners-Lee, un elemento che potrebbe aver giocato contro la diffusione di questa tecnica è che essa è a "grana grossa", imponendo una scelta binaria tra fiducia o non fiducia (trusted/not trusted), mentre sarebbe necessaria una infrastruttura in cui le parti possano essere riconosciute e accettate come credibili in specifici domini. Con una granularità più fine come questa, la firma digitale potrebbe essere utilizzata per stabilire la provenienza delle ontologie e delle deduzioni, oltre che dei dati.

Il *livello logico* è il livello immediatamente superiore al livello ontologico. A questo livello le asserzioni

⁷ A puro titolo di esempio, per sottolineare l'importanza dell'interoperabilità semantica, e le difficoltà che si incontrano anche quando si fa riferimento ad un contesto specifico, ancorché ricco e complesso, si rimanda allo sforzo compiuto nel progetto *Aquarelle* (<http://aquarelle.inria.fr/aquarelle/welcome.html>), in cui, tra le altre cose, erano stati definiti delle DTD generali per la descrizione del patrimonio artistico-culturale, coerenti con le tradizioni culturali dei vari paesi partecipanti.

esistenti sul Web possono essere utilizzate per derivare nuova conoscenza. Tuttavia, i sistemi deduttivi non sono normalmente interoperabili, per cui, secondo Berners-Lee, invece di progettare un unico sistema onnicomprensivo per supportare il ragionamento (*reasoning system*), si potrebbe pensare di definire un linguaggio universale per rappresentare le dimostrazioni. I sistemi potrebbero quindi autenticare con la firma digitale queste dimostrazioni ed esportarle ad altri sistemi che le potrebbero incorporare nel Semantic Web.

Anche se i livelli più alti dell'architettura potranno richiedere diversi anni prima di poter raggiungere uno stadio in cui siano effettivamente implementati e affidabili, già oggi esiste una quantità significativa di lavoro nel settore delle ontologie. Le soluzioni pratiche includono l'uso di XSLT per derivare RDF da documenti XML, la comparsa di database e motori di ricerca RDF di tipo generalizzato, interfacce grafiche generalizzate e specifiche per RDF. La riorganizzazione delle attività sui metadati in ambito W3C dovrebbe dare un impulso al lavoro sul Semantic Web.

Non sarebbe corretto concludere questa illustrazione del Semantic Web senza segnalare che su di esso, che pure si presenta come un'idea affascinante, è più che lecito mostrare qualche perplessità o scetticismo, specialmente riandando con la memoria alle illusioni create dall'Intelligenza Artificiale negli anni '60 e '70. Inoltre, questa linea di sviluppo non sembra avere molti punti in comune con i "web services" che molti venditori XML propongono come il prossimo futuro, investendovi anche significative risorse.

XML: luci e ombre

L'e-business, supportato dalla tecnologia Internet, può a buona ragione essere considerato oggi la forza trainante nell'evoluzione del modello di business. A breve termine, questa rivoluzione avrà un impatto massiccio sulle transazioni business-to-business.

Per supportare le transazioni e-business, è necessario un linguaggio comune per scambiarsi le informazioni. Abbiamo già detto come HTML non possedesse le caratteristiche necessarie, mentre XML, con le sue potenzialità di strutturazione, permette di definire significato e struttura delle informazioni. Quasi ovvio che esso sia apparso come la panacea, il nuovo esperanto per il mondo dell'e-business, e sia stato accolto con enorme favore dai principali attori del mondo IT. Imprese concorrenti tra di loro, come IBM, Microsoft, Sun, Oracle, supportano lo standard XML e stanno effettuando investimenti su prodotti coerenti con gli standard che essi stessi, in modo cooperativo, aiutano a definire. Tuttavia, è evidente che XML costituisce solo la base per definire uno standard per le transazioni e-business, mentre mancano ancora tasselli importanti per raggiungere una reale interoperabilità semantica. La libertà di markup di XML ne costituisce infatti sia un punto di forza che di debolezza. Inoltre, la spinta verso una strutturazione estremamente elevata può portare, in alcuni ambiti, in cui il testo libero ha una valenza considerevole, a una autentica incomunicabilità delle informazioni, che fanno riferimento ad organizzazioni concettuali e vocabolari diversi e non riconciliabili.

La situazione che si è venuta a creare richiama in qualche modo alla memoria quella determinata agli inizi degli anni '80 dall'affermarsi dei DBMS relazionali. Anche in quel caso, il modello si limitava a definire come rappresentare e memorizzare le informazioni, senza imporre nessun vincolo semantico (anche perché le implementazioni commerciali gestivano un concetto molto debole di dominio). Esattamente come in XML ognuno può definire a suo piacimento strutture di documenti e nomi di tag, così nel mondo relazionale ognuno poteva definire a suo piacimento i nomi delle tabelle e delle colonne. I risultati sono evidenti a tutti: anche se tuttora accade di sentir dire che due sistemi informativi che utilizzano lo stesso DBMS sono "compatibili senza problema", tutti sanno che in realtà informazioni semanticamente equivalenti sono memorizzate e gestite in maniera difforme sia tra imprese diverse, che talvolta anche nell'ambito della stessa organizzazione.

Qualcuno stima i costi di integrazione delle applicazioni anche a un 40% dei costi IT totali. Anche se questa stima fosse eccessiva, andrebbero comunque considerati i costi indotti, determinati dalla difficoltà a sviluppare nuove applicazioni e i conseguenti ritardi nella loro realizzazione.

È quindi evidente il rischio implicito nell'adozione di XML come soluzione dei problemi di interscambio dell'informazione: esso costituisce unicamente uno standard a livello fisico, e parzialmente a livello logico, ma non offre nessuna garanzia a livello concettuale, l'unico che potrebbe garantire una reale interoperabilità.

In apparenza, il problema parrebbe risolto, almeno nell'ambiente e-business, dall'adozione di uno degli standard emergenti, ma purtroppo, al momento, ne esistono diversi.

L'ipotesi di realizzare un repository delle varie descrizioni dei messaggi XML non sembra essere davvero in grado di risolvere l'enorme problema di avere comunque un numero di procedure di conversione che è dell'ordine di N^2 (se esistono N standard di messaggi si devono implementare $N \cdot (N-1)$ traduzioni). Per riportare la complessità del problema all'ordine N , occorrerebbe implementare uno schema generale onnicomprensivo, valido per tutti i settori, che riscuota l'accordo di tutti. Questo tipo di approccio, caldeggiato in passato anche in altri contesti, non ha mai avuto successo, ed è improbabile che lo abbia in questo caso.

Conclusioni

Le linee di sviluppo del Web sono coordinate dal W3C, che le definisce con il supporto e l'accordo dei suoi membri. La tumultuosa crescita delle applicazioni Web e delle tecnologie relative impone ritmi quasi impensabili, come provato dalla quantità di documentazione tecnica che si sussegue con cadenza incredibile. In un clima di apparente confusione, esistono tuttavia alcune linee di sviluppo ben definite.

L'utilizzo di XML si va diffondendo con una rapidità e una pervasività che testimoniano come molti abbiano compreso che si tratta di una tecnologia, o meglio di una famiglia di tecnologie, che sono centrali nello sviluppo del Web.

In questo quadro si inserisce l'ambizioso progetto del Semantic Web, che, considerato da un certo punto di vista, tende a rendere reale il grande sogno di Ted Nelson ([NelsonTH], [LitMach1993]). Va però sottolineato che un reale sviluppo di queste idee si potrà avere solo se si condividerà la filosofia ispiratrice del Web, mondo libero, che rifiuta o comunque riduce al minimo i vincoli della centralizzazione, delle soluzioni proprietarie, della costituzione e prevalenza di posizioni di potere.

La realtà aziendale nell'implementazione di progetti in questo contesto non è omogenea, accanto ad imprese o gruppi che sposano appieno la filosofia del Web, e realizzano applicazioni di assoluta eccellenza, permangono alcune posizioni molto conservative, legate ad approcci tradizionali e sistemi sostanzialmente chiusi, che privilegiano la stabilità e affidabilità a breve termine, perseguendo strade che difficilmente garantiscono un recupero degli investimenti in tempi meno ravvicinati, ma non remoti, in un contesto evolutivo così rapido come il Web.

Riferimenti

- [ATAG] *Authoring Tool Accessibility Guidelines 1.0*, <http://www.w3.org/TR/WAI-AUTOOLS/>
- [CSS-AF] *CSS2 accessibility features*, <http://www.w3.org/TR/CSS-access>
- [DOM] *Document Object Model (DOM)*, <http://www.w3.org/DOM/>
- [Holman2000] Holman G. Ken: *What is XSLT? (I)* <http://www.xml.com/lpt/a/2000/08/holman/s1.html>
- [HTML-AF] WAI Resource: *HTML 4.0 Accessibility Improvements*, <http://www.w3.org/WAI/References/HTML4-access>
- [Lee2000] Lee Wei Meng: *Transforming XML into WML*, http://www.wirelessdevnet.com/channels/wap/training/xslt_wml.html
- [LitMach1993] Nelson Theodor Holm: *Literary Machines 93.1*, <http://www.sfc.keio.ac.jp/~ted/TN/PUBS/LM/LMpage.html>
- [MathML] W3C's Math Home Page, <http://www.w3.org/Math/>
- [NelsonTH] <http://www.sfc.keio.ac.jp/~ted/>
- [RDFMSS] O.Lassila, R.Swrick: *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>
- [SGML] ISO 8879:1986: *Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)*; <http://www.iso.ch/cate/d16387.html>
- [Signore1995a] Signore O.: *Issues on Hypertext Design*, DEXA'95 - Database and Expert Systems Application, Proceedings of the International Conference in London, United Kingdom 4-8 September 1995, Lecture Notes in Computer Science, N. 978, Springer Verlag, ISBN 3-540-60303-4, pp. 283-292
- [Signore1995b] Signore O.: *Modelling Links in Hypertext/Hypermedia*, in Multimedia Computing and Museums, Selected papers from the Third International Conference on Hypermedia and

- Interactivity in Museums (ICHIM'95 - MCN'95), October 9-13, San Diego, California (USA), ISBN 1-88-5626-11-8, pp. 198-216
- [Signore1996] O. Signore «*Exploiting Navigation Capabilities in Hypertext/Hypermedia* ». In Proceedings of HICSS '96. Maui Hawaii, 3-6 January 1996.
- [SMIL-AF] *SMIL accessibility features*, <http://www.w3.org/WAI/EO/SMIL-access>
- [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>
- [TUUAG] *Techniques for User Agent Accessibility Guidelines*, <http://www.w3.org/WAI/UA/WAI-USERAGENT-TECHS/>
- [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>
- [W3CML] *World Wide Web Consortium (W3C) Members* - <http://www.w3.org/Consortium/Member/List>
- [W3CPD] *World Wide Web Consortium Process Document* - 19 July 2001 <http://www.w3.org/Consortium/Process-20010719/>
- [WAI-AERT] *Techniques For Accessibility Evaluation And Repair Tools*, <http://www.w3.org/TR/AERT>
- [WAI-ER] *Evaluation and Repair Tools Working Group*, <http://www.w3.org/WAI/ER/>
- [WAI-Tablin] *WAI Table Linearizer*, <http://www.w3.org/WAI/ER/WG/tablin/>
- [WAI-Tools] *Evaluation, Repair, and Transformation Tools for Web Content Accessibility*, <http://www.w3.org/WAI/ER/existingtools.html>
- [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/>
- [Worden2000] Worden, R.: *XML E-Business Standards: Promises and Pitfalls* <http://www.xml.com/lpt/a/2000/01/ebusiness/index.html>
- [WWW10] <http://www10.org/>
- [XInclude] *XML Inclusions (XInclude) Version 1.0* - W3C Working Draft 26 October 2000 <http://www.w3.org/TR/xinclude>
- [XLink] *XML Linking Language (XLink) Version 1.0* - W3C Recommendation 27 June 2001, <http://www.w3.org/TR/2001/REC-xlink-20010627/>
- [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>
- [XMLBase] *XML Base* - W3C Proposed Recommendation 20 December 2000 <http://www.w3.org/TR/xmlbase>
- [XMLCanonical] *Canonical XML Version 1.0* - W3C Recommendation - 15 March 2001, <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- [XMLns] *Namespaces in XML* - World Wide Web Consortium 14-January-1999 <http://www.w3.org/TR/REC-xml-names/>
- [XMLqa] *XQuery 1.0 Formal Semantics* - W3C Working Draft 07 June 2001 <http://www.w3.org/TR/query-algebra/>
- [XMLqdm] *XML Query Data Model* - W3C Working Draft - 7 June 2001 <http://www.w3.org/TR/query-datamodel/>
- [XMLqrf] *XML Query Requirements* - W3C Working Draft - 15 February 2001 <http://www.w3.org/TR/xmlquery-req>
- [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/>

| | |
|-----------------|---|
| [XMLstylesheet] | Associating Style Sheets with XML documents Version 1.0 - W3C Recommendation - 29 June 1999, http://www.w3.org/TR/xml-stylesheet/ |
| [XPath] | <i>XML Path Language (XPath) Version 1.0</i> - W3C Recommendation 16 November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116 |
| [XPointer] | XML Pointer Language (XPointer) Version 1.0 - W3C Last Call Working Draft 8 January 2001, http://www.w3.org/TR/xptr |
| [Xquery] | XQuery: A Query Language for XML - W3C Working Draft - 7 June 2001 http://www.w3.org/TR/xquery/ |
| [XSL] | <i>Extensible Stylesheet Language (XSL) Version 1.0</i> - W3C Proposed Recommendation 28 August 2001, http://www.w3.org/TR/xsl/ |
| [XSLT] | <i>XSL Transformations (XSLT) Version 1.0</i> - W3C Recommendation - 16 November 1999, http://www.w3.org/TR/xslt |