

Dal Web dei Documenti al Web Semantico	1
Web Semantico – Concetto e ostacoli	2
La Nascita dei Linked Data	3
Dall’Utopia alla Pratica	3
Logica e rappresentazione della conoscenza	3
La logica proposizionale	4
La logica dei predicati	4
Arietà, ordine e decidibilità	4
Web e semantica – Logica & Segni	5
Segni e interpretazioni erronee	6
Interpretazioni nella Logica e nei predicati	6
Linguistica VS Logica	6
Logica e significati nel Web Semantico	7
Logiche descrittive	7
Assiomi	8
Complessità	9
Esempi & Limitazioni:	10
Ontologia	10
Metodo OntoClean (Guarino e Welty)	11
Specializzazione delle ontologie	12
Ragionamenti	12
Tavole di verità	13
Tableaux	13
Ragionamenti nei concetti	14
Resource Description Framework	15
RDF come modello di rappresentazione	16
Tipi di elementi nelle triple	16
RDF Come Grafi	16
RDF Schema (RDFS)	17
Sintassi di RDF	18
Dallo schema RDF a OWL	19
I profili di OWL	20
Sintassi astratta e semantica di OWL	20
Assiomi e caratteristiche formali	21
Come RDF, anche OWL dispone di diverse sintassi di serializzazione:	22
Le annotazioni	22

Punning e separazione dei livelli logici	23
Integrazione e riuso di ontologie	23
Il ciclo di sviluppo di un'ontologia	23
Funzione epistemica e comunicativa dell'ontologia	24
Architettura e mediazione concettuale	24
Modellazione concettuale e differenze con l'approccio E-R	24
Analisi del dominio e selezione delle fonti	25
Analisi linguistica e impegno ontologico	25
Categorie ontologiche e domande di competenza	25
Il ciclo di sviluppo dell'ontologia	25
Strumenti e pratiche di sviluppo	26
Le aree applicative del Web Semantico – Esterno SEO	26
Architettura generale della pubblicazione e indicizzazione	27
Annotazioni semantiche: definizione e scopi	27
RDFa	28
Microformats	28
Microdata	29
Schema.org	29
Il Knowledge Graph di Google	30
Le aree applicative del Web Semantico - Interno	30
Creazione di una base di conoscenza	31
Estrazione da testi	31
Estrazione da basi di dati	32
Costruzione della Knowledge Base RDF	32
Approccio basato sulla virtualizzazione	32
Query congiuntive	33
SPARQL	33
Integrazione semantica di servizi	33
Web Services: WSDL, SOAP e UDDI	34
Architetture orientate ai servizi (SOA)	34
L'esigenza di semantica nei servizi	35
L'affermazione delle architetture REST problemi di Semantic	36
Il problema dell'interoperabilità semantica	36
Linked open data	37
I requisiti dei dati aperti	38
I principali ambiti applicativi	39

Formati per gli open data	39
Linked data e linked open data	40
Semantica nella pubblica amministrazione	41
Il quadro europeo	42
European Interoperability Framework (EIF)	42
I core vocabularies europei	43
Il quadro italiano	43
La cooperazione applicativa nel SPC	44
SPC Coop	44
Valutazione dell'esperienza e direzioni attuali	45
Conclusioni e prospettive Linked Data e Web Semantico	45
La visione del Web Semantico e dei Linked Data	46
Problemi fondamentali: linguaggio, semantica, interpretazione	47
Quine e la relatività ontologica: il problema di Gavagai	47
Dov'è la semantica nel Web Semantico?	48
Dalla visione del Giant Global Graph alle grandi piattaforme centralizzate	48
Problemi aperti e direzioni di ricerca	48

Dal Web dei Documenti al Web Semantico

Sin dai primi anni di sviluppo del web, la comunità scientifica si interrogò sulla possibilità di accedere a un livello semantico dell'informazione, cioè a un piano che non riguardasse soltanto la struttura e i collegamenti tra documenti, ma anche il loro significato. Per "semantico" si intende ciò che concerne il significato dei dati: il modo in cui essi si riferiscono a oggetti, eventi o concetti del mondo reale.



Il **World Wide Web** nasce come effetto collaterale di una ricerca condotta dal fisico Tim Berners-Lee presso il CERN di Ginevra, uno dei principali centri di ricerca scientifica del mondo. L'obiettivo originario era facilitare la condivisione e l'integrazione dei documenti scientifici tra i ricercatori, mediante l'uso di un nuovo strumento di collegamento ipertestuale: l'ipertesto. L'idea di era quella di rendere fluido e automatico il passaggio da un testo a un altro tramite collegamenti, detti **link**. Su questa base vennero definiti gli standard fondamentali del web, che ne costituiscono tuttora l'ossatura:

1. **Gli URI (Uniform Resource Identifiers)**, che consentono di identificare in modo univoco risorse e frammenti di informazione;
2. **L'HTML (HyperText Markup Language)**, linguaggio di formattazione dei documenti ipertestuali;
3. **Il protocollo HTTP (HyperText Transfer Protocol)**, che regola la trasmissione delle informazioni secondo un modello *client-server*.

Questi tre elementi, estremamente semplici ma potenti, hanno reso possibile la creazione di un sistema decentralizzato di scambio di informazioni, nel quale ogni nodo della rete può comunicare con gli altri in modo paritetico.

Web Semantico – Concetto e ostacoli

Il **Web Semantico**, concetto introdotto sempre da Berners-Lee nel convegno fondativo del W3C nel 1994, nasce dunque con l'obiettivo di superare il web puramente documentale, consentendo non solo di collegare pagine, ma di mettere in relazione concetti e informazioni in modo formalmente descritto e interpretabile anche dalle macchine.

Uno degli ostacoli principali alla realizzazione del web semantico risiede nella natura non strutturata dei dati che circolano sul web. A differenza delle tabelle dei database relazionali, i documenti testuali non specificano in modo esplicito la natura dei dati che contengono. Ciò rende complessa l'integrazione delle informazioni, soprattutto in un contesto decentralizzato come quello del web.

Un secondo problema riguarda proprio la *decentralizzazione* stessa. Il web, concepito come una rete orizzontale e democratica di nodi autonomi, garantisce libertà e parità di accesso, ma rende difficile stabilire accordi di significato comuni. Come far sì che due sistemi diversi si riferiscano allo stesso concetto nello stesso modo? È questa la sfida semantica che il W3C, sin dagli anni Novanta, cerca di affrontare.

Tale **Web Semantico** viene sviluppata attraverso la definizione di nuovi linguaggi e formalismi, tra cui:

- **RDF (Resource Description Framework)**, uno standard per la descrizione delle risorse e delle relazioni tra esse;
- **OWL (Ontology Web Language)**, un linguaggio per la definizione di *ontologie*, cioè di rappresentazioni formali di modelli concettuali.

Un'ontologia può essere considerata come un modello concettuale analogo a quelli utilizzati nel disegno di basi di dati (ad esempio il modello *Entity-Relationship* o UML), ma espresso in forma logico-formale e condivisibile sul web.

La Nascita dei Linked Data

Nel 2006 Berners-Lee rilanciò il progetto con un nuovo impulso: la pubblicazione di una nota intitolata *Linked Data*. In essa sosteneva che, oltre alla definizione di standard formali, fosse necessario promuovere buone pratiche di pubblicazione di dati aperti sul web.

L'obiettivo era incoraggiare organizzazioni pubbliche e private a rilasciare *dataset* accessibili, interoperabili e collegabili tra loro attraverso URI condivisi, in modo da costruire una rete di dati integrata e coerente.

Il movimento dei *Linked Data* ha avuto un notevole sviluppo, sostenuto da iniziative governative (come quella britannica e, più recentemente, italiana). Oggi si contano migliaia di basi di dati interconnesse, per un totale di centinaia di miliardi di affermazioni (*statements*). Al centro di questa rete si colloca **Dbpedia**, la versione strutturata e formalizzata dei contenuti di Wikipedia, che funge da nodo di riferimento (*hub*) per molte altre basi di dati.

Dall'Utopia alla Pratica

L'idea del web semantico conserva un carattere utopico: quello di una rete globale di informazioni concettualmente integrate, capace di connettere in modo coerente le conoscenze umane. Questo ideale può essere perseguito anche in forma progressiva, pubblicando inizialmente dati “grezzi” e lasciando che i legami semanticci emergano gradualmente, attraverso un processo collettivo e decentralizzato di connessione e integrazione. L'utopia del web semantico, pur non completamente compiuta, ha generato strumenti, pratiche e risultati che hanno trasformato profondamente il nostro modo di produrre e condividere conoscenza.

Logica e rappresentazione della conoscenza

La logica nasce nell'antichità come strumento per analizzare la validità dei ragionamenti. In Aristotele essa è già concepita come una teoria della deduzione, capace di esprimere relazioni di verità tra enunciati. Tuttavia, la logica si lega presto alla filosofia del linguaggio: il linguaggio diviene il mezzo attraverso cui la realtà viene rappresentata, e le proposizioni linguistiche assumono la funzione di descrivere stati di cose nel mondo. Questo legame tra linguaggio e mondo, che attraversa la storia del pensiero da Aristotele a Frege, fino a Wittgenstein, costituisce il presupposto teorico del Semantic Web. Infatti, la sfida principale delle tecnologie semantiche è proprio quella di tradurre i significati del linguaggio naturale in strutture formali interpretabili dalle macchine, mantenendo il più possibile la fedeltà ai concetti che esse intendono rappresentare.

La logica proposizionale

La forma più semplice di formalizzazione logica è la **logica proposizionale**. Essa si fonda sull'idea che ogni proposizione sia un enunciato dotato di valore di verità, cioè possa essere vera o falsa. Le proposizioni vengono simbolicamente rappresentate da lettere maiuscole, come P, Q, R, e collegate mediante connettivi logici: la negazione (\neg), la congiunzione - e (\wedge), la disgiunzione - oppure (\vee), l'implicazione - if (\rightarrow) e l'equivalenza (\leftrightarrow).

Attraverso tali operatori è possibile costruire formule logiche complesse. Ad esempio, l'enunciato "se piove, allora prendo l'ombrelllo" può essere formalizzato come $P \rightarrow Q$, dove P rappresenta la proposizione "piove" e Q "prendo l'ombrelllo". In tal modo, la logica proposizionale permette di rappresentare le relazioni deduttive tra enunciati in modo preciso e verificabile.

Tuttavia, la sua capacità espressiva è limitata: essa non consente di rappresentare relazioni interne tra oggetti o proprietà di elementi individuali. Per superare questo limite è necessario introdurre una struttura più articolata, offerta dalla logica dei predicati.

La logica dei predicati

La **logica dei predicati**, o logica del primo ordine, amplia la logica proposizionale introducendo i concetti di variabile, costante e predicato. Le costanti rappresentano entità specifiche (ad esempio, "Socrate"), le variabili rappresentano entità generiche (x, y, z), e i predicati esprimono proprietà o relazioni tra oggetti.

Un predicato unario, come "Uomo(x)", descrive una proprietà di un singolo elemento; un predicato binario, come "Ama(x, y)", indica invece una relazione tra due individui. A queste strutture si aggiungono i quantificatori: il quantificatore universale \forall ("per ogni") e quello esistenziale \exists ("esiste"). Con essi è possibile formulare proposizioni generali, come $\forall x. \text{Uomo}(x) \rightarrow (\text{Razionale}(x) \wedge \text{Mortale}(x))$, che si legge "tutti gli uomini sono mortali e razionali". Questa forma di logica consente di esprimere conoscenze più complesse e strutturate, in cui le relazioni tra entità assumono un ruolo centrale. Proprio questa capacità di modellare le relazioni è ciò che la rende particolarmente adatta alle tecnologie del Web semantico.

Arietà, ordine e decidibilità

Ogni predicato è caratterizzato da una **arietà**, ovvero dal numero di argomenti che accetta “Ama(x, y)”. Le relazioni unarie o binarie sono le più comuni, ma nulla vieta, in teoria, l’uso di relazioni di ordine superiore. Tuttavia, nel Semantic Web, l’uso prevalente è limitato a relazioni binarie, poiché esse corrispondono alla forma tripartita della struttura **RDF** (soggetto, predicato, oggetto). Quando è necessario rappresentare una relazione tra più elementi, si ricorre a una tecnica detta **reificazione**, che consiste nel trasformare la relazione stessa in un oggetto del discorso.

Si introduce un nuovo individuo, si trasforma la relazione come se fosse un individuo singolo in modo da avere una sotto radice di individui più semplice.

Il **grado di ordine** di una logica indica invece il livello a cui i predicati possono operare. Nella logica del primo ordine essi si applicano a individui, mentre nella logica del secondo ordine possono riferirsi a insiemi o a predicati stessi. Le logiche del Web semantico, come quelle su cui si basano RDF e OWL, sono essenzialmente logiche del primo ordine, arricchite da strumenti che ne estendono il potere espressivo.

Infine, il tema della **decidibilità** riguarda la possibilità di determinare, mediante un procedimento finito, se una formula logica è vera o falsa. La logica proposizionale è completamente decidibile, mentre la logica dei predicati non lo è. Per questa ragione, gli standard del Semantic Web adottano logiche parzialmente ristrette, note come logiche descrittive, che consentono di mantenere la computabilità senza rinunciare del tutto alla capacità espressiva.

Web e semantica – Logica & Segni

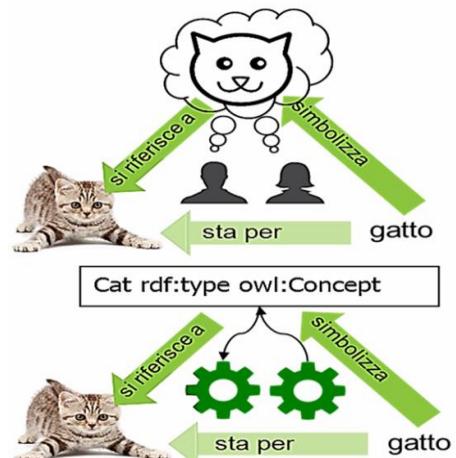
Il discorso sulla semantica, quando viene affrontato nell'ambito del web semantico, richiede di chiarire innanzitutto che cosa intendiamo per significato. L'idea generale è che il web semantico miri a costruire una rete capace di riferirsi alle cose del mondo, non solo a testi o stringhe di caratteri.

Tuttavia, questa intuizione deve essere rafforzata da un'introduzione più rigorosa, perché la semantica è un ambito in cui linguistica, logica e filosofia si intrecciano e richiedono chiarezza. È utile quindi risalire al concetto di segno, che è l'elemento fondamentale di ogni teoria del significato.

Segni e interpretazioni erronee

Molte tracce e indizi infatti sono segni senza che nessuno li abbia prodotti con volontà comunicativa. L'intenzionalità riguarda invece l'interpretazione: comprendere qualcosa come segno è sempre un atto intenzionale. Le parole sono l'esempio più evidente di segni, ma anche il tono con cui le pronunciamo può essere significativo; un ordine può essere veicolato anche dall'intonazione. Allo stesso modo, impronte sulla sabbia o il fumo che esce da un camino possono essere interpretati come segni di presenza, pur non essendo stati prodotti con intenzione comunicativa. La semiosi, cioè la produzione di significato, richiede dunque questo elemento interpretativo e mentale.

Per comprendere la struttura profonda del segno, si ricorre spesso alla rappresentazione del **triangolo semiotico**, sviluppato nella tradizione logica e ripreso da studiosi come Ogden e Richards. Esso mostra la relazione tra tre elementi: il **simbolo**, il **concetto** e il **referente**. Si prenda la parola "gatto": come simbolo scritto o pronunciato è un puro segno esteriore. Essa rimanda però a un concetto, qualcosa che esiste nella nostra mente e che organizza la nostra conoscenza dell'animale. Questo concetto, a sua volta, si riferisce a un insieme di oggetti nel mondo, cioè ai gatti reali, ma anche eventualmente a entità astratte non fisiche, come idee o forme matematiche. La relazione tra simbolo e referente non è immediata, ma mediata dal concetto, che è formato socialmente e linguisticamente; per questo motivo non possiamo assumere che due individui condividano sempre lo stesso significato quando usano una parola.



Interpretazioni nella Logica e nei predicati

Quando ci si sposta dalla linguistica alla logica formale, la questione della semantica viene trattata in modo diverso. La logica ha bisogno di una definizione rigorosa e controllata del significato, e lo fa mediante la nozione di interpretazione. Nel **calcolo proposizionale**, interpretare significa semplicemente assegnare valori di verità a proposizioni, indipendentemente da ciò che descrivono nella realtà. La logica non giudica se una proposizione sia sensata o realistica: l'assegnazione è arbitraria. Nella **logica dei predicati**, l'interpretazione diventa una funzione che associa a ogni predicato l'insieme delle combinazioni di elementi del dominio per cui esso risulta vero. Si definisce così ciò che un predicato "significa" all'interno di un determinato universo di discorso. Gli assiomi che si introducono in una teoria vincolano queste interpretazioni. Quando un'interpretazione soddisfa tutti gli assiomi, prende il nome di modello. Una teoria è soddisfacibile se ammette almeno un modello.

Linguistica VS Logica

La **linguistica**, a differenza della logica, non definisce il significato a priori, ma lo ricava dall'osservazione degli usi concreti della lingua. Questo studio si intreccia con grandi tradizioni filosofiche: il **realismo** sostiene che la realtà ponga le basi del significato; il **cognitivismo** attribuisce un ruolo decisivo alle strutture mentali; il **convenzionalismo** considera invece il significato come frutto delle pratiche sociali.

Queste prospettive non sono necessariamente alternative, ma spesso convivono, come nella posizione di Umberto Eco, secondo cui la realtà non determina il significato in modo totale, ma pone comunque dei limiti a ciò che possiamo sensatamente esprimere.

Logica e significati nel Web Semantico

Quando si arriva al web semantico, la situazione si complica perché si cerca di applicare strumenti formali della logica a un dominio, quello del linguaggio naturale, che è caratterizzato da polisemia, vaghezza e creatività. Una stessa parola può avere molti significati diversi, può avere confini sfumati nell'uso e può essere modificata o reinventata liberamente. Queste proprietà sono essenziali per la comunicazione umana, ma rappresentano un ostacolo quando si tenta di formalizzare il significato in modo rigoroso per farlo elaborare da macchine.

Realizzare il web semantico significa dunque lavorare con consapevolezza di questo divario: da una parte la logica richiede definizioni precise, dall'altra le lingue naturali si muovono entro margini elastici. La sfida consiste nel progettare modelli concettuali, ontologie e sistemi di rappresentazione del significato che rispettino i vincoli della formalizzazione logica, ma che siano al tempo stesso capaci di rappresentare, almeno in parte, la complessità e la ricchezza del linguaggio umano. Solo riconoscendo tale distanza, e non tentando di cancellarla, è possibile sviluppare strumenti realmente efficaci e coerenti per il web semantico.

Logiche descrittive

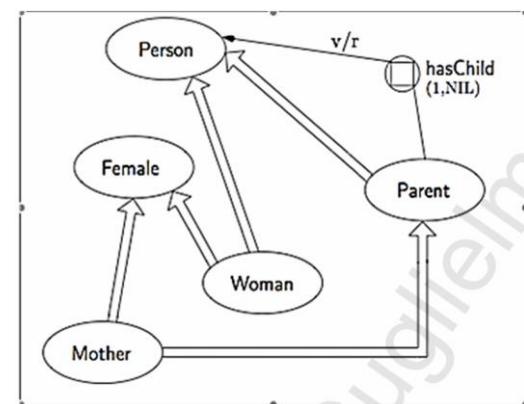
La logica descrittiva si sviluppa all'interno della logica del primo ordine, ma nasce con una finalità diversa da quella della logica matematica tradizionale. Se quest'ultima mira a formalizzare dimostrazioni e strutture deduttive astratte, la logica descrittiva è pensata per **rappresentare in modo rigoroso la conoscenza di un dominio**, in forma tale che un sistema automatico possa ragionare su ciò che è stato espresso. Questo interesse deriva dalla riflessione sul linguaggio naturale e sulle relazioni tra concetti, come quella tra un concetto più generale e uno più specifico, tra una proprietà e gli oggetti ai quali essa si applica, o tra un individuo e il ruolo che esso assume nelle relazioni.

La logica del primo ordine è teoricamente capace di rappresentare tutto ciò, ma risulta in generale **indecidibile**, cioè non garantisce sempre la terminazione dei processi di inferenza. Per questo motivo si è sentita la necessità di individuare un **frammento della logica del primo ordine** che fosse sufficientemente espressivo per modellare domini reali, ma **controllato** al punto da garantire la decidibilità delle operazioni di inferenza. La logica descrittiva si colloca precisamente in questa zona intermedia tra espressività e trattabilità computazionale.

In questo contesto, i **concetti** vengono interpretati come insiemi di individui (la base, prediciati unari), i **ruoli** invece, come relazioni binarie tra individui e gli individui come elementi specifici del dominio.

Alcuni concetti risultano **primitivi**, ossia introdotti senza definizione interna; altri sono **definiti** a partire dai primitivi tramite operatori logici (congiunzione, disgiunzione, negazione) e tramite vincoli sui ruoli, come dire che un individuo ha almeno un certo tipo di relazione verso un altro individuo appartenente a un dato concetto, oppure che tutte le relazioni di un certo tipo conducono a individui appartenenti a un concetto determinato.

- I prediciati del linguaggio (sostantivi, verbi, aggettivi, avverbi) sono collegati tra loro da una varietà di relazioni concettuali:
 - Specificità \ Generalità (cane < animale):
 - Iponimia \ superonimia.
 - Disgiunzione (animale ≠ vegetale):
 - Antonimia.
- Restrizione ($\text{mangia}(x,y) \rightarrow \text{cibo}(y)$).



- Una DL si basa su:
 - Individui (identificati da nomi distinti).
 - Concetti (predicati unari):
 - Primitivi (es. uomo).
 - Definiti (es. sposato).
 - Ruoli (Predicati binari):
 - Primitivi (es. ha-coniuge).
 - Con restrizioni sugli argomenti (dominio, codominio).
 - Con talune proprietà formali (cardinalità, transitività, simmetria, riflessività).
- Due concetti sono sempre definiti:
 - $T = \text{top}$ (l'insieme di tutti gli individui).
 - $\perp = \text{bottom}$ (l'insieme vuoto).

Principali costruzioni

- Se C e D sono concetti lo sono anche:
 - $\neg C$ (negazione) es. non Maschio
 - $C \sqcap D$ (congiunzione) es. Maschio e Sposato\o
 - $C \sqcup D$ (disgiunzione) es. Maschio o Sposato\o
- Se C è un concetto e R è un ruolo (es. Persona, coniuge) allora sono concetti:
 - $\exists R$ (quantificazione esistenziale):
 - $\{x \mid \exists y. \text{coniuge}(x,y)\}$: quelli\le che hanno almeno un\o coniuge
 - $\forall R.P$ (quantificazione universale):
 - $\{x \mid \forall y. \text{coniuge}(x,y) \rightarrow \text{Persona}(y)\}$: quelli\le coniugati solo con persone
 - $\geq n R, \leq n R$ (restrizioni di cardinalità):
 - $\{x \mid \#\{y \mid \text{coniuge}(x,y)\} \geq n \ (\leq n)\}$: quelli\le coniugati con almeno (al più) n individui

Assiomi

- Assiomi terminologici:
 - $C \sqsubseteq D$: inclusione tra concetti (tutti i C sono D , *definizioni primitive*)
 - $C \sqsubseteq \neg D$: disgiunzione tra concetti (nessun C è un D)
 - $C \equiv D$: equivalenza tra concetti (tutti i C sono D e viceversa, *definizioni di condizioni necessarie e sufficienti*)
 - $R \sqsubseteq P$: inclusione tra ruoli (tutti gli (a,b) in relazione R sono anche in relazione P)
- Asserzioni:
 - $C(a)$: a è nella estensione di C ($a \in C$)
 - $R(a,b)$: la coppia (a,b) è nella estensione di R ($\langle a,b \rangle \in R$)
- L'insieme dei concetti, dei ruoli e degli assiomi terminologici è detto TBox (Terminological Box).
- L'insieme delle asserzioni è detto ABox (Assertional Box):
 - Ciò che non è asserito in una ABox non è necessariamente falso (Open World Assumption, OWA) – notare la differenza con le basi di dati.

Per chiarezza, si può distinguere la struttura formale della conoscenza rappresentata in logica descrittiva nelle due componenti fondamentali:

Componente	Nome	Contenuto	Esempi di affermazioni
Parte terminologica	TBox	Definizione e organizzazione dei concetti e ruoli	Uomo \sqsubseteq Persona; Madre \equiv Donna \wedge \exists haFiglio.Persona
Parte assertiva	ABox	Affermazioni sugli individui concreti	anna : Donna; (anna, marco) : haFiglio

Questa distinzione consente di separare la conoscenza **strutturale** (ciò che vale sempre nel dominio) dalla conoscenza **osservativa** (ciò che riguarda casi specifici). Tale separazione risulta importante per i motori di inferenza, che possono così effettuare controlli di consistenza, classificazione dei concetti e riconoscimento delle appartenenze individuali.

Un aspetto essenziale della logica descrittiva è l'**assunzione di mondo aperto**, secondo cui ciò che non è affermato non è considerato falso, ma semplicemente non noto. Questa caratteristica la distingue profondamente dai sistemi di basi di dati tradizionali, dove invece vige l'assunzione di mondo chiuso (ciò che è falso resta falso).

Complessità

L'espressività della logica descrittiva non è uniforme, ma modulabile. Esistono famiglie di linguaggi con poteri espressivi differenti: alcuni più semplici e computazionalmente efficienti, altri più ricchi ma con un costo inferenziale maggiore. In generale, **all'aumentare dell'espressività aumenta la complessità dell'inferenza**. Questa scelta permette di modellare domini molto semplici oppure strutture concettuali complesse, a seconda delle esigenze applicative.

La semantica della logica descrittiva è definita tramite interpretazioni, dove ogni concetto è interpretato come un **sottoinsieme** del dominio, ogni ruolo come un insieme di coppie di elementi del dominio e ogni individuo come un elemento specifico del dominio stesso. Questo significa che la logica descrittiva è pienamente compatibile con la logica del primo ordine: non è una logica alternativa, ma un suo frammento con restrizioni ben progettate.

\mathcal{AL}	Concetti e gerarchie, ruoli, congiunzione e disgiunzione, quantificazione universale ed esistenziale	$A, B, A \sqcap B, A \sqcup B, A \sqsubseteq B, C \equiv A \sqcup B$ $\exists R.T$ $\forall R.D$
C	Negazione	$\neg C$
S	\mathcal{ALC} con chiusura transitiva sui ruoli	R^+
H	Gerarchie tra ruoli	$R \sqsubseteq P$
O	Enumerazioni	$C \equiv \{a_1, \dots, a_n\}$
I	Ruoli inversi	R^-
F	Cardinalità funzionale	$=1 R.T$
N	Cardinalità non qualificata	$\geq n R.T$
Q	Cardinalità qualificata	$\geq n R.D$
D	Domini concreti	STRING, INTEGER, ..

- La semantica formale delle DL è definita in termini insiemistici
 - Δ = insieme degli individui
 - I = funzione di interpretazione

NB: si assume che non vi siano definizioni cicliche, es. $C \sqsubseteq A \sqcap \forall R.C$

$$\begin{aligned}
 A^I &\subseteq \Delta^I \\
 P^I &\subseteq \Delta^I \times \Delta^I \\
 \neg C^I &= \Delta^I \setminus C^I \\
 (C \sqcap C')^I &= C^I \cap C'^I \\
 (\exists R)^I &= \{o \mid \exists o'. (o, o') \in R^I\} \\
 (\exists R.C)^I &= \{o \mid \exists o'. (o, o') \in R^I \wedge o' \in C^I\} \\
 (\forall R.C)^I &= \{o \mid \forall o'. (o, o') \in R^I \supset o' \in C^I\} \\
 (\leq n R.C)^I &= \{o \mid |\{o' \in C^I \mid (o, o') \in Q^I\}| \leq n\} \\
 (P^-)^I &= \{(o, o') \mid (o', o) \in P^I\}
 \end{aligned}$$

Esempi & Limitazioni:

TBox	ABox
$\begin{array}{lcl} \text{Woman} & \equiv & \text{Person} \sqcap \text{Female} \\ \text{Man} & \equiv & \text{Person} \sqcap \neg \text{Woman} \\ \text{Mother} & \equiv & \text{Woman} \sqcap \exists \text{hasChild}.\text{Person} \\ \text{Father} & \equiv & \text{Man} \sqcap \exists \text{hasChild}.\text{Person} \\ \text{Parent} & \equiv & \text{Father} \sqcup \text{Mother} \\ \text{Grandmother} & \equiv & \text{Mother} \sqcap \exists \text{hasChild}.\text{Parent} \\ \text{MotherWithManyChildren} & \equiv & \text{Mother} \sqcap \geq 3 \text{ hasChild} \\ \text{MotherWithoutDaughter} & \equiv & \text{Mother} \sqcap \forall \text{hasChild}. \neg \text{Woman} \\ \text{Wife} & \equiv & \text{Woman} \sqcap \exists \text{hasHusband}.\text{Man} \end{array}$	$\begin{array}{l} \text{Man(Joe)} \\ \text{Woman (Mary)} \\ \text{hasChild(Joe, Tom)} \\ \text{hasChild(Mary, Tom)} \\ \text{hasChild(Tom, Jerry)} \\ \text{hasChild(Tom, Titti)} \\ \text{hasChild(Tom, Jane)} \\ \text{-----} \\ \text{Parent(Joe)?} \\ \text{Grandmother(Mary)?} \\ \text{MotherWithManyChildren(Tom)?} \\ \text{MotherWithoutDaughter(Mary)?} \\ \text{Wife(Mary)?} \end{array}$ <p style="text-align: right;">-> NB: OWA -> NB: OWA</p>

Limitazioni

- Le DL sono logiche del primo ordine con restrizioni:
 - Limitazioni di arietà:
 - Tenere comunque conto di quanto detto sulla reificazione.
 - Separazione tra individui e predicati:
 - Non si può quantificare sui concetti (es. contare tutti i concetti x definiti sulla base del concetto C).
 - Esistono tuttavia estensioni all'ordine superiore (Higher-Order Description Logics):
 - Uso di variabili:
 - Indigeno (x) $\equiv \text{vive_in}(x,y) \sqcap \text{nato_in}(x,y)$ – l'insieme delle persone che vivono nello stesso luogo in cui sono nate non si può definire in DL.
 - Ci si deve in tal caso accontentare di un «concetto primitivo» la cui interpretazione dovrà essere garantita dalle applicazioni.

Ontologia

Nel contesto informatico, una **definizione classica di ontologia** è: “specificazione di una concettualizzazione”. Secondo questa definizione, qualsiasi teoria formalizzata in una logica descrittiva può essere considerata un’ontologia, poiché specifica una concettualizzazione. Tuttavia, questa definizione è incompleta e potenzialmente fuorviante. Ad esempio, si potrebbe definire una concettualizzazione in cui esista la nozione di “quadrato rotondo”. Tale concettualizzazione sarebbe formalmente un’ontologia, pur caratterizzando qualcosa che non può esistere nella realtà. Dunque, il livello ontologico mira ad andare oltre la mera formalizzazione delle relazioni logiche, per entrare nel merito del contenuto concettuale, verificando la plausibilità e coerenza di ciò che viene descritto.



Per questo motivo, nella ricerca sulle ontologie per il web semantico ci si concentra sulla **qualità del contenuto** concettuale. Sono state sviluppate e proposte ontologie standard di base: l’idea è di non ripartire da zero, ma usare un insieme di concetti fondamentali già costruiti e disponibili come “cassetta degli attrezzi”, da estendere e specializzare nel proprio dominio applicativo.

I metodi di elicitatione dei concetti sono il modo in cui si **estraggono** concetti e relazioni da documenti o linguaggio naturale. Questi metodi derivano in larga parte dalla tradizione della modellazione concettuale di basi di dati: si parte da sostantivi, verbi, aggettivi e da osservazioni sul lessico. Tuttavia, oltre all’estrazione terminologica, l’ontologia formale introduce criteri di buona struttura concettuale, basati su principi ontologici e non solo logici.

Etimologicamente, ontologia deriva dal greco *to on*, “ciò che è”. La riflessione sull’essere nasce già nella filosofia antica, in particolare nella linea da Parmenide ad Aristotele, contrapposta alla tradizione sofistica che sottolineava invece l’evanescenza del linguaggio. Nel Novecento, Edmund Husserl introduce la nozione di “ontologia formale”, che studia proprietà formali come identità, unità, dipendenza, considerate fondamentali e indipendenti dai contenuti specifici dei vari domini di conoscenza.

Questa tradizione è stata ripresa nell’informatica contemporanea, in particolare nella ricerca italiana guidata da Nicola Guarino presso il CNR. Questo filone ha introdotto il concetto di livello ontologico: il livello in cui ci si pone il problema di restringere lo spazio delle interpretazioni di una teoria logica, per eliminare i modelli “non intesi”, ossia interpretazioni formalmente valide ma semanticamente inaccettabili rispetto allo scopo applicativo. Ad esempio, se abbiamo i concetti “cubo”, “tavolo” e una relazione “essere su”, una teoria troppo debole potrebbe considerare lecito anche un modello in cui un cubo risulta sospeso nel vuoto sopra un tavolo senza contatto. È un modello formalmente possibile, ma non conforme all’intento semantico. Il livello ontologico si concentra quindi nel definire vincoli che escludano modelli non desiderati, affinando così la concettualizzazione condivisa.

Naturalmente non disponiamo di una teoria completa e definitiva del significato che garantisca la perfetta trasmissione dei concetti tra agenti diversi. Tuttavia, lo scopo dell’ontologia è fare “il meglio possibile” per rendere il significato quanto più stabile e condivisibile, soprattutto quando le ontologie sono pubblicate sul web e devono poter essere utilizzate da comunità non note a priori.

Metodo OntoClean (Guarino e Welty)

I criteri formali introdotti dall'ontologia formale e sistematizzati nel metodo OntoClean (Guarino e Welty) sono:

1. Il primo criterio è l'**identità**. Alcuni concetti implicano un criterio di identità per le loro istanze; questi sono detti *sortali*. Ad esempio, il concetto di “statua” è sortale: il David di Michelangelo è precisamente quella statua. Se la statua viene distrutta, non esiste più. Il “marmo”, invece, è un nome di massa: se prendo una parte del marmo, è sempre marmo, ma non è più “la statua”.
2. Il secondo criterio è l'**unità**: alcuni concetti definiscono oggetti le cui parti contribuiscono essenzialmente alla loro identità. Ad esempio, una “squadra” è costituita da esattamente quei membri; se cambio i membri, non è più la stessa squadra. Un “gruppo di persone” invece non ha questo vincolo essenziale.
3. Il terzo criterio è la **rigidità**. Un concetto rigido è tale che, se un individuo cade sotto quel concetto, lo è necessariamente in tutte le situazioni possibili. “Persona” è rigido: un individuo che è una persona lo è sempre finché esiste. “Studente” è non rigido: si può essere studenti oppure smettere di esserlo. Questo criterio spiega perché l’assioma “Ogni studente è una persona” è corretto, mentre “Ogni persona è uno studente” violerebbe la rigidità.
4. Il quarto criterio è la **dipendenza**: alcune entità dipendono necessariamente da altre per esistere formalmente. Per esempio, il colore nero di una lavagna dipende dalla lavagna stessa: distruggendo la lavagna, sparisce quella particolare istanza di colore.

Oltre ai criteri formali, le ontologie includono anche proprietà materiali fondamentali, come spazio, tempo, intenzionalità. Queste permettono distinzioni essenziali come: astratto vs. concreto, oggetto vs. evento (ciò che ha parti temporali, come l’atto della caduta della penna), azione vs. evento, dove l’azione implica agentività.

Specializzazione delle ontologie

Le ontologie sono generalmente stratificate. Al livello più alto (**top level**, o **upper level**) si trovano concetti e relazioni universali validi per qualunque dominio. A questo livello seguono le ontologie di dominio (medicina, commercio, ecc.), e infine ontologie applicative più specializzate. È comune che le ontologie di dominio siano integrate in modelli industriali o sistemi software, dove possono essere ulteriormente estese secondo le necessità.

La Basic Formal Ontology di Barry Smith, distingue tra *continuant* (oggetto) e *occurrent* (evento), e introduce sistematicamente le nozioni di dipendenza, materialità e spazialità. La struttura tassonomica si basa sul principio aristotelico di genus e differentia, già codificato nell’albero di Porfirio. Un’altra proposta è DOLCE, sviluppata anch’essa nella ricerca italiana (Borgo, Guarino e altri), che adotta un approccio simile ma con definizioni e criteri concettuali differenti.

Ragionamenti

La logica proposizionale è una logica nella quale le proposizioni rappresentano fatti che possono essere veri o falsi. Dalle proposizioni atomiche si possono costruire formule più complesse, collegate da connettivi come **negazione**, **congiunzione** (Vera se entrambi sono veri), **disgiunzione** (Vera se entrambi non sono Falsi) e **implicazione** (sempre Vera , Falsa se la premessa è Vera e la conseguenza Falsa, se il predicato a sinistra è F l'implicazione è sempre V).

a	b	$\neg a$	$a \wedge b$	$a \vee b$	$a \rightarrow b$
V	V	F	V	V	V
V	F	F	F	V	F
F	V	V	F	V	V
F	F	V	F	F	V

Tavole di verità

Un metodo elementare per stabilire la verità di una formula complessa è quello delle tavole di verità, che considerano tutte le possibili combinazioni di valori assegnabili alle proposizioni atomiche.

a	b	$\neg a$	$\neg b$	$a \rightarrow b$	$(a \rightarrow b) \wedge \neg b$	$((a \rightarrow b) \wedge \neg b) \rightarrow \neg a$
V	V	F	F	V	F	V
V	F	F	V	F	F	V
F	V	V	F	F	F	V
F	F	V	V	V	V	V

Una **formula** è considerata valida quando risulta vera per qualunque interpretazione possibile. Questo approccio si fonda sui tre principi essenziali della logica classica: **identità**, **non contraddizione** e **terzo escluso**. Tuttavia, sebbene corretto, il metodo delle tavole di verità non è pratico quando il numero di proposizioni cresce, poiché la dimensione della tabella aumenta in modo esponenziale.

- Una formula vera per qualsiasi interpretazione dei predicati è detta «valida» (tautologia).

- $a = a$ (identità)
- $\neg(a \wedge \neg a)$ (non contraddizione)
- $a \vee \neg a$ (terzo escluso)
- $\neg\neg a = a$ (doppia negazione)
- $\neg(a \wedge b) = \neg a \vee \neg b$ (forma normale negativa)
- $\neg(a \vee b) = \neg a \wedge \neg b$ (forma normale negativa)
- $a \rightarrow b = \neg a \vee b$ (eliminazione di \rightarrow)

Tableaux

Per questo motivo si utilizza il metodo dei **tableaux**. Il procedimento consiste nel negare la formula della quale si vuole verificare la validità e nel riscriverla in forma normale negativa, cioè una forma in cui la negazione si applica soltanto a proposizioni atomiche e le implicazioni sono eliminate. A questo punto si applicano

- Metodo per dimostrare che una formula è valida, mostrando che la sua negazione è insoddisfacibile:
 1. Negare la formula.
 2. Riscrivere la formula ottenuta in *forma normale negativa*.
 3. Per ogni congiunzione di formule (\wedge), incolonnare le formule componenti (regola α).
 4. Per ogni disgiunzione di formule (\vee), biforcicare le formule componenti in due colonne (regola β).
 5. Se una colonna contiene una contraddizione (es. $a \wedge \neg a$) «chiuderla», altrimenti ripetere (3)(4).
 6. Se tutte le colonne sono «chiuse», la formula negata è insoddisfacibile, quindi la formula di partenza è valida.

$((a \rightarrow b) \wedge \neg b) \rightarrow \neg a$
$\neg(((a \rightarrow b) \wedge \neg b) \rightarrow \neg a)$
$\neg\neg((a \rightarrow b) \wedge \neg b) \wedge \neg\neg\neg a$
$((a \rightarrow b) \wedge \neg b) \wedge a$
$((\neg a \vee b) \wedge \neg b) \wedge a$
$(\neg a \vee b) \wedge \neg b$
a
$\neg b$
a
/ \
$\neg a \quad b$
$\neg b \quad \neg b$
$a \quad a$
$\times \quad \times$

regole di espansione che permettono di scomporre la formula in formule più semplici: quando si tratta di una congiunzione, si introducono entrambe le componenti nello stesso ramo; quando invece si tratta di una disgiunzione, si genera una biforcazione che dà luogo a due rami distinti. Se durante questa scomposizione si arriva a una situazione in cui in un ramo compaiono una proposizione e la sua negazione, quel ramo viene chiuso perché contraddittorio. Se tutti i rami si chiudono, la negazione della formula è insoddisfacibile e, dunque, la formula di partenza è valida. Questo metodo è fondamentale perché costituisce la base teorica dei sistemi di ragionamento automatico impiegati nel web semantico e nelle ontologie.

Il **ragionamento nelle logiche descrittive** si svolge invece all'interno di una **knowledge base**, che è articolata in due componenti distinte. La T-Box contiene la parte terminologica, ossia le definizioni dei concetti e dei ruoli, e stabilisce le relazioni che valgono nel dominio in modo generale. La A-Box, invece, contiene le asserzioni sugli individui, cioè affermazioni che riguardano casi specifici. Una knowledge base è soddisfacibile quando esiste almeno un modello che renda vere tutte le sue affermazioni. Il ragionamento consiste nel determinare se certe conclusioni sono derivabili implicitamente dalla base.

- KB = <Tbox, ABox>
 - TBox = terminologia
 - ABox = asserzioni

Un **esempio** chiarisce il funzionamento: se si assume che chi insegna un corso deve essere non-studente o professore, e si afferma che Mario insegna un corso e che Mario è uno studente, ne deriva che Mario deve essere un professore, a meno che non sia specificato che essere studente ed essere professore siano concetti incompatibili. Se tale incompatibilità è dichiarata, allora la base di conoscenza risulterà contraddittoria. La KB implica che Mario sia un professore. Non avendo detto che i professori sono disgiunti dalla Tbox allora l'implicazione è compatibile.

- KB = <TBox, ABox>
- Tbox:
 - Concetti: Corso, Studente, Professore
 - Ruoli: insegnava
 - Assiomi: insegnava.CORSO $\sqsubseteq \neg$ Studente \sqcup Professore
- Abox:
 - insegnava(Mario, SWLD)
 - CORSO(SWLD)
 - Studente(Mario)
- KB \Vdash Professore (Mario)
- Cosa avviene se aggiungo alla TBox:
 - Professore $\sqsubseteq \neg$ Studente ?

Ragionamenti nei concetti

Il ragionamento nei confronti dei concetti avviene invece per **sussunzione**, cioè verificando se un concetto è contenuto in un altro. Questa verifica si riconduce a una forma di **soddisfacibilità**: si ipotizza l'esistenza di un individuo che appartenga al primo concetto ma non al secondo; se tale situazione risulta impossibile, allora il primo concetto è effettivamente incluso nel secondo. Un meccanismo analogo permette di stabilire se un individuo appartenga a un concetto definito: si verifica se l'asserzione corrispondente è derivabile dalla knowledge base.

Soddisfacibilità

- La soddisfacibilità di una KB è la possibilità per questa di avere un modello.
- Quando si introduce un nuovo concetto in una TBox soddisfacibile, si può immediatamente verificare se quel concetto preserva la soddisfacibilità.
- Questo corrisponde a verificare che la KB non fornisca una prova che $C \equiv \perp$
- In termini di soddisfacibilità:
 - $\text{KB} \Vdash C \equiv \perp$

Classificazione

- Consiste nel determinare se un'asserzione individuale è dimostrabile (vera per qualsiasi modello) in una KB:
 - $\text{KB} \Vdash C(a)$
- Esempio:
 - Studente(Mario)
 - Lavoratore(Mario) $\Vdash \text{StudenteLavoratore}(\text{Mario})$
- Si tratta di un tipo di ragionamento molto utile (es. sistemi a regole).
- In termini di soddisfacibilità:
 - $\text{KB} \cup \{\neg C(x)\}$ non è soddisfacibile

Sussunzione

Ragionamento per determinare se una descrizione è logicamente inclusa in un'altra:

- $\text{KB} \Vdash C \sqsubseteq D$

Esempio:

- Studente, Lavoratore
- StudenteLavoratore \equiv Studente \sqcap Lavoratore
- $\Vdash \text{StudenteLavoratore} \sqsubseteq \text{Studente}$
- $\Vdash \text{StudenteLavoratore} \sqsubseteq \text{Lavoratore}$

Il calcolo di sussunzione mostra le inclusioni logiche in una TBox, anche non banali o previste, aiutando a rendere la concettualizzazione intellegibile.

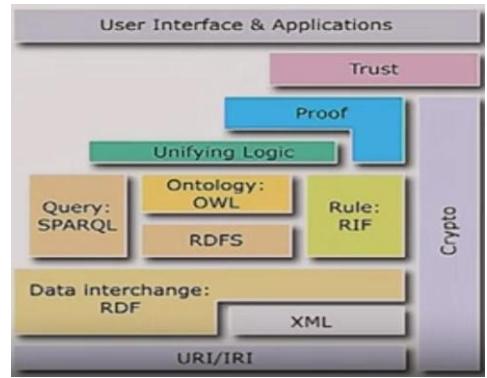
In termini di soddisfacibilità:

- $\text{KB} \Vdash C \sqsubseteq D$ sse $\text{KB} \cup \{(C \sqcap \neg D)(x)\}$ non è soddisfacibile.

Sebbene il ragionamento tramite tableaux sia corretto e completo, la sua complessità è elevata, tipicamente esponenziale rispetto alla dimensione della base di conoscenza. Per questa ragione, nella progettazione delle ontologie è fondamentale trovare un equilibrio tra la ricchezza espressiva del linguaggio utilizzato e la sostenibilità computazionale delle inferenze.

Resource Description Framework

RDF fornisce un **modello di dati** generale che consente di rappresentare informazioni in forma di **asserzioni** su risorse identificate tramite **URI/IRI**. È lo strato fondamentale su cui si costruiscono gli standard successivi come **RDF Schema**, **OWL (Web Ontology Language)** e il linguaggio di interrogazione **SPARQL**. La prima versione di RDF risale al 1999, dopo il consueto processo di discussione ed approvazione del W3C volto a definire una *recommendation*, ovvero il livello più alto di standardizzazione.



RDF come modello di rappresentazione

RDF può essere visto come una **realizzazione pratica** del concetto logico di **predicato binario**. Un predicato binario collega due entità e può essere interpretato come un **arco orientato** in un grafo che collega un **soggetto** ad un **oggetto**, etichettato con una **proprietà**.

Per questo motivo, la struttura fondamentale di RDF è la **tripla**:

Posizione	Significato	Esempio
Soggetto	la risorsa di cui si parla	:Adriano
Predicato (proprietà)	il tipo di relazione	:abitaIn
Oggetto	la risorsa o il valore collegato	:ViaGluck

Quindi: <Soggetto, predicato, oggetto>

Le triple possono concatenarsi, formando un **grafo orientato ed etichettato**, permettendo di modellare conoscenze complesse.

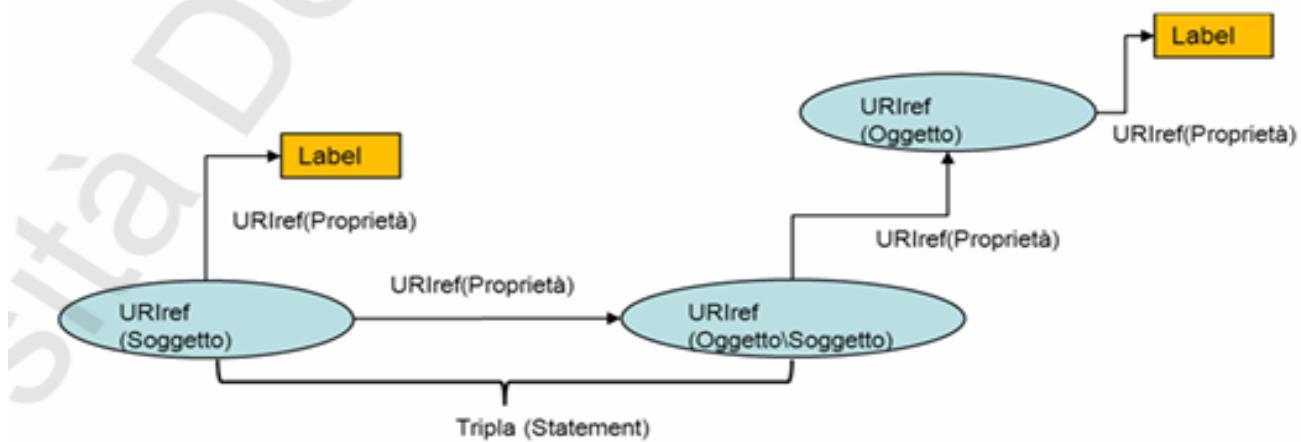
Tipi di elementi nelle triple

Elemento della tripla	Che cosa può contenere
Soggetto	URI oppure blank node (risorsa anonima)
Predicato	solo URI (mai literal, mai blank node)
Oggetto	URI, blank node oppure literal (valore tipato, es. numero, data, stringa)

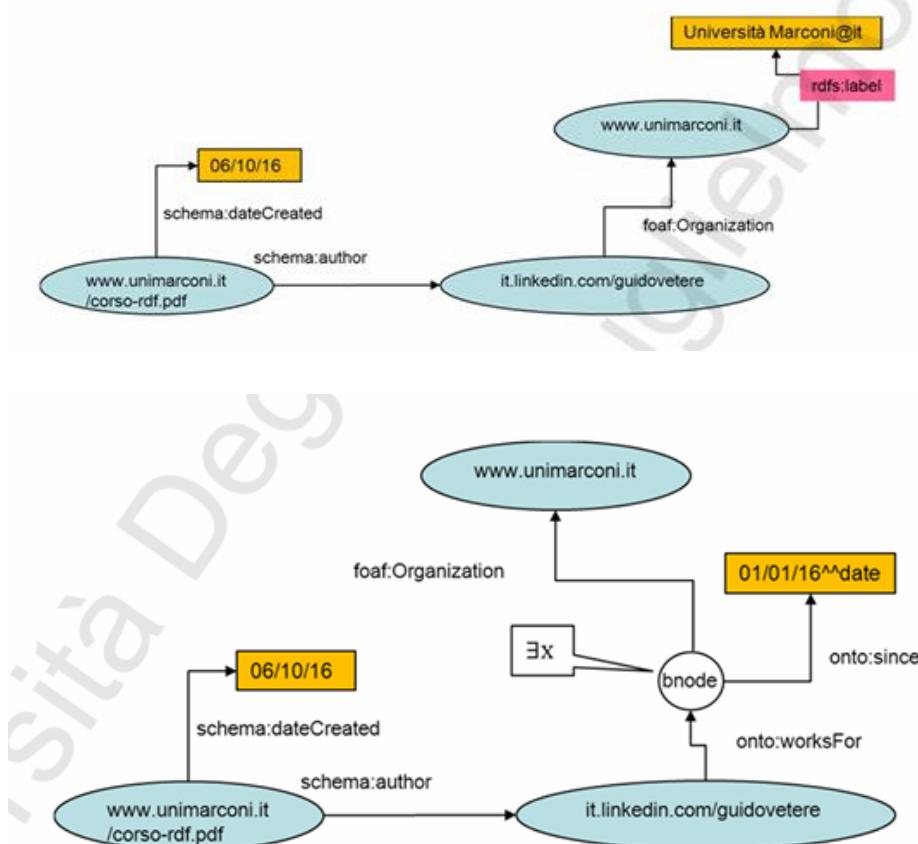
I **blank node** permettono di rappresentare informazioni strutturate senza dover assegnare un URI esplicito.

RDF Come Grafi

- Le strutture RDF possono essere visualizzate (e manipolate) come grafi diretti ed etichettati.



Esempio grafico di RDF - Descrivere una presentazione e il suo autore. Aggiunta con bnode.



RDF Schema (RDFS)

RDF da solo descrive relazioni tra risorse, ma **non definisce** la natura dei concetti coinvolti.

Per questo si usa **RDF Schema**, che permette di definire:

- **Classi** (`rdfs:Class`)
- **Gerarchie di classi** (`rdfs:subClassOf`)
- **Proprietà** (`rdf:Property`)
- **Dominio e range** delle proprietà (`rdfs:domain` e `rdfs:range`)
- **Tipo** esclusiva di RDF: (`rdf:type`) - non aggiunto nativamente da **RDF Schema**

Alcune URI, definite nello spazio dei nomi RDF, hanno un significato logico-descrittivo, servono quindi per definire relazioni formali e concettualizzazioni

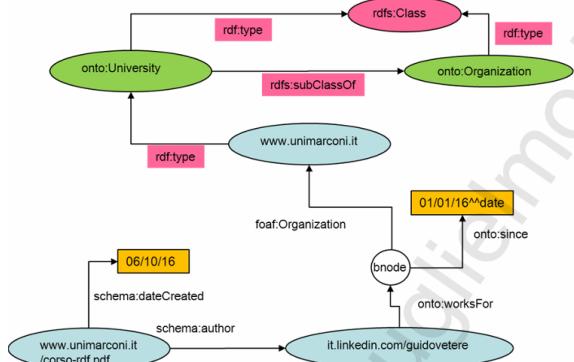
<https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>

- `rdfs:Class` a `rdfs:Class` (la classe delle classi)
- `rdfs:Resource` a `rdfs:Class` (la classe di delle risorse)
- `rdf:Property` a `rdfs:Class` (la classe delle proprietà)
- `rdfs:domain` a `rdf:Property` (restrizione di dominio)
- `rdfs:range` a `rdf:Property` (restrizione di codominio)
- `rdf:type` a `rdf:Property` (istanziazione)
 - `rdfs:range` `rdfs:Class`
 - `rdfs:domain` `rdfs:Resource`
- `rdfs:subClassOf` a `rdf:Property` (sottoclasse)
 - `rdfs:range` `rdfs:Class`
 - `rdfs:domain` `rdfs:Class`
- `rdfs:subPropertyOf` a `rdf:Property` (sottoproprietà)
 - `rdfs:range` `rdf:Property`
 - `rdfs:domain` `rdf:Property`
- `rdfs:Container` a `rdf:Class` (collezioni)
- `rdfs:member` a `rdf:Property` (appartenenza a collezione)

Ad esempio:

```
:Università a rdfs:Class
:Organizzazione a rdfs:Class
:Università rdfs:subClassOf :Organizzazione
<Unimarconi> a :Università
```

In questo modo introduciamo una **struttura concettuale (uno schema)** sopra al grafo RDF.



Sintassi di RDF

RDF non è un linguaggio unico, ma un **modello di dati**. Può essere scritto in diverse sintassi:

Sintassi	Caratteristiche
RDF/XML	storica, basata su XML, poco leggibile
Turtle	sintassi compatta e leggibile, basata sulla sequenza soggetto–predicato–oggetto, per semplificare la serializzazione dei dati RDF
JSON-LD	integrabile facilmente con applicazioni Web e API moderne

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:sc="http://schema.org/"
  xmlns:onto="http://my.org/ontology"
  xmlns:foaf="xmlns.com/foaf/spec"/>

<rdf:Description rdf:about="http://www.unimarconi.it/rdf-presentation.pdf">
  <sc:dateCreated>06/10/16</sc:dateCreated>
  <sc:author rdf:resource="http://it.linkedin.com/guidovetere"/>
</rdf:Description>

<rdf:Description rdf:about="http://it.linkedin.com/guidovetere">
  <mo:worksFor>
    <rdf:Description>
      <onto:since rdf:datatype=&xsd:date>01/01/16</mo:since>
      <foaf:Organization rdf:resource="http://www.unimarconi.it/">
    </rdf:Description>
  </mo:worksFor>
</rdf:Description>

</rdf:RDF>
```

Turtle:

```
- <URIref/Bnode> <URIref> <URIref/Bnode/Literal>;
@prefix schema: <www.schema.org>

<www.unimarconi.it/rdf-presentation.pdf>
  schema:dateCreated "06/10/16";
  schema:author <it.linkedin.com/guidovetere>

<it.linkedin.com/guidovetere>
  onto:worksFor [ foaf:Organization <www.unimarconi.it>;
    onto:since "06/10/16"^^xsd:date;
  ];

onto:University rdf:type rdfs:Class;
<www.unimarconi.it> rdf:type onto:University;
```

Dallo schema RDF a OWL

In RDF Schema è possibile definire **classi** (relazioni monadiche o predicati unari), **proprietà** (predicati binari) e le rispettive **gerarchie di sussunzione** tra concetti generali e particolari. È inoltre possibile specificare **restrizioni di dominio e codominio**, tipi di dato, e altri elementi di base.

Tuttavia, RDF Schema **non è un linguaggio logico descrittivo**. Esso non consente di definire un concetto in termini di **condizioni necessarie e sufficienti**, né di esprimere **congiunzioni**, **disgiunzioni**, o altre costruzioni logiche tipiche delle logiche descrittive. Non supporta, inoltre, **definizioni ricorsive** né **quantificazioni**, e non permette di combinare restrizioni di dominio e codominio all'interno di una descrizione concettuale.

Per colmare questa lacuna e fornire maggiore **espressività formale**, nel **2004** il **W3C** ha promulgato la raccomandazione ufficiale del linguaggio **OWL**, frutto di anni di lavoro di standardizzazione. Nel **2009** è stata pubblicata una seconda versione, **OWL 2.0**, che introduce la nozione di **profili** (profiles) per differenziare livelli di espressività e trattabilità.

I profili di OWL

La motivazione principale per la creazione dei profili è la necessità di adattare il linguaggio a **diversi casi applicativi** e garantire, ove possibile, **proprietà computazionali adeguate**.

- **OWL Full** rappresenta la versione più espressiva, ma è **indecidibile** dal punto di vista computazionale.
- **OWL DL** (Description Logic) mantiene la trattabilità tipica delle logiche descrittive, pur con limitazioni di scalabilità.
- **OWL EL** e **OWL RL** sono ottimizzati rispettivamente per grandi **vocabolari terminologici** (TBox) e per applicazioni orientate alle **regole**.
- **OWL QL** (Query Language) è orientato alla **trattabilità di grandi basi di conoscenza** (ABox), come quelle derivabili da database relazionali.

Quest'ultimo profilo, **OWL QL**, deriva da una logica descrittiva italiana, **DL-Lite**, sviluppata presso l'Università “La Sapienza” di Roma dal gruppo del professor Lenzerini. Rinunciando a parte dell'espressività, OWL QL consente di **mappare ontologie su schemi relazionali**, permettendo la **riformulazione automatica di query OWL in SQL**, senza necessità di ragionamento sui dati.

Sintassi astratta e semantica di OWL

Abstract Syntax	DL Syntax	Semantics
Descriptions (C)		
A (URI Reference)	A	$A^2 \subseteq \Delta^2$
owl:Thing	\top	$\text{owl:Thing}^2 = \Delta^2$
owl:Nothing	\perp	$\text{owl:Nothing}^2 = \emptyset$
$\text{intersectionOf}(C_1 C_2 \dots)$	$C_1 \sqcap C_2$	$C_1^2 \cap C_2^2$
$\text{unionOf}(C_1 C_2 \dots)$	$C_1 \sqcup C_2$	$C_1^2 \cup C_2^2$
$\text{complementOf}(C)$	$\neg C$	$\Delta^2 \setminus C^2$
$\text{oneOf}(o_1 \dots)$	$\{o_1, \dots\}$	$\{o_1^2, \dots\}$
$\text{restriction}(R \text{ someValuesFrom}(C))$	$\exists R.C$	$\{x \exists y (x, y) \in R^2 \cup y \in C^2\}$
$\text{restriction}(R \text{ allValuesFrom}(C))$	$\forall R.C$	$\{x \forall y (x, y) \in R^2 \rightarrow y \in C^2\}$
$\text{restriction}(R \text{ hasValue}(o))$	$R : o$	$\{x (x, o^2) \in R^2\}$
$\text{restriction}(R \text{ minCardinality}(n))$	$\geq n R$	$\{a \in \Delta^2 \{b (a, b) \in R^2\} \geq n\}$
$\text{restriction}(R \text{ maxCardinality}(n))$	$\leq n R$	$\{a \in \Delta^2 \{b (a, b) \in R^2\} \leq n\}$
$\text{restriction}(U \text{ someValuesFrom}(D))$	$\exists U.D$	$\{x \exists y (x, y) \in U^2 \cup y \in D^2\}$
$\text{restriction}(U \text{ allValuesFrom}(D))$	$\forall U.D$	$\{x \forall y (x, y) \in U^2 \rightarrow y \in D^2\}$
$\text{restriction}(U \text{ hasValue}(v))$	$U : v$	$\{x (x, v^2) \in U^2\}$
$\text{restriction}(U \text{ minCardinality}(n))$	$\geq n U$	$\{a \in \Delta^2 \{b (a, b) \in U^2\} \geq n\}$
$\text{restriction}(U \text{ maxCardinality}(n))$	$\leq n U$	$\{a \in \Delta^2 \{b (a, b) \in U^2\} \leq n\}$
Data Ranges (D)		
D (URI reference)	D	$D^2 \subseteq \Delta_D^2$
$\text{oneOf}(v_1 \dots)$	$\{v_1, \dots\}$	$\{v_1^2, \dots\}$
Object Properties (R)		
R (URI reference)	R	$\Delta^2 \times \Delta^2$
	R^-	$(R^2)^-$
Datatype Properties (U)		
U (URI reference)	U	$U^2 \subseteq \Delta^2 \times \Delta_U^2$
Individuals (o)		
o (URI reference)	o	$o^2 \in \Delta^2$
Data Values (v)		
v (RDF literal)	v	v^2

OWL si basa sullo stesso stack tecnologico di RDF: adotta le **URI** come identificativi univoci, conserva la struttura stratificata (base URI → RDF → OWL) e introduce due concetti fondamentali: **Thing** e **Nothing**, equivalenti rispettivamente al **tutto** e al **nulla** delle logiche descrittive.

OWL supporta i principali **operatori logici insiemistici**:

- **Intersezione** (congiunzione tra classi);
- **Unione** (disgiunzione di concetti);
- **Complemento** (negazione o complemento rispetto al dominio di interpretazione).

Tra i costrutti più utili figura l'**enumerazione**, che consente di definire un insieme elencandone esplicitamente gli elementi.

Il linguaggio supporta inoltre le **restrizioni sui ruoli**:

- **Esistenziali**, per descrivere individui che hanno almeno una relazione con un certo tipo di individuo;
- **Universali**, per esprimere vincoli su tutti i valori di una proprietà;
- **Di cardinalità**, per indicare il numero minimo o massimo di relazioni.

Queste restrizioni si applicano sia a **object properties** (ruoli tra individui) sia a **data properties** (ruoli che collegano individui a valori di dati, come numeri o stringhe). OWL consente anche la definizione di **data ranges**, ossia insiemi enumerati di valori concreti, per esempio tutti i nomi di battesimo italiani. Le **object properties** possono avere un **inverso** (ad esempio, “ha parte” e “è parte di”), e possono essere caratterizzate formalmente come **simmetriche**, **funzionali**, **inverse funzionali** o **transitive**.

Assiomi e caratteristiche formali

OWL permette di definire **assiomi terminologici**, quali:

- Inclusione di classi;
- Equivalenza;
- Disgiunzione.

Le caratteristiche formali delle relazioni (simmetria, transitività, funzionalità) sono trattate come **proprietà logiche standard**.

Rispetto alle logiche descrittive viste in precedenza, OWL mantiene piena corrispondenza semantica con esse.

Come RDF, anche OWL dispone di diverse sintassi di serializzazione:

- una basata su **XML** (estensione di RDF/XML);
- una più leggibile, chiamata **Manchester Syntax**, sviluppata dal gruppo dell'Università di Manchester (Ian Horrocks e colleghi), oggi standard W3C.

Sintassi di Manchester

```
Prefix: : <http://ex.com/owl/families#>
Prefix: q: <http://ex.com/owl2/families#>

Ontology: <http://example.com/owl/families> <http://example.com/owl/families-v1>
Import: <http://ex.com/owl2/families.owl>
Annotations: creator John,
Annotations: rdfs:comment "Creation Year"
creationYear 2008,
mainClass Person

ObjectProperty: hasWife
Annotations: ...
Characteristics: Functional, InverseFunctional, Reflexive, Irreflexive, Asymmetric
Domain: Annotations: rdfs:comment "General domain",
creator John
Person,
Annotations: rdfs:comment "More specific domain"
Man
Range: Person, Woman
SubPropertyOf: hasSpouse, loves
EquivalentTo: isMarriedTo, ...
DisjointWith: hates, ...
InverseOf: hasSpouse, inverse hasSpouse
SubPropertyChain: Annotations: ... hasChild o hasParent o...

DataProperty: hasAge
Annotations: ...
Characteristics: Functional
Domain: Person, ...
Range: integer, ...
SubPropertyOf: hasVerifiedAge, ...
EquivalentTo: hasAgeInYears, ...
DisjointWith: hasSSN, ...

AnnotationProperty: creator
Annotations: ...
Domain: Person, ...
Range: integer, ...
SubPropertyOf: initialCreator, ...

Datatype: NegInt
Annotations: ...
EquivalentTo: integer[< 0]
```

```
Class: Person
Annotations: ...
SubClassOf: owl:Thing that hasFirstName exactly 1 and hasFirstName only string[minLength 1], ...
SubClassOf: hasAge exactly 1 and hasAge only not NegInt, ...
SubClassOf: hasGender exactly 1 and hasGender only {female , male} , ...
SubClassOf: hasSSN max 1, hasSSN min 1
SubClassOf: not hates Self, ...
EquivalentTo: g:People , ...
DisjointWith: g:Rock , g:Mineral , ...
DisjointUnionOf: Annotations: ... Child, Adult
HasKey: Annotations: ... hasSSN

Individual: John
Annotations: ...
Types: Person , hasFirstName value "John" or hasFirstName value "Jack"^^xsd:string
Facts: hasWife Mary, not hasChild Susan, hasAge 33, hasChild :child1
SameAs: Jack, ...
DifferentFrom: Susan, ...
```

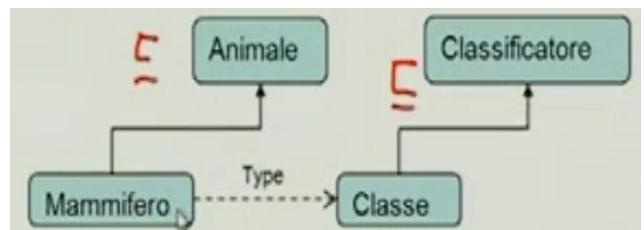
Sono inoltre disponibili versioni in **Turtle** e binding per **JSON** (utile per le API moderne).

Le annotazioni

OWL 2 introduce un potente meccanismo di **annotazioni**, che consente di allegare **informazioni descrittive o meta-informazioni** a qualsiasi elemento del linguaggio: classi, proprietà, individui e perfino assiomi. Ad esempio, si può annotare un assioma come “tutti i cani sono mammiferi” indicando la **fonte** o **l'autore** dell'informazione. È possibile usare tipi di annotazione standard o definirne di propri, includendo anche **entità della stessa base di conoscenza**. Le annotazioni possono quindi svolgere una funzione **meta-livello**, ad esempio per descrivere concetti come la **rigidità**: un concetto è rigido (es. *persona*) se vincola necessariamente le proprie istanze, mentre è non rigido (es. *professore*) se può cessare di valere nel tempo. È importante notare che OWL 2 **non assegna semantica formale** alle annotazioni: eventuali ragionamenti su di esse sono demandati all'applicazione che utilizza l'ontologia.

Punning e separazione dei livelli logici

Le logiche descrittive impongono una netta separazione tra **predicati** e **individui** per garantire la decidibilità. Tuttavia, nella pratica ontologica è frequente la necessità di rappresentare **meta-classi**, cioè classi che parlano di altre classi. OWL 2 DL



rilassa parzialmente questa restrizione, consentendo il cosiddetto **punning**: l'uso dello stesso nome per una classe e per un individuo. Ad esempio, si può definire *Mammifero* sia come classe di animali sia come individuo appartenente alla classe *Classificatore Linneano*. Così come le annotazioni questo costrutto è ammesso solo **sintatticamente**: OWL 2 non ne definisce la semantica formale, lasciando all'utente o all'applicazione il compito di interpretarla.

Integrazione e riuso di ontologie

OWL consente di **importare ontologie esterne** attraverso la direttiva `import`. Ciò permette di **riusare concetti, estendere proprietà, o dichiarare equivalenze e disgiunzioni** rispetto a concetti definiti altrove. L'operazione è analoga all'importazione di moduli in un linguaggio di programmazione. Tuttavia, è necessario **valutare con attenzione** ciò che si importa:

- **Le convenzioni di denominazione** (uso delle label e dei namespace);
- **L'orientamento concettuale o filosofico** (ad esempio la distinzione tra concetti dipendenti e indipendenti, qualità ed entità);
- **La granularità** dei concetti;
- **La qualità** dell'ontologia di origine.

Non tutte le ontologie pubblicate sul web sono ben progettate; alcune possono contenere **errori concettuali o anti-pattern**. Importare un'ontologia senza verificarla può compromettere la coerenza del proprio modello. OWL si basa sulla **logica classica**, nella quale ogni proposizione è vera o falsa. Non prevede logiche fuzzy, soggettive o basate su stati cognitivi di agenti. Questo conferisce grande **forza formale**, ma impone cautela quando si integrano conoscenze provenienti da fonti eterogenee o non affidabili.

Il ciclo di sviluppo di un'ontologia

A prima vista, un'ontologia può sembrare analoga a un modello concettuale di base di dati, come il modello Entity-Relationship. In realtà, si tratta di strumenti profondamente diversi. Il modello E-R nasce per essere tradotto in uno schema logico e fisico, legato strettamente alla struttura di un database. L'ontologia, invece, non è vincolata a un livello di implementazione: rappresenta una concettualizzazione indipendente, che può poggiare su architetture di dati diverse, dai sistemi relazionali ai database a grafo, senza perdere la propria validità.

Un altro aspetto distintivo è che l'ontologia non scompare una volta terminata la fase di progettazione, ma rimane attiva durante l'esecuzione del sistema. È infatti parte integrante del processo di inferenza: attraverso i vincoli e le relazioni che essa definisce, consente di interpretare le query e di generare risposte semanticamente coerenti. In questo senso, essa costituisce una componente operativa, non solo descrittiva.

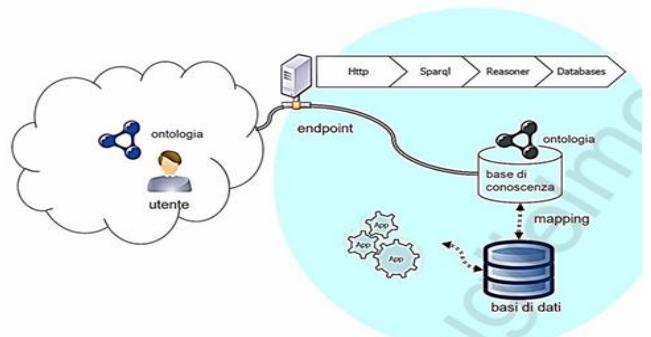
Funzione epistemica e comunicativa dell'ontologia

L'ontologia ha una duplice funzione. Da un lato, rappresenta la visione globale dei dati, fornendo una cornice concettuale all'interno della quale è possibile integrare sorgenti eterogenee. Dall'altro, essa svolge un ruolo comunicativo: costituisce la concettualizzazione che il sistema espone verso l'esterno, il linguaggio con cui dialoga con altri sistemi e con gli utenti.

Mentre in un contesto aziendale chiuso è possibile assumere conoscenze condivise e controllare rigidamente i significati, nel web semantico l'interazione avviene in un ambiente aperto, dove gli interlocutori non sono prevedibili. L'ontologia diventa così una sorta di "interfaccia concettuale" che rende esplicito il modo in cui il dominio viene inteso e rappresentato.

Architettura e mediazione concettuale

In un sistema basato su tecnologie semantiche, le basi di dati, le applicazioni e la base di conoscenza formano una gerarchia. Al vertice si colloca l'ontologia, che costituisce il punto di mediazione tra i dati e gli utenti. Quando un utente interroga il sistema attraverso un endpoint SPARQL, la richiesta attraversa diversi livelli – dal protocollo HTTP fino al motore di inferenza – e viene interpretata secondo i vincoli logici definiti nell'ontologia.



Questa architettura segna una discontinuità rispetto ai sistemi tradizionali, nei quali il livello concettuale è nascosto e funge solo da strumento di progettazione. Nel Web Semantico, al contrario, la concettualizzazione diventa parte integrante dell'interazione con l'utente.

Modellazione concettuale e differenze con l'approccio E-R

La modellazione ontologica si distingue da quella concettuale classica per finalità e metodo. Non si tratta di costruire un modello orientato a un'applicazione specifica, ma di definire una rappresentazione condivisibile del dominio, capace di essere riutilizzata in contesti diversi.

In passato, si è tentato di stabilire corrispondenze dirette tra le categorie linguistiche e i costrutti dei modelli E-R, ipotizzando che ogni nome o verbo potesse tradursi automaticamente in una classe o in una relazione. Ma la costruzione di un'ontologia richiede un ragionamento più profondo. È necessario interrogarsi sulla natura dei concetti, ponendo domande di tipo aristotelico: a quale genere appartiene un concetto e cosa lo distingue dagli altri dello stesso genere. Solo attraverso questa riflessione si può costruire una tassonomia coerente.

Analisi del dominio e selezione delle fonti

La prima fase operativa di un progetto ontologico consiste nello studio del dominio e nell'identificazione delle fonti di conoscenza. Possono essere utilizzati documenti tecnici, manuali, regolamenti o interviste a esperti, ma anche fonti aperte come encyclopédie online o repertori di vocaboli già esistenti. È tuttavia importante ricordare che modelli e schemi preesistenti, nati per finalità applicative, non sempre sono concettualmente adeguati e richiedono una revisione critica.

Esistono strumenti in grado di estrarre automaticamente termini e relazioni dai testi, ma il loro uso deve essere accompagnato da un esame umano, perché il linguaggio naturale è ambiguo e i significati variano a seconda del contesto.

Analisi linguistica e impegno ontologico

Ogni termine porta con sé un certo “impegno ontologico”, cioè un insieme di assunzioni implicite su ciò che esiste nel mondo. Quando si analizzano i testi per individuare i concetti rilevanti, occorre chiedersi quale tipo di entità viene presupposta da ciascuna parola.

Un esempio classico è il termine “libro”, che può riferirsi a un oggetto fisico, a un contenuto informativo o a un processo di produzione. Dal punto di vista ontologico, questi significati non sono equivalenti, e l'ontologia deve esplicitarne le differenze e le relazioni. La costruzione di un'ontologia implica dunque un'operazione di disambiguazione e di chiarificazione concettuale.

Metafora: uso di analogie, es. sostanza per proprietà (*essere di ghiaccio*).

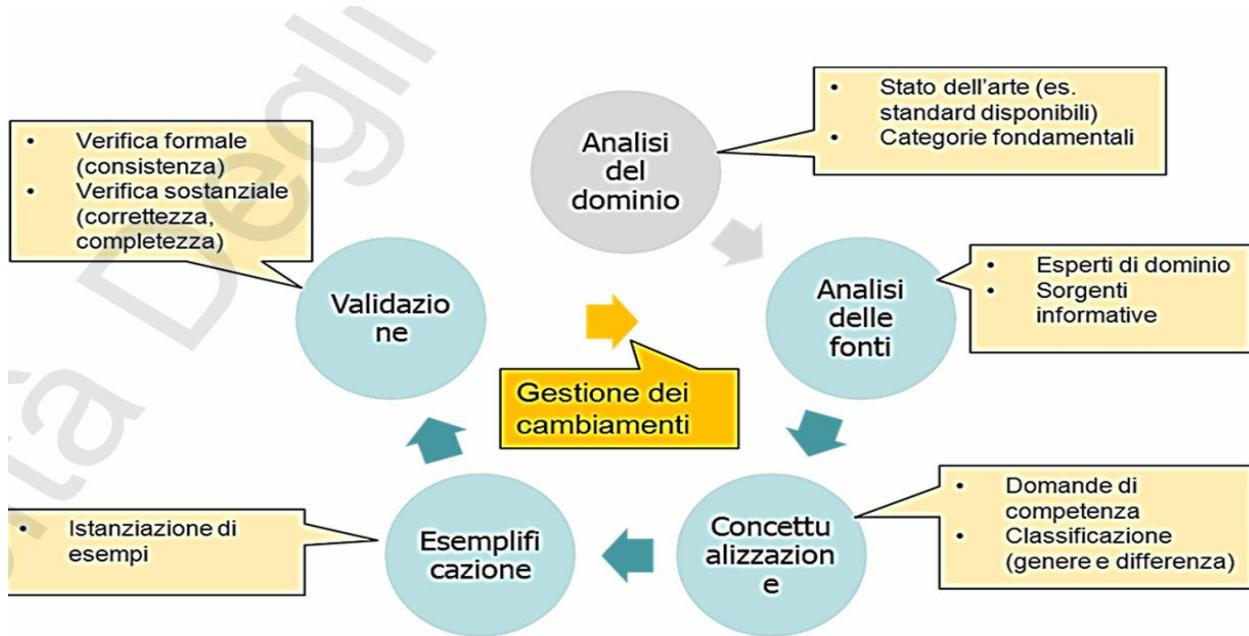
Metonimia: uso di relazioni di dipendenza, es. parte per il tutto.



Categorie ontologiche e domande di competenza

Per orientarsi nella definizione dei concetti, è utile adottare categorie ontologiche di alto livello, come oggetti, eventi, processi o qualità. Esse costituiscono il quadro di riferimento nel quale collocare i concetti specifici del dominio. Le cosiddette “domande di competenza” svolgono qui una funzione metodologica essenziale: servono a verificare se l'ontologia è in grado di rispondere alle questioni fondamentali che riguardano il dominio. La loro formulazione aiuta a chiarire lo scopo della rappresentazione e a testarne la coerenza logica.

Il ciclo di sviluppo dell'ontologia



Le ontologie non sono mai statiche: evolvono con l'evolversi della conoscenza e dei contesti d'uso. Per questo motivo, la manutenzione e il versionamento costituiscono parte integrante del loro ciclo di vita. In genere, ogni ontologia viene pubblicata con un identificatore persistente (URI o IRI) che consente di distinguere le diverse versioni e di garantire la tracciabilità nel tempo.

Strumenti e pratiche di sviluppo

Tra gli strumenti più diffusi per la progettazione ontologica vi è **Protégé**, un ambiente open source sviluppato dall'Università di Stanford. Esso consente di definire classi, proprietà e relazioni, di eseguire controlli di coerenza e di esportare l'ontologia in vari formati OWL. Un esempio didattico noto è la “Pizza Ontology”, che, pur nella sua semplicità, mostra il funzionamento dei principali costrutti logici e il modo in cui essi si riflettono nel reasoning automatico.

Le aree applicative del Web Semantico – Esterno SEO

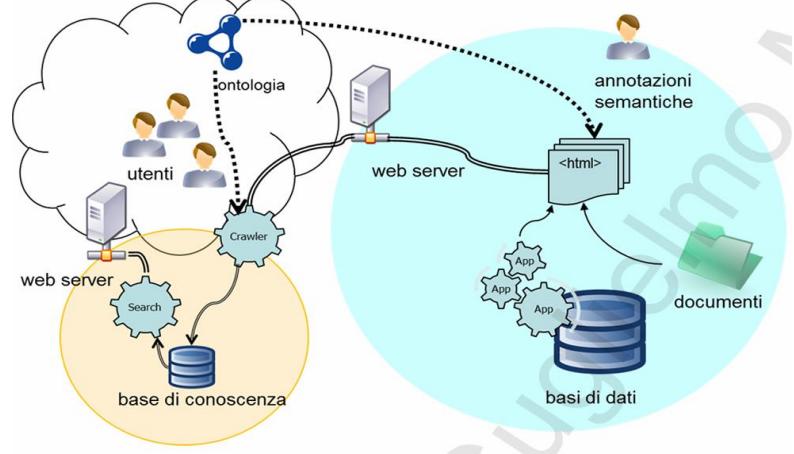
Le applicazioni delle tecniche del Web Semantico possono essere classificate in due grandi ambiti.

Il primo è quello dell'**ottimizzazione per i motori di ricerca** (*Search Engine Optimization, SEO*). Chi produce contenuti sul web — aziende, istituzioni, enti pubblici — desidera che le proprie pagine siano facilmente reperibili e che compaiano tra i primi risultati restituiti dai motori di ricerca, in particolare da Google. Questo ha portato alla nascita di figure professionali dedicate esclusivamente all'ottimizzazione dei contenuti per i motori di ricerca, con l'obiettivo di aumentare la visibilità e la rilevanza dei siti web.

Il secondo ambito applicativo riguarda l'**organizzazione interna della conoscenza** all'interno di un'azienda o di un ente. In questo contesto, le metodologie del Web Semantico vengono utilizzate per consolidare, integrare e gestire le informazioni interne, ricorrendo a rappresentazioni della conoscenza e tecniche di ragionamento automatico.

Architettura generale della pubblicazione e indicizzazione

Dal punto di vista architettonico, possiamo immaginare due entità. Da un lato(blu), un'organizzazione che produce e pubblica contenuti — un'azienda o una pubblica amministrazione — che rende disponibili i propri documenti in formato HTML, sia in modo statico (pagine HTML direttamente codificate e servite via HTTP), sia dinamico (pagine generate da applicazioni che traducono le richieste HTTP in query a database e restituiscono risultati in forma di pagina HTML).



Dall'altro lato(giallo), un'organizzazione che offre **servizi di indicizzazione e ricerca**, come Google o Bing. Questi soggetti utilizzano **crawler** (o *bot*), programmi che esplorano automaticamente il web, scaricano le pagine HTML, le analizzano e costruiscono indici. Tali indici non sono semplici elenchi di parole chiave, ma — nel contesto del Web Semantico — assumono la forma di **basi di conoscenza**, strutture che organizzano e rappresentano l'informazione in modo concettualmente più ricco.

Quando un utente esegue una query, questa viene risolta sull'indice del motore di ricerca, che restituisce i link alle pagine HTML pertinenti. In questo schema, **l'ontologia** svolge un ruolo centrale: fornisce il vocabolario standard e la struttura concettuale sia per le annotazioni semantiche, sia per il ragionamento necessario alla risoluzione delle query.

Annotazioni semantiche: definizione e scopi

Le **annotazioni semantiche** (o *semantic markup*) consistono nell'inserimento di metadati descrittivi direttamente all'interno di un documento HTML. Esistono due principali strategie:

1. utilizzare gli attributi standard del linguaggio HTML, adottando convenzioni condivise sul loro contenuto;
2. introdurre nuovi attributi specifici, pensati per supportare esplicitamente la semantica.

Le annotazioni semantiche devono soddisfare due requisiti fondamentali:

- essere **leggibili dagli esseri umani**, in modo che il significato sia intuitivamente comprensibile;
- essere **facilmente processabili dai crawler**, che devono poter identificare e interpretare le sezioni semantiche del documento.

Storicamente, diversi standard si sono succeduti. Tra i primi troviamo l'**RDFa**, derivato direttamente dal modello RDF (Resource Description Framework), seguito dai **Microformats** e infine dai **Microdata**, oggi lo standard più diffuso.

RDFa

L'RDFa (RDF Annotation), sviluppato dal consorzio W3C, consente di esprimere all'interno di pagine HTML le triple RDF, composte da soggetto, proprietà e oggetto.

Il **soggetto** è, per impostazione predefinita, la pagina stessa; può però essere un suo frammento o anche una risorsa esterna identificata da un URI.

La **proprietà** rappresenta il predicato, cioè la relazione binaria che collega soggetto e oggetto.

L'**oggetto** può essere un'altra risorsa (URI) oppure un valore letterale (literal).

In questo modo, una pagina HTML può essere arricchita con metadati strutturati secondo vocabolari standard, come il **Dublin Core**. Nell'esempio, un'annotazione RDFa può dichiarare che una pagina ha una proprietà "titolo" con un determinato valore. I motori di ricerca in grado di interpretare RDFa possono così rispondere a query basate su proprietà semantiche, non solo su parole chiave.

```
<html  
  xmlns="http://www.w3.org/1999/xhtml"  
  prefix="dc: http://purl.org/dc/terms/">  
<head><title>My Home Page</title></head>  
<body>  
  <h1 property="dc:title">My home-page</h1>  
</body>  
</html>
```

Microformats

I **Microformats**, proposti nel 2005 da una comunità di pratiche indipendente, rappresentano un tentativo di semplificare ulteriormente il processo di annotazione. L'obiettivo era rendere la semantica accessibile anche a chi non avesse competenze tecniche specifiche in RDF.

I Microformats si basano su **pattern predefiniti** per descrivere elementi comuni, come persone, eventi o post. Nell'esempio, la classe HTML vcard identifica un biglietto da visita con nome, cognome e organizzazione.

```
<div class="vcard">
<a class="url fn org" href="http://microformats.org/">
microformats.org
</a>
</div>
```

http://microformats.org

Questo approccio è pratico ma limitato: ogni Microformat ha una struttura rigida e non esistono meccanismi di generalizzazione o estensione. Ciò ne ha limitato l'adozione, rendendo necessario uno standard più flessibile.

Microdata

I **Microdata**, introdotti in HTML5 e promossi dal W3C in collaborazione con Google, Yahoo, Microsoft e Yandex, rappresentano la risposta a questa esigenza. L'obiettivo era coniugare la semplicità d'uso dei Microformats con la maggiore potenza espressiva del modello RDF.

Gli attributi fondamentali di questo schema sono:

- **itemscope**, che definisce l'ambito dell'annotazione;
- **itemtype**, che indica il tipo (cioè la classe ontologica di riferimento);
- **itemprop**, che specifica le proprietà associate agli elementi;
- **itemid**, che identifica un identificatore
- **itemref**, che aggiunge un riferimento ad un altro item nella pagina

Pur non utilizzando formalmente RDF, i Microdata mantengono la stessa struttura logica basata su soggetto, proprietà e valore. La novità è

```
<div itemscope>
<p>My name is <span itemprop="name">Neil</span>.</p>
<p>My band is called <span itemprop="band">Four Parts Water</span>.</p>
<p>I am <span itemprop="nationality">British</span>.</p>
</div>
```

https://www.w3.org/TR/microdata/

che il **vocabolario semantico** — cioè l'insieme di proprietà e classi — non è imposto dallo standard, ma può essere scelto dall'utente. Tuttavia, per garantire interoperabilità e comprensione da parte dei motori di ricerca, è necessario fare riferimento a un vocabolario condiviso.

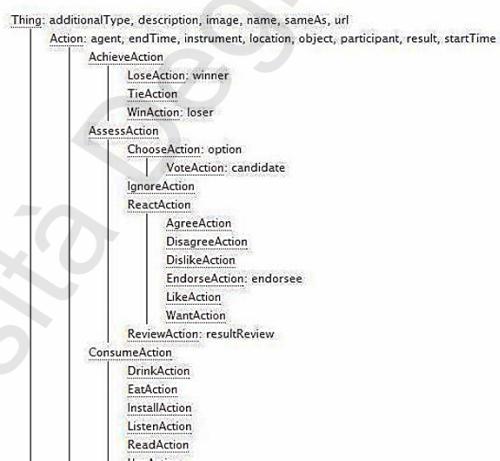
Schema.org è un'iniziativa congiunta di Google, Yahoo, Microsoft e Yandex per creare un vocabolario comune, un'ontologia di riferimento destinata a descrivere i contenuti del web in modo uniforme.

In pratica, Schema.org fornisce un insieme esteso di **classi e proprietà** che coprono i principali domini di contenuto (persone, eventi, organizzazioni, prodotti, luoghi, recensioni, ecc.). Il suo modello concettuale è ispirato a **RDF Schema**, con classi e sottoclassi, proprietà e gerarchie di inclusione, ma senza gli aspetti più complessi di OWL, come la negazione o le disgiunzioni.

Il top-level di Schema.org è la classe generica **Thing**, da cui derivano entità come **CreativeWork**, **Event**, **Organization**, **Person**, **Place** e altre. Sebbene manchi una piena espressività logica, la struttura è sufficiente per rappresentare in modo standardizzato i principali tipi di informazioni presenti sul web.



- I concetti più generali di Schema non si discostano molto da quelli dei «top level» standard
- Notare l'assenza di alcune inclusioni e alcune (implicite) disgiunzioni
- Osservare che i concetti di Schema.org sono tutti primitivi (vs. definiti)



- I concetti specifici di Schema.org sono quelli su cui è possibile fare ragionamenti concreti
- Le scelte ontologiche più rilevanti («che cosa c'è?») si trovano a questo livello
- Le eventuali estensioni fornite dall'utente non sono necessariamente prese in considerazione dal motore di ricerca

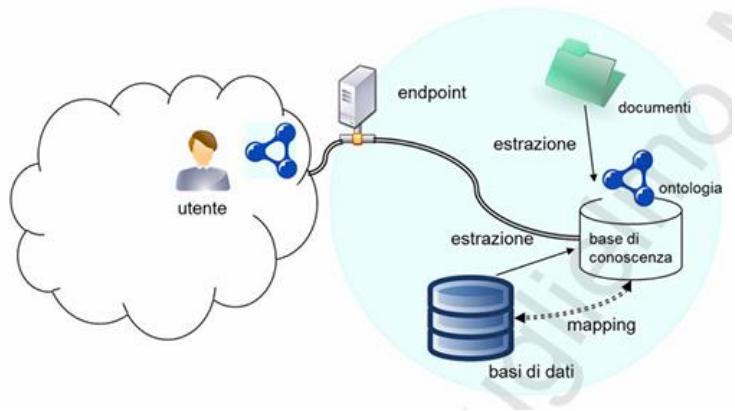
È possibile estendere il vocabolario introducendo nuove classi o proprietà, ma tali estensioni devono essere compatibili con i criteri adottati dai motori di ricerca per poter essere effettivamente utilizzate nei processi di indicizzazione e inferenza.

Il Knowledge Graph di Google

Il **Google Knowledge Graph** rappresenta l'applicazione più importante di Schema.org. Si tratta di una grande base di conoscenza progettata da Google per supportare la ricerca di “things, not strings” — cioè non semplici sequenze di caratteri, ma concetti, entità e relazioni. Quando un utente cerca, ad esempio, “Leonardo da Vinci”, Google non restituisce soltanto un elenco di link, ma anche una **scheda informativa** che riassume la conoscenza su quell'entità: dati biografici, opere, luoghi correlati, immagini. Questa scheda deriva in gran parte da fonti come Wikipedia e Freebase, ma è continuamente arricchita dalle annotazioni semantiche trovate sul web, in particolare da quelle codificate secondo lo standard Schema.org. La speranza dell'information provider è comparire nella scheda informativa, in modo da avere anche un valore alto SEO.

Le aree applicative del Web Semantico - Interno

Lo schema architetturale di riferimento è relativamente semplice: all'interno dei confini organizzativi, intranet, risiedono le basi di dati e le basi documentali. Parte di questa informazione deve essere resa accessibile all'esterno attraverso un endpoint, un portale. Il problema tecnico diventa quindi costruire una base di conoscenza estraendo e integrando dati da queste sorgenti. Gli utenti esterni accederanno alla conoscenza attraverso il portale, comprendendo lo schema concettuale, cioè l'ontologia, del nostro dominio, e potranno così formulare interrogazioni semanticamente rilevanti.



L'ontologia svolge un duplice ruolo: è la concettualizzazione globale del nostro universo informativo ed è anche il linguaggio di mediazione semantica con l'esterno.

Creazione di una base di conoscenza

Quando un'organizzazione decide di esportare i propri dati, può farlo rendendo accessibile direttamente un **dataset** (come avviene per gli open data) oppure collocando i dati dietro **servizi di ricerca** e un portale semantico. I meccanismi di estrazione e costruzione della base di conoscenza sono in parte comuni ai Linked Open Data e ai sistemi con endpoint semantici come quelli basati su SPARQL. La differenza principale è che, negli open data, il carico dell'analisi e dell'integrazione ricade sugli utenti esterni, mentre nel portale semantico tale responsabilità rimane all'interno dell'organizzazione.

In scenari più complessi, l'organizzazione può gestire dati interni, dati raccolti e curati nel tempo, oppure dati provenienti da sorgenti esterne, che possono richiedere operazioni di lettura, pulizia, integrazione e consolidamento. In tutti questi casi occorre considerare:

1. **La provenienza delle informazioni**, che implica anche valutazioni di affidabilità.
2. **L'eterogeneità** nella rappresentazione dei concetti, spesso molto elevata anche all'interno della stessa organizzazione.
3. **La dinamicità**, cioè la frequenza di aggiornamento dei dati.
4. **L'accesso**, cioè il tipo di query e di potenza espressiva che si vuole supportare.

Spesso si lavora anche su documenti HTML annotati con **markup semantico**, oppure su documenti HTML **senza annotazioni** semantiche ma con una struttura nota all'organizzazione. In questi casi è possibile estrarre conoscenza sfruttando convenzioni sulla struttura tipografica del documento. Esistono strumenti specifici per produrre triple RDF da XML o HTML esteso, sviluppati anche nell'ambito del W3C.

Estrazione da testi

Un'altra area fondamentale è l'estrazione di conoscenza da testi, affrontata con tecniche di Natural Language Processing (**NLP**). Questo comprende:

- **Named Entity Recognition**, cioè il riconoscimento di entità nominate (aziende, istituzioni, persone). È un compito complesso perché molte entità sono richiamate tramite espressioni anaforiche o descrittive, che variano nel tempo o nel contesto.
- **Concept recognition (concept insight)**, cioè il riconoscimento della presenza di concetti non direttamente nominati, per esempio identificare che un testo parla del “cinema italiano”. Molte tecniche si basano su informazioni estratte da Wikipedia, e spesso includono machine learning.
- **Relationship extraction**, cioè il riconoscimento di strutture predicative (verbo-argomenti) e dei ruoli semanticici nelle frasi. Questo richiede parsing sintattico, risorse lessicali e tecniche di machine learning su corpora annotati.

Le informazioni estratte devono poi essere trasformate in strutture predicative, tipicamente triple RDF, secondo l'ontologia di riferimento. L'estrazione in lingua italiana è possibile, ma più difficoltosa rispetto all'inglese per via della minore disponibilità di parser open source e risorse lessicali.

Estrazione da basi di dati

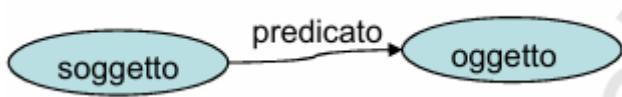
Per i dati relazionali, il problema classico è il mapping tra schema relazionale e schema ontologico. È un'operazione meno

automatizzabile di quanto si pensasse: occorre definire mapping concettuali (global as views - GAV) che associno concetti dell'ontologia a viste (query) sulle tabelle. Le associazioni **intensionali** lavorano sui predicati dell'ontologia, nell'alfabeto dei predicati; quelle **estensionali** riguardano la trasformazione degli identificativi delle basi di dati in URI globali, spesso più complesso del semplice riutilizzo della chiave surrogata del database, questo perché l'ID ha senso solo nella tabella di riferimento.

- Cliente (*ID2URI(ID)*) ↳ SELECT ID FROM UTENTI WHERE AQUISTI > 0
- ordine (*ID2URI(UID)*, *ID2URI(IID)*) ↳ SELECT UID, IID FROM ORDINI

Costruzione della Knowledge Base RDF

Dopo l'estrazione da documenti HTML, testi e basi di dati, si ottiene un **database RDF**, basato sulla struttura soggetto–predicato–oggetto. I database RDF



o triple store, oggi spesso inglobati nel concetto più ampio di database a grafo, sono ottimizzati per questo modello e supportano interrogazioni congiuntive. Tra le piattaforme più note: Jena, Sesame (oggi RDF4J) e Virtuoso. I database a grafo condividono la stessa modellazione, pur differendo nel focus sulla navigazione di nodi più che sulle query congiuntive. Entrambi i modelli offrono un vantaggio rispetto ai sistemi relazionali: lo schema concettuale è disgiunto dalla struttura fisica dei dati, facilitando l'evoluzione dello schema, è a livello dei dati, software. Il limite principale, comune a RDF store e graph database, riguarda le prestazioni nelle query congiuntive complesse, campo in cui i DB relazionali hanno una storia molto più lunga. Il loro punto di forza è la search and expansion cioè espandere la ricerca operativa nei contorni concettuali di quell'entità.

Approccio basato sulla virtualizzazione

Esiste anche l'approccio della virtualizzazione: invece di materializzare la knowledge base, si mappa l'ontologia sui database e si riformulano le query dell'ontologia in query sulle sorgenti (ad esempio SQL). Questo approccio, noto come Ontology-Based Data Access (OBDA), è sostenuto dal profilo OWL QL ed è sviluppato in particolare nelle università italiane.

Query congiuntive

Le query congiuntive corrispondono, intuitivamente, alle query SQL basate su selezione, proiezione e join. Formalmente, sono formule del primo ordine costituite da formule atomiche congiunte, variabili libere e variabili quantificate esistenzialmente. L'esecuzione su una knowledge base produce **risposte certe**, cioè risposte che sono vere in tutti i modelli compatibili con la base di conoscenza, sotto un approccio a mondo aperto. Il motore di inferenza utilizza un determinato **entailment regime**, definendo quali assiomi dell'ontologia vengono considerati nel ragionamento.

SPARQL

Per supportare questo modello, il W3C ha sviluppato SPARQL (2008), linguaggio che implementa query congiuntive su RDF, con prefissi, pattern e corrispondenze alle triple. SPARQL 1.1 permette anche di specificare l'entailment regime richiesto.

- Le query congiuntive sono formule di logica del primo ordine che sono costruite solo con:
 - formule atomiche;
 - connettivo di congiunzione (\wedge);
 - quantificatore esistenziale (\exists).
- La forma di una query congiuntiva presenta variabili libere (distinte), variabili vincolate, e atomi predicativi su entrambi i tipi di variabile:
 - $(x_1 \dots x_n) . \exists x_{n+1} \dots \exists x_{n+m} . A_1 \wedge \dots \wedge A_k$
 - $(name, email) . \exists person . Person(person) \wedge name(person, name) \wedge mbox(person, email)$.

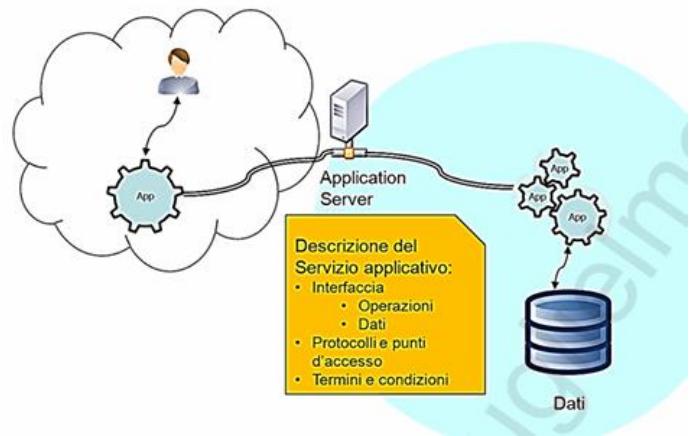
SPARQL supporta query congiuntive (anche con unioni) sulle triple, con facilitazioni sintattiche per RDF:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?email
WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:mbox ?email.
}
```

Integrazione semantica di servizi

Secondo la definizione classica del consorzio OASIS, un servizio è un meccanismo che consente l'accesso a una o più funzionalità software tramite un'interfaccia predefinita e coerente con vincoli e politiche indicate nella descrizione del servizio. Un servizio è dunque un'interfaccia tecnica per accedere a funzionalità remote e, allo stesso tempo, un “contratto” che specifica aspetti come qualità del servizio, disponibilità e condizioni d'uso offerte dal provider.

Un servizio è una funzionalità unitaria, erogata autonomamente da un fornitore attraverso un protocollo di accesso client-server. Offre una descrizione dei dati, delle operazioni e dei vincoli relativi al suo uso, e cerca di minimizzare le dipendenze dal contesto operativo: è il principio del *loose coupling*, ovvero accoppiamento debole tra client e server. L'idea è quella dell'isolamento: un servizio deve essere *self-contained*, richiedere al client solo la conoscenza dell'interfaccia e nulla altro.



Tipicamente i servizi software sono utilizzati da applicazioni, non direttamente dagli utenti; l'eventuale interazione con l'utente passa sempre attraverso uno strato applicativo e una user interface. Le interfacce di servizio sono spesso incapsulate in API, ad esempio in Java o PHP. Il protocollo di trasporto è quello dell'Internet stack, e a livello applicativo il protocollo è HTTP.

Le interfacce dei servizi sono definite da due elementi fondamentali:

1. **Operazioni** attivabili tramite il servizio;
2. **Tipi di dati** scambiati tra client e server.

Web Services: WSDL, SOAP e UDDI

Poiché i servizi software sfruttano l'infrastruttura del Web, non sorprende che il W3C abbia definito standard specifici, dando origine ai “web services”. Questi servizi scambiano dati strutturati, non pagine HTML: è dunque naturale l'uso dell'XML come linguaggio di strutturazione.

Il lavoro del W3C si fonda su due pilastri:

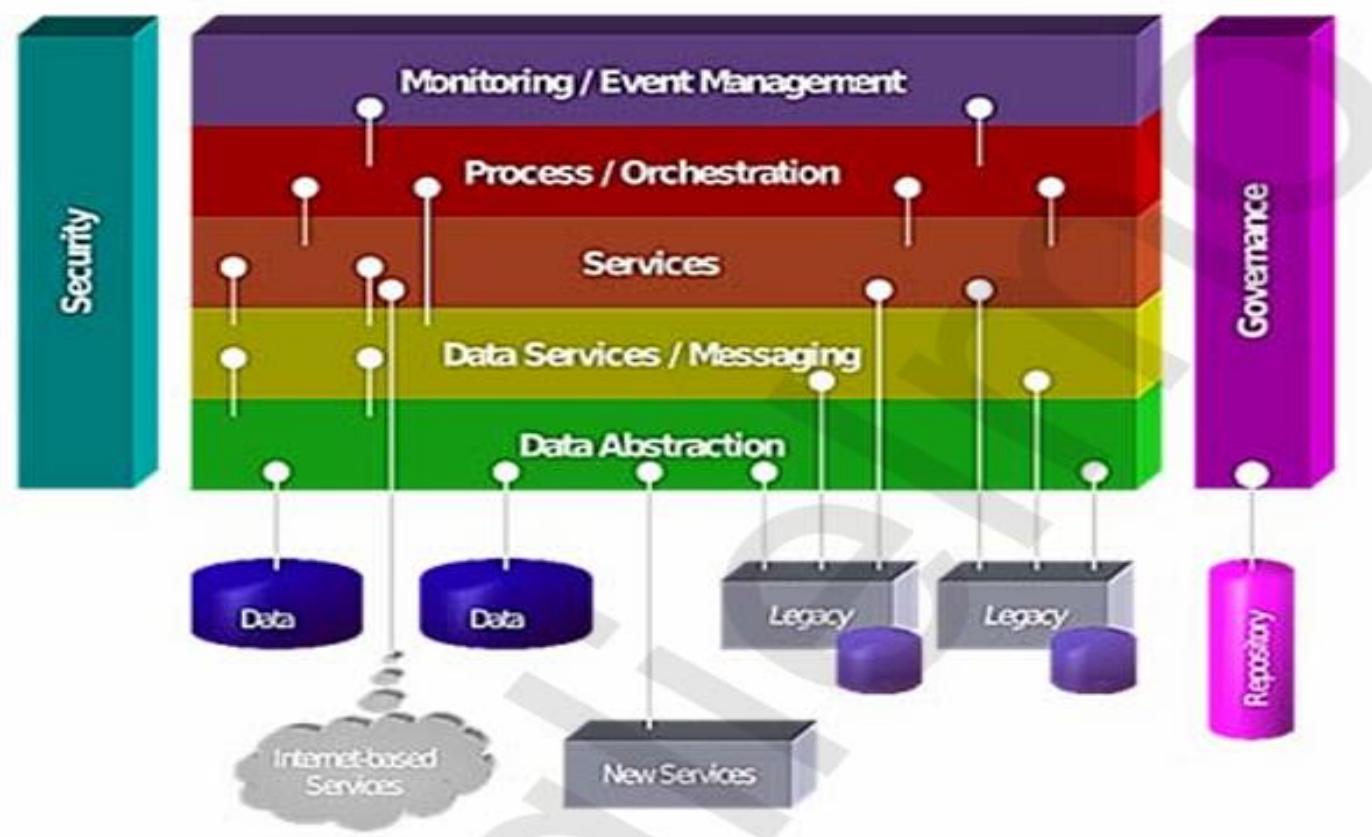
1. **WSDL (Web Service Description Language)**: definisce l'interfaccia del servizio, cioè operazioni e strutture dati. È lo standard per la descrizione statica del servizio.
2. **SOAP (Simple Object Access Protocol)**: definisce il formato dei messaggi scambiati dinamicamente tra client e server.
3. Un ulteriore elemento, standardizzato da OASIS, è **UDDI**, che specifica come organizzare le descrizioni dei servizi in registri interrogabili dagli utenti, supportando così la pubblicazione e la scoperta dei servizi.

Architetture orientate ai servizi (SOA)

Attorno alla nozione di servizio software nasce l'idea di architettura orientata ai servizi (SOA), un modello di cooperazione e integrazione tra componenti autonome (*self-contained* e *loosely coupled*) che collaborano per fornire funzionalità applicative complesse.

Una SOA prevede diversi livelli:

- **Descrizione dei dati e della messaggistica**, tramite standard come WSDL e SOAP.
- **Orchestrazione dei servizi**: definisce come diversi servizi possono essere integrati e coordinati. L'orchestrazione gestisce flussi, failure, latenze, sicurezza e governance tramite motori specializzati.
- **Monitoraggio**: una volta che l'orchestrazione è operativa, è necessario monitorare l'intero sistema per individuare malfunzionamenti, analizzare latenze e gestire i *point of failure*.



L'esigenza di semantica nei servizi

Come il Web semantico ha affrontato il problema del significato dei contenuti nelle pagine HTML, analogamente nei servizi software sorge l'esigenza di **caratterizzare semanticamente i dati scambiati**. Dal primo decennio degli anni 2000 diversi gruppi di ricerca, tra cui **DERI**, hanno lavorato su questo tema seguendo due filoni principali:

1. **Descrizione logica completa dei servizi**, tramite linguaggi come OWL (entrambi mirano a descrivere logicamente servizi, interfacce, flussi e pattern di interazione).
 - WSMO (Web Service Modeling Ontology)
 - OWL-S (Semantic Markup for Web Services)
2. **Annotazioni semantiche delle descrizioni WSDL**, un approccio più leggero e compatibile con gli standard esistenti.
 - WSDL-S (Web service Semantics)
 - SA-WSDL (Semantic Annotations for WSDL and XML Schema)

Il risultato più maturo è **SA-WSDL** che permette di mappare operazioni e strutture dati del servizio a qualsiasi concetti definiti in qualsiasi ontologie esterne.

L'affermazione delle architetture REST problemi di Semantica

Negli ultimi anni ha prevalso un approccio più semplice e “low-level”: le architetture **REST** (Representational State Transfer), concettualizzate attorno al 2000. Esse osservano come molti servizi siano stati sviluppati in modo più leggero rispetto agli standard formali dei web services.

Caratteristiche principali del REST:

- Uso diretto di HTTP come mediazione, senza protocolli aggiuntivi come SOAP.
- Sfruttamento di URL e dei verbi principali (GET e POST).
- Chiara divisione di responsabilità tra client e server.
- **Stateless**: la risposta del server non dipende dalla storia delle richieste precedenti.
- Gestione della cache tramite scadenza dei dati.
- Possibilità di comporre servizi tramite catene di invocazioni.

I dati scambiati sono tipicamente strutturati in XML o, più frequentemente oggi, in JSON. Anche in REST emerge il problema della **semantica**: occorre garantire coerenza tra invocazioni, dati restituiti e concetti applicativi. Non essendoci specifici livelli architetturali, ci si affida soprattutto a buone pratiche nella progettazione degli URI e, più recentemente, a tentativi di annotazioni ispirate ai microformats. Tool come Swagger, molto diffusi per documentare API REST, sono invece sostanzialmente agnostici rispetto alla semantica.

Il problema dell'interoperabilità semantica

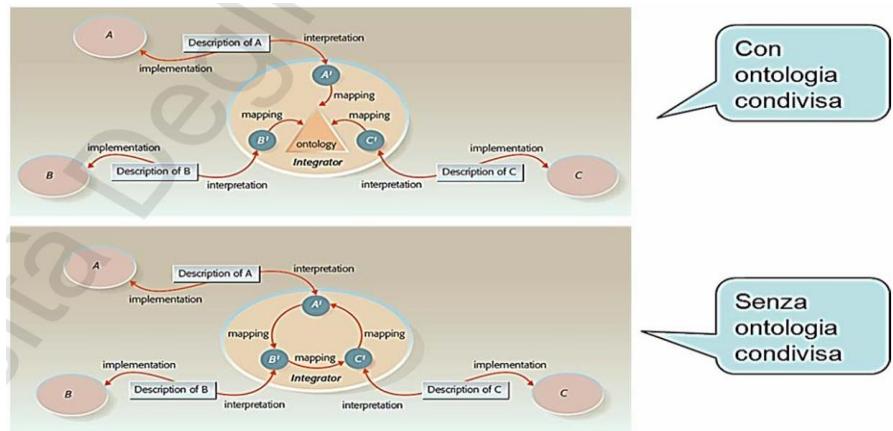
L'interoperabilità semantica consiste nella capacità di servizi diversi di integrarsi e collaborare in modo trasparente rispetto al significato dei dati scambiati. Il problema riguarda i dati prodotti e mantenuti da ciascun service provider, che devono essere interpretati correttamente da chi li riceve.

Il quesito centrale è: **come garantire che l'interpretazione dei dati rimanga coerente tra sorgente e destinatario?**

Dal punto di vista teorico, il problema è insolubile a causa dell'elemento di arbitrarietà intrinseco nel significato dei simboli (come insegnava il triangolo semiotico di Peirce e di Ogden & Richards). Tuttavia esistono modi per contenere le divergenze semantiche.

Le trasformazioni e i mapping tra schemi semanticci avvengono spesso in modo distribuito, e a seconda di dove tali operazioni sono collocate si generano diversi modelli architetturali. Tra essi:

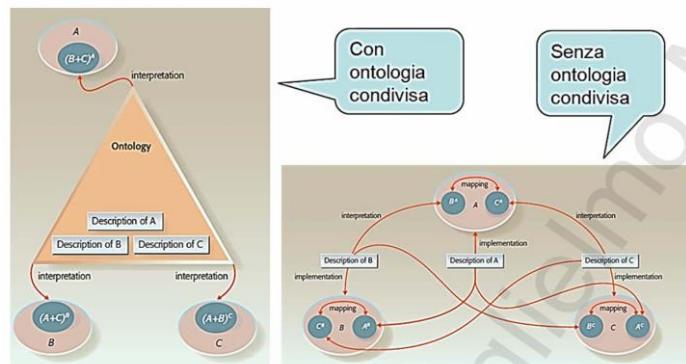
- **Modello centralizzato con ontologia condivisa** (simile a data warehousing o master data management).
- **Modello distribuito con trasformazioni centralizzate**, tipico degli enterprise service bus.
- **Modello completamente decentralizzato peer-to-peer**, con ontologie e mapping locali.



	Ontologia condivisa	Ontologia distribuita
Trasformazioni centralizzate	Es. Master Data Management	Es. Enterprise Service Bus
Trasformazioni locali	Es. Knowledge Graphs	Es. Semantic Web

Il modello **decentralizzato**, su cui hanno lavorato vari gruppi di ricerca, tra cui quello dell'Università di Roma, mantiene coerente la visione architettonale del Web, ma introduce problemi complessi, soprattutto epistemici: ad esempio, come gestire inconsistenze tra dati provenienti da sorgenti diverse? Se due servizi forniscono orari ferroviari divergenti, come posso determinare ciò che è vero? Le soluzioni non sono generali: si possono introdurre valutazioni di affidabilità delle fonti o altre strategie, ma non esistono metodi universalmente validi.

Integrazione decentralizzata (peer-to-peer)



Linked open data

Fin dai primi anni del successo del web come infrastruttura di condivisione dell'informazione, si sviluppa un movimento che promuove l'apertura, la disponibilità e la libera accessibilità dei contenuti informativi. Tale movimento si affianca a quello dell'open source, che negli anni Novanta riceve un impulso decisivo grazie all'iniziativa di Linus Torvalds e del sistema operativo Linux. L'idea di rendere pubblico il codice del kernel Unix e di chiamare a collaborare una comunità di sviluppatori dimostra i vantaggi della condivisione della conoscenza tecnica. Il progetto si rivela così solido e affidabile da essere infine adottato anche dall'industria.



A questo fenomeno si affianca quello di Wikipedia, iniziativa che ribalta il modello editoriale tradizionale: non più un'enciclopedia scritta da un ristretto gruppo di esperti, ma un'opera collettiva, redatta dalla comunità degli utenti. Quella che inizialmente sembrava un'impresa utopica si rivela invece straordinariamente efficace.

Questi casi di successo alimentano un movimento globale che sottolinea la necessità di rendere disponibili alla collettività dati di ogni tipo e provenienza. L'accesso diretto ai dati favorisce, tra le altre cose, maggiore trasparenza amministrativa. Ne è un esempio il bilancio partecipato, in cui la cittadinanza può esprimere priorità e valutazioni sulle scelte di allocazione delle risorse pubbliche, grazie alla disponibilità aperta dei dati di bilancio.

All'apertura dei dati è associata anche la possibilità di creare valore economico e sociale. Molte applicazioni per dispositivi mobili, ad esempio, sfruttano dati pubblici sui trasporti o sul traffico, messi a disposizione dalle amministrazioni in formati adeguati. La combinazione e rielaborazione di questi dati permette di offrire servizi ai cittadini e genera nuove opportunità economiche. In parallelo, l'accesso aperto ai dati favorisce forme di partecipazione politica e di governo della comunità più informate e consapevoli.

A livello culturale, tutto ciò si colloca dentro una più ampia visione di **“nuovo illuminismo”**: la disponibilità dei dati promuove la costruzione collaborativa della conoscenza. Il dato in sé, inteso nella prospettiva della teoria dell'informazione di Claude Shannon, è semplice informazione priva di significato. Diventa conoscenza solo quando è interpretato e trasformato in contenuto semanticamente strutturato. Il movimento open data sostiene che l'accesso libero e organizzato ai dati favorisca proprio questo processo di trasformazione.



I requisiti dei dati aperti

La disponibilità dei dati aperti richiede il rispetto di alcuni requisiti fondamentali:

1. **Licenza aperta:** una licenza è definita aperta quando permette l'uso e la redistribuzione dei dati senza oneri e per qualsiasi scopo. Le licenze Creative Commons, in alcune configurazioni, soddisfano questi criteri, anche se molte varianti introducono restrizioni (ad esempio il divieto di uso commerciale).
2. **Accessibilità completa e gratuita via Internet:** i dati devono essere scaricabili integralmente, senza costi e senza barriere tecnico-organizzative.
3. **Formato aperto:** un formato è aperto quando la sua specifica tecnica è pubblica e mantenuta da una comunità di standardizzazione, non da un singolo attore privato. L'obiettivo è evitare che l'uso dei dati sia vincolato a software specifici o a formati proprietari.

I principali ambiti applicativi

Sono stati identificati vari settori nei quali gli open data hanno particolare rilevanza:

- **Beni culturali** (cataloghi museali, letteratura non più coperta da copyright, progetti di editoria libera come Liber Liber).
- **Ricerca scientifica**, i cui dati sono spesso prodotti con fondi pubblici ma non sempre resi disponibili ai cittadini. In Italia, un esempio virtuoso è l'ISTAT.
- **Pubblicazioni scientifiche**, spesso soggette a costi elevati nonostante siano il prodotto di istituzioni pubbliche.
- **Dati economici e statistici**, fondamentali anche per il giornalismo basato su evidenze.
- **Dati meteorologici, ambientali, geografici e di trasporto**, molto rilevanti per applicazioni e servizi.

L'analisi degli usi effettivi effettuata da iniziative europee come Open Data Monitor mostra che una parte significativa degli utenti scarica i dati per sperimentare ("to play"), mentre una quota rilevante li impiega per sviluppare applicazioni, analisi di business e data journalism.



Formati per gli open data

I dati possono essere:

- **Non strutturati o semistrutturati** (testo),
- **Strutturati** (fogli di calcolo, tabelle relazionali).

Un aspetto centrale nella pubblicazione dei dati aperti riguarda il tema della standardizzazione dei formati. Parlare di formato aperto significa infatti riferirsi non soltanto alla disponibilità pubblica delle sue specifiche tecniche, ma anche al fatto che tali specifiche siano definite, mantenute e aggiornate all'interno di un processo riconoscibile, documentato e partecipato. La standardizzazione assicura che un formato non sia controllato da un singolo soggetto, ma sia il risultato di un lavoro collettivo che ne garantisce stabilità, interoperabilità e neutralità tecnologica.



Nel contesto degli open data, questa caratteristica è essenziale: se un formato fosse legato a un software proprietario o a una tecnologia non documentata, l'accesso ai dati risulterebbe di fatto limitato. L'adozione di standard aperti permette invece a chiunque di leggere, elaborare o riutilizzare le informazioni senza vincoli tecnici o economici, favorendo così la massima riusabilità dei dataset.

La presenza di uno standard comune consente inoltre di trattare i dati in modo coerente anche quando provengono da fonti diverse. Ciò è evidente sia nei formati testuali, come RTF o ODT, sia nei formati per dati strutturati come CSV, XML o JSON, il cui funzionamento si basa su convenzioni pubbliche e stabili. In questo senso, la standardizzazione non è un elemento accessorio, ma uno dei presupposti fondamentali dell'apertura: garantisce che i dati possano essere compresi, interpretati e integrati senza ambiguità e favorisce la costruzione di ecosistemi informativi realmente interoperabili.

Per i testi semplici può essere usato il formato ASCII puro (TXT), ma ciò comporta la perdita di strutture tipografiche. Esistono però formati testuali aperti che preservano la struttura editoriale: **RTF, PDF, ODT**. I formati Word (.doc) sono documentati, ma in molte versioni non rispettano pienamente i criteri dei formati aperti.

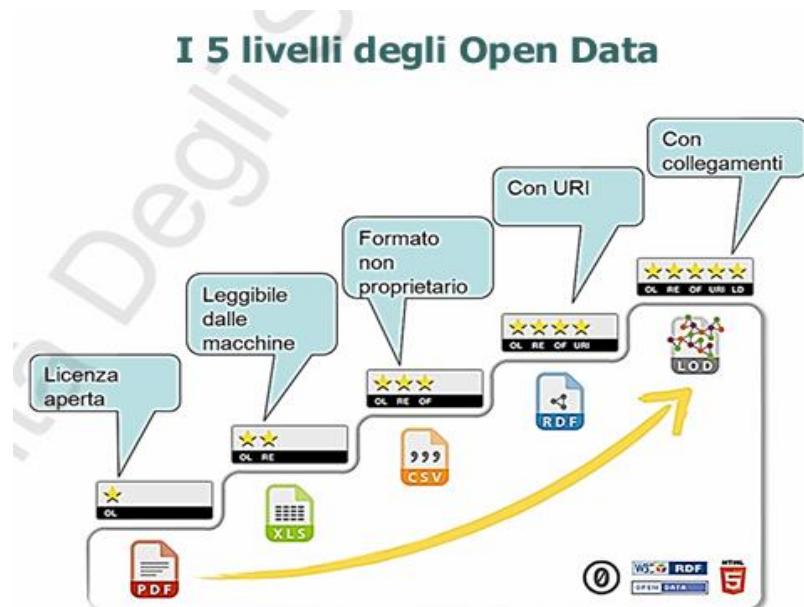
Per i dati strutturati si utilizzano formati come **JSON, XML, CSV** e, nel contesto del web semantico, **RDF**. Il CSV è molto diffuso ma espone a rischi di allineamento; JSON e XML sono più espressivi; RDF introduce inoltre la possibilità di descrivere relazioni semantiche.

Linked data e linked open data

Intorno al 2006 Tim Berners-Lee osserva che, se i dati rimangono chiusi negli archivi, gli sforzi del Semantic Web non raggiungono il loro obiettivo. Propone quindi di pubblicare i dati sul web in modo che possano essere collegati tra loro. Se collegare documenti ipertestuali significa creare link tra pagine, collegare dati significa riferirsi alle stesse entità del mondo mediante gli stessi identificatori. Questo permette di connettere dataset diversi attraverso riferimenti condivisi.

I dati che supportano questi collegamenti sono detti **linked data**; quando vengono pubblicati secondo i principi degli open data diventano **linked open data**.

Berners-Lee propone un sistema di valutazione a cinque stelle per classificare il grado di apertura e interoperabilità:



1. ★ Dato disponibile sul web con licenza aperta, anche in formati non leggibili dalle macchine (es. PDF).
2. ★★ Dato leggibile automaticamente (machine readable).
3. ★★★ Dato leggibile automaticamente e in formato non proprietario (es. CSV).
4. ★★★★ Uso di URI per identificare le entità, tipicamente tramite formati come RDF.
5. ★★★★★ I dataset utilizzano URI che consentono di collegarli ad altri dataset, realizzando l'integrazione su scala web.

Gli identificatori URI rappresentano quindi l'elemento unificante alla base dell'integrazione dei linked open data e costituiscono un requisito essenziale per garantire qualità, interoperabilità e intelligibilità dei dati.

- I fornitori di LOD possono seguire alcune «buone pratiche» (es. www.w3.org/TR/cooluris/):
 - Associare informazione descrittiva alle URI, leggibile agli umani (es. html) e alle macchine (es. rdf).
 - Distinguere le URI delle entità da quelle dei documenti:
 - Usare hash (#) per isolare l'identificativo.
 - Usare meccanismi di redirectione (HTTP 303).
- Da considerare anche la «persistenza» delle URI:
 - Convenzioni di temporizzazione (es. Permalink).
 - Registri di redirectione (es. PURL).

Semantica nella pubblica amministrazione

Il quadro europeo

L'Europa discute da decenni di interoperabilità e integrazione dei sistemi informativi pubblici, sia a livello nazionale sia transnazionale. L'Agenda Digitale Europea, parte della strategia Europa 2020, considera questi aspetti centrali per conseguire obiettivi di crescita economica e di partecipazione civile, anche nei processi decisionali.



L'Agenda Digitale Europea afferma che il suo scopo generale è ottenere vantaggi socio-economici sostenibili tramite un mercato digitale unico fondato su connettività veloce e su applicazioni interoperabili. Viene dato rilievo non solo ai benefici economici, ma anche ai benefici sociali, e viene sottolineata l'importanza di un mercato unico in cui le aziende possano operare oltre i confini nazionali grazie a norme condivise.

Per quanto riguarda i sistemi ICT, l'Agenda si concentra su due dimensioni: la connettività infrastrutturale (banda larga e ultra-larga) e l'interoperabilità applicativa. Avere una buona connettività non basta: occorre che servizi e sistemi possano integrarsi senza barriere tecniche od organizzative. L'Agenda osserva che l'Europa non sfrutta ancora appieno i vantaggi dell'interoperabilità e ribadisce che le diverse componenti devono basarsi su standard comuni e piattaforme aperte.



Il supporto operativo a questi obiettivi è affidato ai programmi **ISA** (Interoperability Solutions for European Public Administration). Si tratta di programmi finanziati dalla Commissione Europea per favorire l'interoperabilità nei sistemi della pubblica amministrazione. Il programma è stato finanziato inizialmente dal 2010 al 2015 e poi rifinanziato fino al 2020, generando bandi e progetti su vari aspetti dell'interoperabilità.

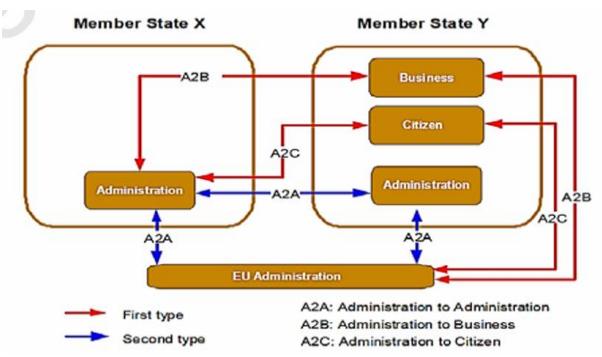


Tra le azioni finanziate vi sono sia interventi generali sull'interoperabilità infrastrutturale, sia iniziative specifiche relative all'interoperabilità semantica e agli open data. L'attenzione verso gli open data deriva dal fatto che la loro connessione e integrazione mette in evidenza la necessità di una semantica condivisa tra dataset pubblici.

Il coordinamento del programma ISA si articola in iniziative dedicate: la definizione di una strategia di interoperabilità, lo sviluppo di un framework comune, una reference architecture e iniziative più tecniche, come quelle legate alla cartografia.

European Interoperability Framework (EIF)

Lo **European Interoperability Framework** è una collezione di definizioni, modelli, linee guida, raccomandazioni, standard e vocabolari per l'interoperabilità dei servizi pubblici a livello europeo. Il suo scopo è permettere, ad esempio, la gestione semplice e sicura della mobilità dei cittadini all'interno dell'UE, rendendo interoperabili i registri anagrafici e molti altri sistemi informativi. L'EIF non è un software da installare, ma un insieme di documenti che forniscono indicazioni ai singoli Stati. Queste linee guida, attraverso un processo che va dalle normative ai bandi pubblici fino ai collaudi dei sistemi reali, si trasformano in specifiche tecniche operative.



L'EIF distingue tre livelli di interoperabilità:

- **Interoperabilità tecnica:** protocolli di trasporto e protocolli applicativi.
- **Interoperabilità semantica:** condivisione del significato dei dati scambiati.
- **Interoperabilità organizzativa:** integrazione dei processi amministrativi tra paesi diversi.

Individua inoltre tre grandi aree applicative: interazioni tra imprese e amministrazioni (incluse procedure come l'apertura di un'azienda o la partecipazione a un bando pubblico), interazioni tra cittadini e amministrazioni, e interazioni tra amministrazioni.

I core vocabularies europei

Per l'interoperabilità semantica un punto cruciale è lo sviluppo dei core vocabularies, veri e propri modelli concettuali (ontologie) che organizzano lo spazio semantico dei sistemi informativi pubblici. L'importanza dei core vocabularies è cresciuta nel tempo, portando a iniziative sempre più specifiche per promuoverne l'adozione.

Attualmente sono sviluppati quattro vocabolari principali:

- **Core Person:** modella le informazioni anagrafiche della persona (simile alla FOAF ontology).
- **Core Business / Core Organization:** descrive organizzazioni ed entità legali.
- **Core Location:** definisce concetti geografici e articolazioni territoriali.
- **Core Public Service:** modella cosa sia e come descrivere un servizio della pubblica amministrazione.

Accanto a questi vocabolari, il programma ISA finanzia la comunità **SEMIC – Semantic Interoperability Community**, ospitata sulla piattaforma di partecipazione civica **JoinUp**. È una comunità aperta a tutti, con lo scopo di contribuire allo sviluppo, all'adozione, alla promozione e alla diffusione dei core vocabularies. La comunità raccoglie buone pratiche, esperienze, metodologie, linee guida, casi di studio, e promuove l'uso dei linked open data governativi. Organizza inoltre una conferenza annuale, occasione di confronto tra amministrazioni di tutta Europa.

Il quadro italiano

In Italia la riflessione sull'integrazione dei sistemi pubblici inizia negli anni Novanta e si concretizza nel 2005 con il **Decreto Legislativo 7 marzo 2005, n. 82**, noto come **Codice dell'Amministrazione Digitale** (CAD). È un testo elaborato dal Governo per delega del Parlamento, e dal 2006 in poi è stato oggetto di continui aggiornamenti, che ancora proseguono, per adeguarlo all'evoluzione tecnologica.



Il **Capo VII** del CAD definisce il **Sistema Pubblico di Connettività** (SPC). L'obiettivo è dotare le pubbliche amministrazioni di un'infrastruttura unica di connettività e di regole tecniche per sviluppare, integrare e condividere il patrimonio informativo, garantendo interoperabilità di base ed evoluta, sicurezza, riservatezza e autonomia delle singole amministrazioni.

Un punto essenziale è proprio la salvaguardia dell'autonomia amministrativa, coerente con la Costituzione italiana e con il modello policentrico del sistema pubblico. Ciò implica un'architettura federata, non gerarchica, nella quale ogni amministrazione conserva la gestione dei propri sistemi e opera in condizioni di parità.

Il SPC deve inoltre garantire economicità, apertura del mercato e concorrenza, evitando fenomeni di vendor lock-in. L'implementazione concreta è affidata all' **Agenzia per l'Italia Digitale (AgID)**.



La cooperazione applicativa nel SPC

La cooperazione applicativa, secondo il CAD, assume la forma di un sistema **peer-to-peer**. Ogni amministrazione definisce i propri servizi e stipula “accordi di servizio” con le altre amministrazioni con cui deve interagire. Non esiste un servizio centralizzato unico (come un e-government service bus), ma una rete di scambi tra pari basata su specifiche condivise. Gli accordi di servizio contengono: interfacce operative, dati e ontologie utilizzate e livelli di servizio (SLA).



Il modello tecnico adottato è quello dei **web services**, conformi agli standard W3C. Questa scelta ha garantito apertura al mercato, trasparenza e possibilità per qualunque fornitore di sviluppare servizi conformi.

Le annotazioni semantiche sono espresse tramite **SA-WSDL** (Semantic Annotations for WSDL), divenuto raccomandazione W3C proprio mentre si stava definendo l'infrastruttura SPC, consentendone così l'adozione.

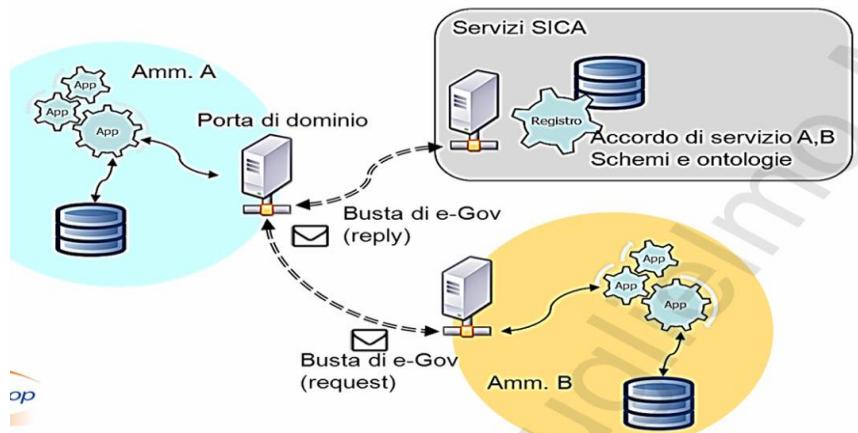
SPC Coop

L'architettura **SPC Coop** segue il paradigma **SOA** e utilizza la busta **SOAP**, estesa con attributi specifici richiesti da un componente chiamato **porta di dominio**, responsabile della gestione di sicurezza, autenticazione, autorizzazione, logging, diagnostica e tracciabilità delle transazioni.

Un elemento centrale, benché l'architettura sia peer-to-peer, è il servizio **SICA**, un registro che memorizza e rende ricercabili gli accordi di servizio e le ontologie pubblicate dalle amministrazioni, estendendo e specializzando le funzionalità dei registri UDDI. Accanto a questo esiste un indice delle pubbliche amministrazioni e delle loro articolazioni organizzative.

Architettura SOA:

- «Busta di e-Gov» (SOAP Envelope 1.1):
 - Intestazione (Header) infrastrutturale.
 - Corpo (Body) applicativo.
- «Porta di Dominio» (message proxy):
 - Intermediario per i servizi applicativi di ciascuna amministrazione: gestisce messaggistica operata attraverso buste di e-Gov.
 - Identificazione, autenticazione, confidenzialità.
 - Sicurezza, diagnostica, cronologia.

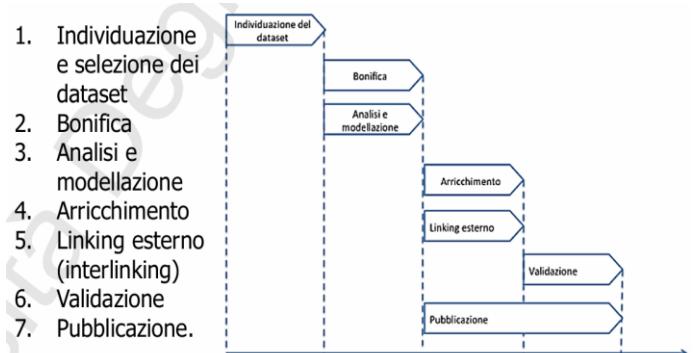


Valutazione dell'esperienza e direzioni attuali

L'architettura **SPC Coop** è entrata in esercizio tra il 2008 e il 2010. Sebbene tecnicamente solida, è risultata ampiamente sottoutilizzata: poche amministrazioni hanno definito ontologie o popolato il servizio SICA. Le ragioni risiedono soprattutto nella distanza tra la complessità dell'architettura e il livello di maturità tecnica reale della pubblica amministrazione.

Oggi si registra un cambio di rotta verso i **Linked Open Data**. È in corso una nuova gara – in parte già aggiudicata – sui servizi di interoperabilità e cooperazione applicativa dei dati, orientata al modello dei dataset pubblicati come open data o come endpoint SPARQL. Il modello è meno complesso: si possono pubblicare dataset bonificati, organizzati semanticamente, con la possibilità di interrogarli tramite query congiuntive, senza dover definire specifici web service e ontologie applicative.

Le linee guida italiane per gli open data riprendono i cinque livelli del modello W3C e identificano gli ambiti prioritari di pubblicazione (dati territoriali, ambientali, del personale, scolastici, universitari, di ricerca, competenze, dati statistici). Richiamano esplicitamente le best practice W3C e le indicazioni dei programmi ISA. Per la pubblicazione dei linked open data viene descritto un processo metodologico consolidato (vedi schema).



Conclusioni e prospettive Linked Data e Web Semantico

In logica, una **proposizione** è una formula linguistica – di un linguaggio formale – suscettibile di essere vera o falsa. È un *truth bearer*, un portatore di verità.

Un'**interpretazione** è un'assegnazione di valori di verità alle proposizioni nel caso del calcolo proposizionale; nel caso della logica predicativa, secondo la visione di Tarski, un'interpretazione assegna individui alle costanti individuali e sottoinsiemi del dominio ai predicati.

Nel contesto logico questa attribuzione è un fatto tecnico e arbitrario: si assume semplicemente che certe proposizioni valgano vero o falso senza indagarne le ragioni psicologiche o sociali. In linguistica, invece, l'attribuzione della verità ha una complessità enorme, legata ai processi cognitivi e comunicativi. La logica non entra in questo livello: non spiega *perché* si attribuisce un valore di verità, ma solo *come* certe proposizioni derivino logicamente da altre.

- Proposizione: formula linguistica dotabile di verità.
- Interpretazione: assegnazione di verità a proposizioni.
- Verità: corrispondenza tra proposizioni e stati di cose.
- Credenza: atteggiamento di un agente rispetto alla verità di una proposizione.
- Conoscenza: credenza giustificata in proposizioni vere.
- Logica: sistema formale che definisce l'inferenza valida di proposizioni da altre proposizioni.
- Teoria: sistema composto da proposizioni (assunte come) vere, regole di inferenza logica, e tutto ciò che ne consegue.
- Ontologia: teoria su entità concrete o astratte, o stati di cose reali o immaginari.

La **verità**, in senso tecnico, è la corrispondenza tra proposizioni e stati di cose. Ma mentre nei sistemi formali è assegnata a priori e meccanicamente, nei sistemi sociali la verità è frutto di mediazione, negoziazione, processi cognitivi complessi.

La **credenza** è un atteggiamento proposizionale: un agente può credere vera o falsa una proposizione.

La **conoscenza**, nella classica definizione filosofica, è credenza giustificata in proposizioni vere.

La **logica**, in generale, è un sistema formale che definisce le inferenze valide: stabilisce come derivare alcune proposizioni da altre, assumendo come date le attribuzioni di verità.

Una **teoria**, come sappiamo dalla matematica, è un sistema basato su una logica, dotato di assiomi (proposizioni assunte come vere) e di regole di inferenza che producono teoremi.

Un'**ontologia** è una teoria: una costruzione ipotetico-deduttiva che propone un modo di organizzare la realtà, nominare le entità, partizionare il continuum del mondo in classi, stabilire relazioni tra esse (di dipendenza, di parte, ecc.).

In informatica, usiamo il termine in senso più debole e colloquiale: indichiamo vocabolari, modelli concettuali, sistemi di classi e proprietà progettati per descrivere un dominio.

La visione del Web Semantico e dei Linked Data

La visione generale del Web Semantico, oggi condivisa dal movimento Linked Open Data è un sistema di conoscenze distribuite, formalizzate, integrate e accessibili attraverso tecnologie ICT.

Le parole chiave sono *distribuzione* e *decentralizzazione*. Il Web, per come è stato concepito ed evoluto, permette a chiunque di creare un proprio nodo: un server, un sito, un contenuto. Il Web Semantico aderisce a questa stessa visione.

In oltre vent'anni di ricerca parte di questa visione si è realizzata. In particolare, si sono sviluppati standard per riferirsi in modo uniforme alle entità del mondo tramite **URI** condivisi. È un livello referenziale, non concettuale, ma essenziale: parlare della Torre Eiffel attraverso un URI universalmente riconosciuto garantisce stabilità nella referenza.

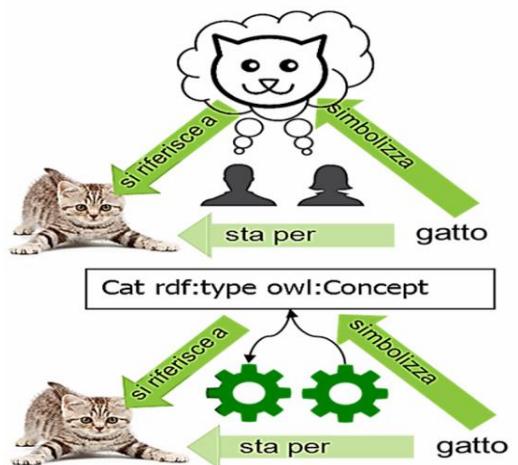
A questo livello si affianca un patrimonio enciclopedico distribuito, nato non dal W3C ma da iniziative come Wikipedia. Wikipedia fornisce nomi, descrizioni testuali, e – tramite DBpedia – dati strutturati estratti dai contenuti encyclopedici. Sebbene Wikipedia sia centralizzata come infrastruttura, distribuisce i propri dati come open data e permette una collaborazione globale.

Accanto a queste realtà, esistono basi di conoscenza centralizzate gestite da singole organizzazioni, che raccolgono, puliscono e forniscono dati in modo uniforme.

Problemi fondamentali: linguaggio, semantica, interpretazione

Nonostante i progressi, restano problemi aperti e profondi, che toccano la natura stessa della semantica.

Il **triangolo semiotico** di Ogden e Richards, già anticipato da Peirce, afferma che il rapporto tra un simbolo e l'oggetto che esso designa non è diretto: è mediato da un **conetto**. Due persone possono usare la parola “gatto” e intendere la stessa categoria concettuale soltanto grazie a processi cognitivi condivisi – complessi, non garantiti. Nei sistemi informatici questo triangolo viene “chiuso” da algoritmi, non da esseri umani. Come garantire allora che due sistemi interpretino allo stesso modo un concetto come “gatto”, soprattutto quando la ricchezza cognitiva umana non è disponibile?



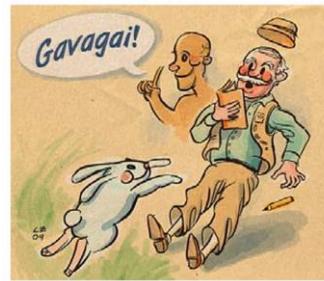
Il linguaggio inoltre è **vago per costruzione**. Predicati come “calvo” hanno casi borderline. La vaghezza non è un difetto: è ciò che permette al linguaggio naturale di funzionare. I sistemi logici tentano di eliminarla, ma nella pratica dei sistemi informativi la vaghezza rientra inevitabilmente attraverso il linguaggio naturale utilizzato per descrivere i dati.

Quine e la relatività ontologica: il problema di Gavagai

Quine introduce il tema della **indeterminatezza dell'interpretazione** e del **relativismo ontologico**.

Il famoso esempio del *Gavagai* illustra

l'impossibilità di garantire che un termine di un linguaggio sia concettualmente equivalente al termine di un altro: l'esploratore che interpreta “Gavagai” come “coniglio” non può sapere se l'aborigeno intendesse “coniglio”, “insieme delle parti proprie del coniglio”, o “apparizione di un coniglio” come evento.



Gavagai: un coniglio? Un insieme delle sue parti? La comparsa di un animale fatto così?

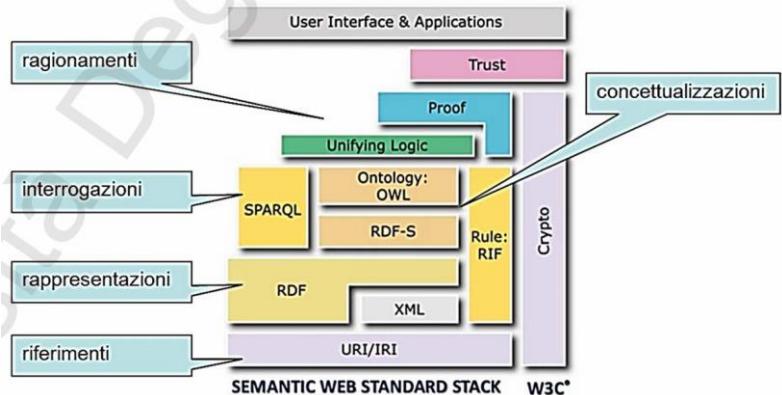
W.O. Quine, Ontological Relativity, 1968
faculty.unlv.edu/rwilburn/Ontological%20Relativity.pdf

Il punto di Quine è radicale: il **commitment ontologico** di un termine è inaccessibile. Possiamo solo formulare assunzioni. Ogni traduzione implica una trasposizione nel nostro sistema concettuale, senza garanzie di equivalenza.

Dov'è la semantica nel Web Semantico?

Nello storico stack del Semantic Web del W3C troviamo livelli per URI, RDF, ontologie, query, regole, ragionamento, trust. Tuttavia, nonostante la struttura stratificata, la “semantica” come tale non ha un blocco dedicato.

La semantica non è un oggetto: è un **processo di interpretazione** che si appoggia su quegli strati, ma non può essere completamente reificato o reso oggettuale. Possiamo però supportare i processi interpretativi riducendo arbitrarietà, incomprensioni e incoerenze: è in questo spazio che operano standard e pratiche del Web Semantico.



Dalla visione del Giant Global Graph alle grandi piattaforme centralizzate

La visione originaria di Berners-Lee era il **Giant Global Graph** (GGG): una base di conoscenza globale emergente dal lavoro di comunità indipendenti. Una realtà distribuita, con concettualizzazioni mediate socialmente e interpretazioni fornite localmente da ciascun attore. Parallelamente si sono affermati sistemi centralizzati che implementano concretamente knowledge graph su larga scala:

- **Il Knowledge Graph di Google**, costruito raccogliendo dati annotati tramite microdata, concettualizzati da schema.org e consolidati in una base unica gestita da Google.
- **L'Open Graph di Facebook**, che aggrega informazioni su profili, preferenze e relazioni sociali degli utenti, includendo dati provenienti da siti esterni tramite API e uno schema propriamente definito.

In questi casi la distribuzione dei dati può anche esistere nelle sorgenti, ma la concettualizzazione e l'interpretazione finale sono centralizzate nelle mani del fornitore della piattaforma.

Problemi aperti e direzioni di ricerca

Alla luce dello scenario attuale, i problemi fondamentali che la ricerca affronta possono essere raggruppati in tre categorie principali, la prima delle quali è il **problema**

dell'interpretazione globale:

come garantire che agenti differenti attribuiscano agli stessi nomi e predicati le medesime estensioni? È, in fondo, il problema di Quine, il problema di Gavagai. Una direzione di ricerca è la costruzione di ontologie ed encyclopedie **cogenti**, cioè strutture concettuali che guidino fortemente l'interpretazione, facendo leva su distinzioni di base difficilmente negabili (ad esempio, la distinzione tra eventi e oggetti). Se aborigeno ed esploratore condividessero almeno queste distinzioni fondamentali, la probabilità di interpretazioni divergenti diminuirebbe.

- Interpretazione globale:
 - Problema: Far sì che ciascun agente assegna a nomi e predicati le medesime estensioni.
 - Ricerca: Sviluppare ontologie ed encyclopedie condivise e «cogenti» (adeguate).
- Consistenza globale:
 - Problema: Comunità diverse possono essere in genuino disaccordo sulle cose, e fornire conoscenze contrastanti.
 - Ricerca: Sistemi di rating e trust; logica epistemica e doxastica.
- Ragionamento globale:
 - Problema: Rispondere in modo efficiente a interrogazioni semantiche su sorgenti indipendenti, eterogenee e distribuite.
 - Ricerca: Studio delle proprietà computazionali delle logiche descrittive; distributed query answering.