

Il Web è stato progettato per essere uno spazio contenente informazioni ed uno strumento per mediare l'accesso alle risorse da parte degli utenti

Sviluppo di un complesso di tecnologie e strumenti ipertestuali ed ipermediali che hanno condotto all'attuale Web

INTERSCAMBIO DI INFORMAZIONI



I dati sul Web sono organizzati e distribuiti in formati molto lontani dall'essere interpretabili e comprensibili (sul piano semantico) in modo automatico

SERVIZI ATTUALI

Raccolta risorse nella rete

Servizi di ricerca e catalogazione sul Web

Strumenti di retrieval

Motori di ricerca

Cataloghi

Meta-cataloghi

Un motore di ricerca è composto da 4 moduli funzionali:

- un modulo che raccoglie le risorse (gatherer)
- un modulo che le cataloga organizzandole in un indice (indexer)
- un modulo che permette agli utenti di effettuare ricerche (search)
- un modulo che raccoglie le risorse individuate nella fase di ricerca (retrieval)

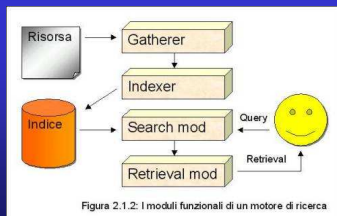


Figura 2.1.2: I moduli funzionali di un motore di ricerca

ACQUISIZIONE DEL SITO

Dal gatherer dipende la "visibilità" all'utente di un dato sito.

Gli algoritmi utilizzati per la visita dei siti sono simili a quelli di attraversamento di un albero

Depth First Breadth First

Visite di "aggiornamento" per far fronte all'estrema dinamicità dei documenti nella rete

Creazione di una mappa dei siti già attraversati con memorizzazione di eventuali link non visitati (pruning point)

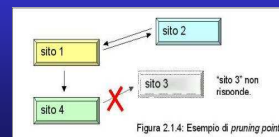


Figura 2.1.4: Esempio di pruning point.

CATALOGAZIONE ED INDICIZZAZIONE

L'operazione di catalogazione è fondamentale, in quanto da essa dipende la buona riuscita dei processi di retrieval



Gli autori di ipertesti inseriscono nel codice HTML informazioni descrittive del documento → METADATI

L'affidabilità di tale catalogazione dipende dalla bontà dell'utente

Considerare le parole ricorrenti scorrendo il testo; l'idea è che vi siano molte parole inerenti all'argomento

L'affidabilità di tale catalogazione dipende dalla bontà dell'utente e dall'ambiguità semantica della lingua

RICERCA E RACCOLTA DEI RISULTATI

Il modulo di search deve essere in grado di rispondere alle query dell'utente

RICERCHE

Associativa

Si basa sulla corrispondenza di una parola chiave keyword

Booleana

E' un potenziamento dell'associativa e permette di legare assieme più parole attraverso operatori logici

CATALOGHI

I documenti sono organizzati in categorie in base al loro argomento

Le categorie sono organizzate in una struttura ad albero ed al livello più alto si hanno poche categorie a larga estensione

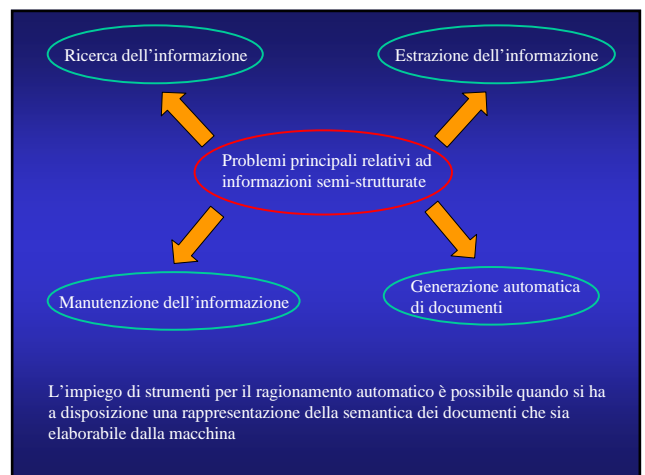
L'interrogazione procede per focalizzazione progressiva dell'argomento

META-CATALOGHI

Un meta-catalogo è la fusione di più cataloghi


Ogni catalogo ha una propria tassonomia e quindi occorre un processo di riclassificazione invece di una semplice unione

Il meta-catalogo si ottiene dalla riclassificazione di informazioni provenienti da gerarchie diverse rispetto ad una gerarchia di riferimento




I documenti oggi sul Web mediante l'impiego di HTML sono scarsamente elaborabili dalla macchina

Le pagine HTML non distinguono il contenuto dalla sua rappresentazione




Impossibilità di elaborazione semantica da parte della macchina

MOTORI DI RICERCA




Algoritmi di matching di stringhe

CATALOGHI



Catologazione manuale

Programmi di IA che prendono frasi in linguaggio naturale e tentano di ricavarne il significato




Agente intelligente che legge le pagine HTML in modo indipendente deducendo da esse le informazioni contenute


Una tecnologia allo stato dell'arte nel campo del linguaggio naturale potrebbe anche ricavare il significato di un documento ma difficilmente potrebbe dedurre informazioni significative dall'impaginazione, dalle componenti grafiche, musicali...

Come è possibile associare semantica alle pagine web?

Le risposte a questo dubbio sono articolate in 2 direzioni



Software



Strategie

VERSO IL WEB SEMANTICO

Obiettivi:

L'informazione non sarà più pensata per essere letta direttamente dall'uomo ma mirata ad essere in un formato facilmente elaborabile dalla macchina, da agenti intelligenti, da motori di ricerca potenziati semanticamente

Supporto per l'interoperabilità sintattica (facilità di leggere i dati e ottenere una rappresentazione utilizzabile da tutte le applicazioni)

Interoperabilità al livello semantico (definire mapping tra termini sconosciuti e termini conosciuti nei dati)

Il formato utilizzato per lo scambio di dati deve avere abbastanza potere per esprimere qualsiasi forma di dati

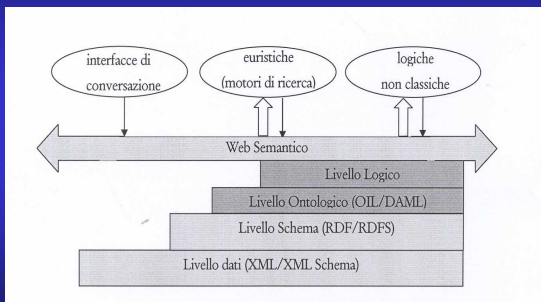
Sul piano dell'architettura logica, il web semantico non è un web separato da quello attuale, ma un layer di estensioni in cui le informazioni si vedono attribuire un significato ben definito

Il web semantico è una architettura strutturata su 4 livelli.

<p>Livello logico (supporto ragionamento) KIF, SQL, linguaggi logici</p>	<p>Non specifica l'implementazione di un motore d'inferenza ma definisce solo il linguaggio</p> <p>Definisce le primitive di modellazione delle ontologie</p>
<p>Livello ontologico (definizione di ontologie) OIL, DAML</p>	
<p>Livello schema (definizione del vocabolario) - RDF Schema</p>	
<p>Livello dei dati (semplice modello dei dati e sintassi per i metadati) - XML, RDF</p>	

Interoperabilità sintattica e semantica

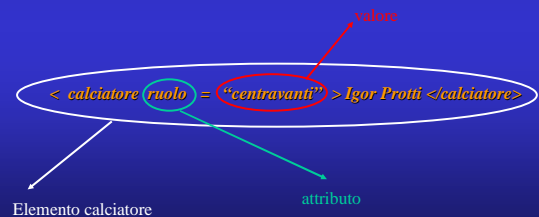
UN MODELLO



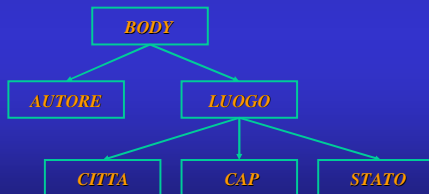
LIVELLO DATI: XML, XML SCHEMA

Un documento xml è un insieme annidato di tag aperti e chiusi che prendono il nome di **elementi** in cui ognuno di essi può avere un numero arbitrario di coppie attributo-valore

Il vocabolario degli elementi è definito per ogni applicazione specifica



```
<BODY>
Pagina scritta da
<AUTORE> Andrea Bellandi </AUTORE>
<LUOGO>
  <CITTA> Livorno </CITTA>
  <CAP> 57100 </CITTA>
  <STATO> Italia </STATO>
</LUOGO>
</BODY>
```



Nasce l'esigenza di avere costrutti dotati di nomi universali, la cui visibilità vada al di fuori del documento che li contiene



NAMESPACE

Collezione di nomi identificati da un riferimento **URI** usati nei documenti XML in associazione a elementi e nomi di attributi

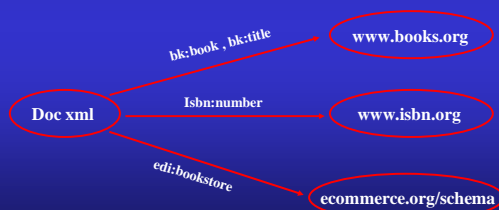
Sono stringhe che identificano risorse sul web. Ogni risorsa ha un unico indirizzo universale

I namespace devono essere dichiarati per un certo elemento utilizzando una famiglia di attributi riservati. Tali attributi hanno per valore un riferimento URI che specifica il nome del namespace e come tale deve identificarlo in modo unico e persistente

```

< bookstore xmlns:edi = 'http://ecommerce.org/schema'>
  <bk:book xmlns:bk = 'http://www.books.org/'
           xmlns:isbn = 'http://www.isbn.org/'>
    <bk:title>Intelligenza Artificiale</bk:book>
    <isbn:number>1586491379</isbn:number>
  </bk:book>
</bookstore>

```



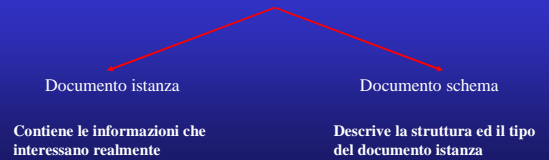
XML Schema

Per consentire sia al software che riconosce xml, sia all'uomo di sapere quale contenuto ci si aspetta in un determinato dominio, si richiede che le strutture comuni a molti documenti xml in quel dominio, siano documentate in un formato comprensibile sia alla macchina che all'uomo



Definizione di una grammatica per i documenti xml

Le specifiche di xml Schema prevedono che si faccia uso di 2 documenti



La definizione di uno schema è inglobata all'interno di un elemento schema e contiene:

1. Definizioni di tipo

2. Dichiarazioni di elementi ed attributi

1. Xml schema supporta 2 categorie distinte di tipi: *complessi* e *semplici*. nuovi tipi complessi possono essere definiti in termini di tipi semplici predefiniti.

```
<simpleType name='stringlist' base='string' derivedBy='list' />
```

Definizione di un nuovo tipo di dati, la lista di stringhe

I tipi complessi sono ottenuti a partire da un insieme di dichiarazioni di elementi e attributi. Tali dichiarazioni non sono tipi di per sé ma piuttosto associazioni tra un nome (di elemento o attributo) e un vincolo specifico che ne governa l'apparizione nei documenti XML.

```

<complexType name="indirizzo">
  <sequence>
    <element name="nome" type="string" />
    <element name="via" type="string" />
    <element name="citta" type="string" />
    <element name="cap" type="decimal" />
    <element name="stato" type="string" />
  </sequence>
</complexType>

```

Un frammento XML che faccia uso dello schema precedente è il seguente

```

<indirizzo>
  <nome> Andrea Bellandi </nome>
  <via> C. Bernheimer </via>
  <citta> Livorno </citta>
  <cap> 57100 </cap>
  <stato> Italia </stato>
</indirizzo>

```

2. Una dichiarazione di elemento associa un tipo ad un nome di elemento in un certo contesto. Per una dichiarazione locale di elementi, il contesto è il tipo complesso che la contiene (come abbiamo visto prima). Per le dichiarazioni globali, il contesto è l'elemento "radice" di un documento.

```
< schema xmlns='http://www.w3.org/1999/XMLSchema'
  targetNamespace='http://lazio.org' >
  < simpleType name='comment' base='string' >
    < maxLenght value='1024' >
  </ simpleType >
  < element name='about' type='comment' >
  < element name='autore' type='string' >
</ schema >
```

Un frammento che faccia uso del seguente schema è il seguente

```
< autore > Andrea Bellandi </ autore >
< about > Studente </ about >
```

Alcune considerazioni

XML fornisce informazioni semantiche mediante un meccanismo per la definizione della struttura di un documento.

Definisce una struttura ad albero per i documenti dove ciascun nodo individua un tag ben definito mediante il quale è possibile in qualche modo interpretare le informazioni che esso racchiude.

La semantica di un documento xml non è specificata in modo esplicito ma è incorporata nei nomi dei tag.

La semantica non è definita formalmente e può risultare eventualmente comprensibile solo all'uomo e non alla macchina.

LIVELLO SCHEMA: RDF, RDF SCHEMA

RDF fornisce uno strumento per l'aggiunta di semantica ad un documento senza fare alcuna assunzione sulla sua struttura. E' come se fosse una applicazione di XML, ovvero la sua sintassi può essere definita in tale linguaggio.

RDF è progettato per aggiungere meta informazioni ai documenti web. E' pensato per esprimere e catturare la struttura concettuale di informazioni presenti sul web.

Mentre XML supporta l'interoperabilità sintattica, RDF mira all'interoperabilità semantica.

RDF fornisce interoperabilità tra applicazioni che scambiano informazioni in formato accessibile alla macchina.

Pone l'enfasi sulla facilità di elaborazione automatica delle risorse del web.

RDF lavora con tre tipi di oggetti

RISORSA, PROPRIETA', AFFERMAZIONE

1. Una risorsa è un'entità a cui ci si può riferire mediante un URI. Le risorse vengono descritte mediante affermazioni RDF.
2. Una proprietà definisce una relazione binaria tra risorse e/o valori atomici forniti da definizioni di tipi di dati primitivi in XML.
3. Un'affermazione specifica il valore di una certa proprietà relativa ad una risorsa.

Il modello di base quindi si fonda sul concetto dei predicati e sull'associazione fra risorse e proprietà tramite una o più frasi ognuna formata da un soggetto (la risorsa), da un predicato (la proprietà) e da un oggetto (l'affermazione).

Siccome si parla di RDF e non della grammatica italiana, cosa sia un soggetto od un oggetto, non dipende dall'uso dei tempi passivi o attivi.

“Andrea è il creatore del sito www.andrea.it/”

Il soggetto è www.andrea.it/ e non Andrea come avremmo avuto in italiano. Il sito è cioè la risorsa il cui creatore (che è la proprietà) è Andrea (che è l'asserzione, ovvero il valore della proprietà creatore).

In generale si usa una struttura fissa per la frase:

SOGGETTO ha come PROPRIETÀ l'OGGETTO

La frase precedente può essere così riscritta nel modo seguente:

“il sito www.andrea.it/ ha come creatore Andrea”

Un oggetto O ha proprietà P con valore V \Rightarrow P(O,V)

Creatore (www.andrea.it/) = Andrea

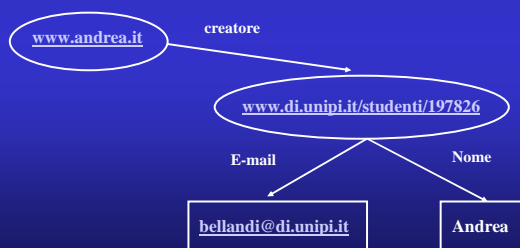
P O V

Un'asserzione RDF può essere rappresentata graficamente utilizzando grafi orientati etichettati. I nodi di forma ovale rappresentano le risorse, gli archi rappresentano le proprietà definite. I nodi che rappresentano stringhe avranno forma rettangolare.



L'oggetto di una asserzione, può essere anche un'altra risorsa invece che una stringa letterale.

“Lo studente a cui si riferisce la matricola 197826 ha come nome Andrea e come indirizzo e-mail bellandi@di.unipi.it. La risorsa è stata creata da questo individuo.”



Il modello RDF è costituito da un insieme non ordinato di affermazioni. Ogni nuova caratteristica può essere introdotta mediante una nuova proprietà RDF.

Sfortunatamente questo porta a descrivere diverse parti di una affermazione per poi poter affermare l'intera proposizione.



REPERIZIONE

I grafi quindi non solo possono essere concatenati, ma anche annidati.

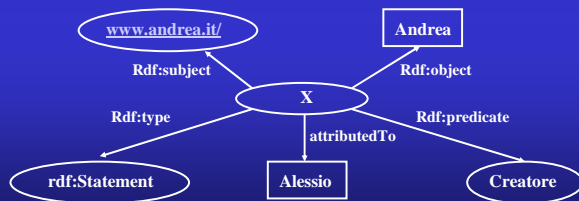
“Alessio dice che Andrea è il creatore della risorsa www.andrea.it/”

Tale frase, non dice nulla a proposito della risorsa, piuttosto esprime un fatto circa una proposizione asserita da Alessio.

La proposizione originale (Andrea è il creatore) viene espressa come una risorsa con 4 proprietà: soggetto, predicato, oggetto, tipo.

E' questo il processo di reificazione

Il grafo seguente esprime l'intera proposizione



La proprietà **attributedTo** stabilisce a chi è attribuita la proposizione identificata dalle quattro triple:

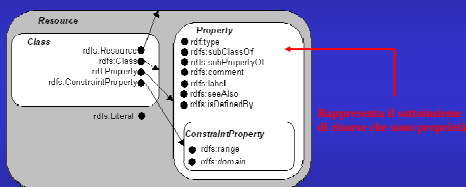
Rdf:subject (X, www.andrea.it/)
Rdf:object (X, Andrea)
Rdf:predicate (X, creatore)
Rdf:type (X, rdf:Statement)

RDF non fornisce alcun meccanismo per la dichiarazione delle proprietà che possono essere usate né per definire relazioni tra tali proprietà e altre risorse

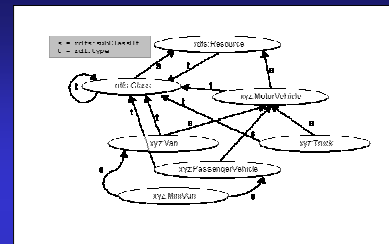
RDF Schema permette di interpretare semanticamente le risorse, le proprietà e le proposizioni definite in un documento RDF

RDF Schema

Permette la definizione di un vocabolario che può essere usato per le proprietà RDF e la specifica dei tipi di oggetti a cui queste proprietà possono essere applicate. E' un insieme di risorse RDF che possono essere utilizzate per descrivere proprietà o altre risorse



I rettangoli ovali mostrano il concetto di classe e sottoclasse (quelli inclusi in altri)
 I pallini neri rappresentano le risorse. Le frecce vanno da una risorsa alla classe che tale risorsa definisce. Per ogni risorsa all'interno di una classe esiste una proprietà "Tipo" il cui valore è tale classe. Tale proprietà specifica il tipo della risorsa.



Esso evidenzia le relazioni che intercorrono tra le classi. Ad esempio, MiniVan è una sottoclasse sia di Van sia di PassengerVehicle.

ALCUNE CONSIDERAZIONI

Uno degli obiettivi di RDF è quello di definire un meccanismo per la descrizione di risorse che non facciano alcuna assunzione su un particolare dominio di applicazione e senza definire la semantica di alcun dominio. La definizione del meccanismo deve essere neutrale rispetto al dominio e, nonostante ciò, consentire la descrizione di informazioni su qualunque dominio

Una delle maggiori differenze tra XML e RDF è nella possibilità di esprimere una affermazione in molti modi sintatticamente diversi. Ad esempio la tripla RDF:

Autore (pagina, Andrea) può essere rappresentata come:

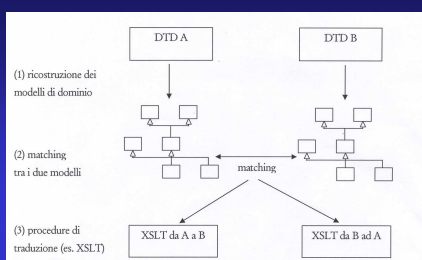
```
<autore>
  <uri> pagina </uri>
  <nome> Andrea </nome>
</autore>
oppure
<documento>
  <dettagli>
    <uri> pagina </uri>
    <autore>
      <nome> Andrea </nome>
    </autore>
  </dettagli>
</documento>
```

Qualsiasi persona comprenderebbe che tali frammenti XML significano la stessa cosa.

Al contrario un parser genera per ognuno di essi alberi XML differenti.

Conoscere la struttura non aiuta a conoscere il significato, infatti esiste un mapping da molte strutture ad un solo significato ed è necessario avere uno schema che specifichi tale mapping.

E' possibile che nella comunicazione tra due applicazioni che facciano uso del solo XML, ambedue codifichino il modello di dominio in due diversi DTD (sostanzialmente Schemi) A e B.



- (1) Ricostruzione del modello di dominio utilizzato per definire il DTD A
- (2) Stabilire mapping tra le entità nei modelli dei domini: i concetti e le relazioni devono essere "mappati" nei due modelli
- (3) Definire procedure di traduzione per documenti XML da un'applicazione all'altra

XML è comodo per lo scambio di dati tra applicazioni che conoscono entrambe il tipo di contenuto

RDF permette una certa interoperabilità semantica. Non è semplice trovare corrispondenze tra oggetti che preservino la semantica.

Un'espressione di interrogazione in termini dell'albero di XML è necessariamente più complicata di un'espressione per l'interrogazione in termini del grafo RDF.

La complessità dell'interrogazione di alberi XML è dovuta proprio al gran numero di modi in cui l'XML può rappresentare il medesimo concetto logico, considerando che l'interrogazione deve essere indipendente dalla scelta della struttura con cui si rappresenta l'informazione.

RDF fornisce quindi un modo standard per scrivere affermazioni in modo tale che, comunque esse appaiano all'interno di un documento, una interrogazione produca sempre lo stesso effetto.

Le differenze tra gli approcci alla semantica delle pagine web di XML e RDF ricordano quelle tra semantica dichiarativa e procedurale.

XML non ha nessuna semantica inerente a livello macchina e la sua semantica è determinata solo dalle azioni che uno o più programmi prendono sulla base delle espressioni che incontrano (ad esempio, annidamento di tag interpretato come relazione *parte-di* oppure *sottotipo-di* ecc....).

Un'espressione RDF/RDFS ha una semantica dichiarativa specifica (ad esempio il significato di *subClassOf*) ed essa è specificata indipendentemente da qualsiasi analizzatore di espressioni RDF, nel senso che ciascuno di essi deve adattarsi a tale semantica.

LIVELLO ONTOLOGICO: OIL + DAML

Si usa RDF Schema per descrivere le primitive di modellazione di OIL.

Una delle idee dietro OIL è che il suo potere espressivo nel modellare primitive può essere utilizzato per arricchire RDF e RDF Schema. Definire un'ontologia in RDF significa definire uno Schema RDF che a sua volta definisce termini e relazioni del particolare linguaggio. Poiché ogni ontologia fa uso del proprio namespace, termini provenienti da diverse ontologie possono essere mescolati in un documento RDF senza fare confusione.

In RDF è quindi possibile fare affermazioni con un linguaggio a proposito di un oggetto definito in termini di un altro linguaggio.

Ogni ontologia scritta con RDF Schema è utilizzabile da qualsiasi ontologia scritta in un'estensione di RDF Schema (ad esempio OIL).

Sarebbe desiderabile anche l'inverso, ovvero definire un linguaggio il più possibile vicino all'RDF Schema in modo tale da permettere il riuso delle applicazioni esistenti basate su RDF Schema. Poiché un'estensione di RDF Schema, come è OIL, di solito contiene nuovi aspetti (quindi un nuovo vocabolario che un'applicazione basata su RDF non conosce), la compatibilità totale non è possibile.

Per mantenere la massima compatibilità con le applicazioni esistenti, quindi, è bene fare uso del vocabolario di RDF Schema il più possibile, in modo da massimizzare la comprensione parziale delle stesse.

Nel lungo periodo applicazioni capaci di riconoscere l'intero potere espressivo di OIL sarebbero in grado di migliorare i servizi che forniscono a partire dalle medesime ontologie.

OIL lavora principalmente su due livelli:

Il contenitore di ontologia { Insieme di informazioni a proposito dell'ontologia come il titolo, l'autore, la descrizione

La definizione dell'ontologia stessa

I costrutti principali per la definizione dell'ontologia prevedono:

Definizione di classi

Definizione di slot, detti anche ruoli o attributi

Espressioni di classe e vincoli sugli slot

ESEMPIO DI CONTENITORE ONTOLOGIA

```
ontology-container
  title "animali vegetariani"
  creator "Pinco Pallino"
  subject "animali, vegetariani"
  description "un esempio didattico di ontologia che descrive animali vegetariani"
  description.release "1.0"
  type "ontology"
  format "pseudo-xml"
  language "OIL"
  language "it"
```

ESEMPIO DI DEFINIZIONE ONTOLOGIA

```
ontology-definitions
  slot-def mangia
  class-def animale
  class-def pianta
  class-def erba
    subclass-of pianta
  class-def albero
    subclass-of pianta
  class-def ramo
    slot-constraint parte-di
      has-value albero
  class-def foglia
    slot-constraint parte-di
      has-value ramo
  class-def erbivoro
    subclass-of animale
    slot-constraint mangia
      value-type pianta OR (slot-constraint parte-di has-value pianta)
  class-def giraffa
    subclass-of erbivoro
    slot-constraint mangia
      value-type foglia
  class-def pecora
    subclass-of erbivoro
    slot-constraint mangia
      value-type erba
```

OIL ha alle spalle una gamma di strumenti che permettono la progettazione, lo scambio, l'integrazione e la verifica di ontologie. In particolare è possibile usare il ragionatore **FACT** per controllare la consistenza di tutte le definizioni di classe all'interno di un'ontologia e scoprire relazioni (sottoclasse/superclasse) che sono implicate dalle definizioni ma non affermate esplicitamente.

OIL ha alcune limitazioni:

Mancanza di default reasoning: i valori ereditati non possono essere sovrascritti. Un tentativo di sovrascrittura porterebbe all'inconsistenza delle definizioni di classi.

Se si definisce una classe "Uccello" con attributo "volare" impostato a T e si eredita da essa la sottoclasse "Pinguino" sovrascrivendo l'attributo "volare" con il valore F, questa sottoclasse sarà inconsistente.

Regole e assiomi: solo un numero fisso di proprietà algebriche (transitività, simmetria, inversa) possono essere espresse in OIL. Non c'è nessun meccanismo per la descrizione di assiomi arbitrari che devono valere per gli elementi dell'ontologia

Mancanza di ulteriori proprietà algebriche come riflessività, irreflessività, antisimmetria, ordine parziale e totale, ecc...

Mancanza di domini concreti: OIL non supporta domini come interi, stringhe

Limitata espressività del secondo ordine: molti linguaggi esistenti per la definizione di ontologie includono qualche meccanismo di reificazione, che permette il trattamento di proposizioni del linguaggio come oggetti rendendo possibile di esprimere proposizioni di proposizioni.

DAML, è un linguaggio che considerate le somiglianze può essere considerato alla stregua di OIL.

L'idea è che agenti e motori di ricerca possano leggere i markup DAML per decifrare il significato di una pagina web.

Partendo da un insieme di statements DAML, si può concludere altri statements DAML.

Esempio

(motherOf subProperty parentOf) }
(Roberta motherOf Andrea) } → (Roberta parentOf Andrea)

DAML + OIL

Questi due linguaggi sembrano tendenti a fondersi in un unico linguaggio DAML + OIL appunto.

Questo è un linguaggio semantico di markup che si fonda sugli standard del comitato W3C, come RDF e RDF Schema, e li estende con primitive di modellazione più ricche, molte delle quali mutate dai linguaggi basati sulla logica descrittiva.

CONCLUSIONI

Il passaggio dal web attuale a quello semantico prevede vari passi.

Occorre una graduale sostituzione delle pagine HTML con documenti XML. I marcatori XML apparterranno a repertori standard per i vari domini applicativi garantendo l'uniformità sintattica delle pagine web necessaria al funzionamento degli agenti. Con l'utilizzo dei namespace ci si riferirà a schemi XML già definiti per ogni tipo di dominio.

La marcatura XML dovrà essere affiancata dai veri e propri metadati, ossia da descrizioni relative al significato e alla validità dei dati presenti nelle pagine stesse. Questo compito spetterà ad RDF. L'integrazione tra i metadati RDF e la marcatura XML dei documenti rappresenterà la chiave per la nascita di motori di ricerca intelligenti e di ambienti di interrogazione per siti, che saranno in grado di rispondere a domande tipo: "cerca tutti i libri che Tizio ha scritto dopo il 2000".

Creazione di agenti software autonomi in grado di viaggiare nella rete per creare accumulare e sintetizzare informazioni sul web, creando riepiloghi a partire dalle nostre richieste.

Grandi difficoltà si trovano nella produzione di ontologie, nella specificazione di un concetto, nel definire una terminologia ed alcune regole comuni tra membri che hanno interessi condivisi.

Questo insieme di termini e regole sul web, dovrà essere condiviso non solo dagli esseri umani, ma anche dagli agenti software e dai programmi d'interfaccia.

Esistono strumenti che mirano allo sviluppo e all'applicazione di ontologie a documenti web, importazione e riuso di ontologie, gestione di ontologie dinamiche (**OntoEdit, OILedit, Protège**)

Quando il puzzle del web semantico sarà completato, sarà possibile abilitare tale tecnologia in varie aree: motori di ricerca, e-commerce, agenti intelligenti, ecc..

BIBLIOGRAFIA

O.Lassila. Introduction to RDF Metadata, 1997.
<http://www.w3.org/TR/NOTE-rdf-simple-intro>

Deckel, J. Broekstra et al. The Semantic Web – on the respective roles of XML And RDF, 2000
<http://www.ontoknowledge.org/oil/download/IEEE00.pdf>

Tim Berners-Lee. Why RDF model is different from the XML mode, 1998
<http://www.w3.org/DesignIssues/RDF-XML.html>

Tim Berners-Lee. Web architecture from 50,000 feet, 1998.
<http://www.w3.org/DesignIssues/Architecture.html>

Tim Berners-Lee. Semantic Web roadmap, 1998.
<http://www.w3.org/DesignIssues/Smantic.html>

Tim Berners-Lee. What semantic can represent, 1998
<http://www.w3.org/DesignIssues/RDFnot.html>

D.Brickley,R.V.Guha(eds). RDF Schema specification, 2000.
<http://www.w3.org/TR/2000/CR-rdf-schema>

D.Connelly (ed.). Naming and addressing: URIs, URLs, 1998.
<http://www.w3.org/Addressing/>

E. Dumbill. The semantic web: a primer, 2000
<http://www.xml.com/pub/a/2000/11/01/semanticweb/index.html>

D.Fensel, F.vanHarmelen, I.Horrocks. OIL: A standard proposal for the semantic web, 1999.
<http://www.ontoknowledge.org/oil/papers.shtml>