

Potenzialità del paradigma ontologico nello sviluppo di applicazioni di eGovernment

GIACOMO BUCCI, VALERIANO SANDRUCCI, ENRICO VICARIO
DIPARTIMENTO SISTEMI E INFORMATICA, UNIVERSITÀ DI FIRENZE

SOMMARIO: 1. MODELLI ONTOLOGICI: viene introdotto il concetto di ontologia nella accezione assunta in Informatica – 2. PARADIGMA ONTOLOGICO: viene descritto un sistema di componenti che rappresentano una base di conoscenza incentrata su un modello ontologico – 3. IL SUPPORTO DELLE TECNOLOGIE DEL SEMANTIC WEB: viene delineato il sistema di linguaggi, protocolli e standard che permettono la realizzazione concreta di una base di conoscenza secondo il paradigma ontologico – 4. ARCHITETTURE SW ONTOLOGICHE: sono descritti i principi di organizzazione di una architettura SW incentrata sul paradigma ontologico e ne sono menzionati i benefici – 5. CONCLUSIONI: sono richiamate le conclusioni del ragionamento svolto.

1. MODELLI ONTOLOGICI

In informatica, l'ontologia è una esplicita specificazione di una concettualizzazione¹, un modello che caratterizza la semantica di un dominio definendone entità e relazioni².

¹ T. R. GRUBER. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993

² N. GUARINO. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human and Computer Studies*, 1995

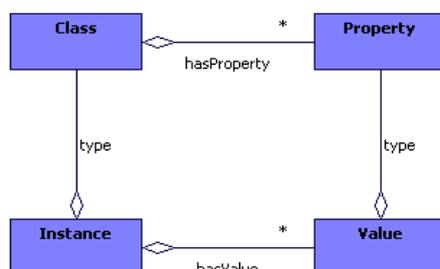


Figura 1 Struttura di un Modello Ontologico

Alla base di ogni modello ontologico sta il concetto di *classe* (Fig.1), la quale rappresenta una categoria, insieme o collezione (e.g. Cittadino, Comune, ma anche Tempo, Spazio). Linguaggi di modellazione differente mettono a disposizione costrutti diversi per definire classi, spesso riconducibili ad una formulazione set-theoretical. Per esempio si può dire che una classe è specializzazione di un'altra (e.g. Comune e Provincia sono sottoclassi di Amministrazione Locale), oppure che una classe è formata come unione, intersezione o complemento di altre classi (e.g. Amministrazione Locale è l'unione di Comune, Provincia, Regione, ...). Il secondo elemento fondamentale di un modello ontologico sono le *proprietà*, che descrivono la struttura interna degli elementi di una classe o la relazione con elementi di altre classi (e.g. gli elementi della classe Cittadino hanno un nome ed hanno un Comune di residenza). Le proprietà hanno un significato intrinseco e vari linguaggi permettono di definirne caratteristiche (e.g. si può dire che essere-amministrato-da-una-Regione una specializzazione di essere-amministrato-da, che essere-incluso-in-una-Provincia ammette una relazione inversa che è includere-un-Comune o che la proprietà essere-confinante è simmetrica). Classi e proprietà rappresentano i tipi di un dominio e le loro relazioni, costituendo la *parte intensionale* di un modello. Su questa si sviluppa la *parte estensionale* che invece definisce gli *individui* che istanziano le classi (e.g. Toscana è una Regione) e la realizzazione concreta dei loro attributi (e.g. Toscana ha un capoluogo che è Firenze).

Esiste una ragionevole analogia tra un modello ontologico ed un modello ad oggetti per il comune uso delle astrazioni di classe, istanza, relazione. Esistono però anche notevoli differenze, che vale la pena

rimarcare³. Una ontologia non include operazioni (metodi) e non è quindi capace di rappresentare in modo diretto ed esplicito algoritmi e processi di calcolo. Un'altra differenza, più profonda anche se forse meno evidente, è nel rapporto tra istanze e classi. Un individuo in un modello ontologico può esistere a prescindere da una classe che lo contenga, e nel corso del tempo può anche cambiare l'insieme delle classi a cui appartiene. E' possibile ad esempio costruire un'istanza, assegnarle degli attributi e sulla base di questi dedurre la classe (o le classi) a cui essa appartiene.

2. IL PARADIGMA ONTOLOGICO

Al di là della sola capacità descrittiva⁴ (potere espressivo), le ontologie traggono molta della loro potenzialità dalla possibilità di trattarle per il tramite di vari componenti di elaborazione⁵. Nel loro insieme (Fig.2), questi delineano un vero e proprio *paradigma ontologico*⁶ per l'organizzazione di un sistema di rappresentazione e elaborazione dell'informazione⁷.

³ S. N. WOODFIELD. The impedance mismatch between conceptual models and implementation environments. *ER'97 Workshop 4 Proceedings*, 1997

⁴ L. ZONG-YONG; W. ZHI-XUE; Y. YING-YING; W. YUE; L. YING. Towards a Multiple Ontology Framework for Requirements Elicitation and Reuse. *Computer Software and Applications Conference*, 2007

⁵ B. SARDER, S. FERREIRA. Developing Systems Engineering Ontologies. *System of Systems Engineering*, 2007

⁶ G. BUCCI, V. SANDRUCCI, E. VICARIO, S. MECCA. An Ontological Sw Architecture For The Development Of Cooperative Web Portals. *ICSoft07*, 2007

⁷ G. BUCCI, V. SANDRUCCI, E. VICARIO. An Ontological SW Architecture Supporting Agile Development of Semantic Portals. *ICSoft 2007 Selected papers*. Springer, 2007

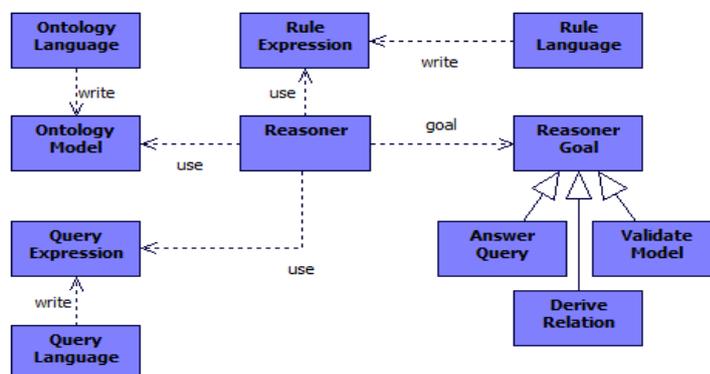


Figura 2 Componenti Architettureali Del Paradigma Ontologico

Molti dei vantaggi legati all'impiego di ontologie deriva dalla possibilità di applicare ad essi un *Reasoner* che è un componente in grado di: risolvere query, eseguire validazioni e inferire nuova conoscenza. Fornendo ad un Reasoner una espressione di query ed un modello ontologico, esso sarà in grado di estrarre la sotto-porzione di modello ontologico che soddisfa i criteri ed i vincoli stabiliti dalla query stessa. Date le caratteristiche dei modelli ontologici ed il potere espressivo dei corrispondenti linguaggi di interrogazione, molto superiore rispetto ad esempio al più convenzionale SQL, si parla di *query semantiche*.

Per mezzo di un Reasoner è anche possibile *validare un modello*, ovvero verificare in modo automatico se l'insieme di asserzioni che esso contiene è consistente. Ad esempio, in due diversi modelli il Comune Cortina potrebbe essere stato dichiarata parte della Provincia di Bolzano e della Provincia di Belluno, il che diventa inconsistente nel momento in cui si aggiunge la regola che un Comune non può appartenere a più Province (ovvero che gli insiemi di Comuni in Province distinte devono essere disgiunti). L'operazione di validazione è particolarmente importante quando ci si trovi ad operare in un ambito distribuito e si vogliono manipolare modelli ottenuti per composizione di sotto-modelli provenienti da sorgenti di informazione sviluppate in maniera concorrente (e.g. modelli di tracciati di informazione sviluppati in maniera concorrente da pubbliche amministrazioni nell'ambito di un sistema di cooperazione applicativa).

Infine, un Reasoner può eseguire ragionamenti su di un modello

ontologico per *dedurre nuove proprietà*. Mentre una convenzionale base di dati relazionale contiene solo i records che vi sono stati esplicitamente inseriti, in una ontologia i dati rappresentati in modo esplicito possono essere espansi attraverso *regole di inferenza* che agiscono su relazioni generali tra classi (e.g. applicando regole di ereditarietà) o su relazioni specifiche che esistono tra le classi del dominio (e.g. se la Regione Veneto include la Provincia Belluno e la Provincia Belluno include il Comune Cortina, allora la regola di transitività sulle relazioni di inclusione implica che Veneto include Cortina).

3. IL SUPPORTO DELLE TECNOLOGIE DEL SEMANTIC WEB

Il paradigma ontologico può avere realizzazione nella costruzione di una base di conoscenza concreta, grazie al supporto di un numero di standards e componenti sviluppati nell'ambito della Semantic Web Iniziative del W3C.

Il Semantic Web è stato concepito come una evoluzione del web attuale nella quale i dati sono associati a una precisa connotazione semantica che ne permetta l'interpretazione per la risoluzione di interrogazioni semantiche e l'elaborazione da parte di agenti intelligenti⁸. Secondo questo disegno, il Semantic Web può essere convenientemente immaginato come una (enorme) collezione di risorse digitali (testi, immagini, filmati) il cui significato (e il significato delle cui relazioni) è descritto attraverso elementi di un sistema di meta-dati semantici (Fig. 3).

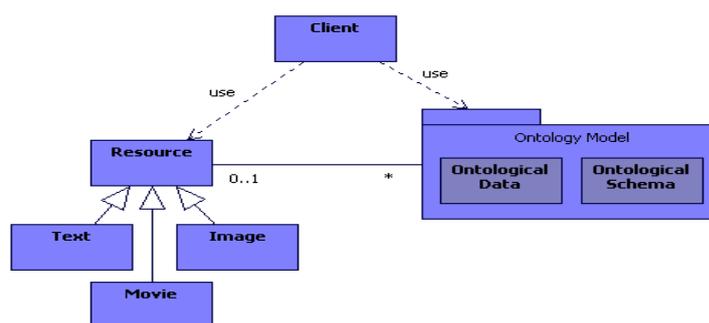


Figura 3 Relazione tra risorse web e relativi meta-dati

⁸ T. BERNERS-LEE. Semantic web roadmap. [http:// www.w3.org/2001/sw/](http://www.w3.org/2001/sw/), 1998

Le responsabilità coinvolte nella realizzazione del Semantic Web possono essere ripartite in un modello stratificato a cui corrisponde una pila di tecnologie e protocolli (Fig.4), in maniera del tutto analoga al modo con cui l'architettura di Internet viene riferita al modello ISO-OSI. In questo schema le ontologie recitano una parte fondamentale, essendo ad esse attribuita la responsabilità di specificare la semantica del sistema dei meta-dati attraverso i quali sono indicizzate le risorse.

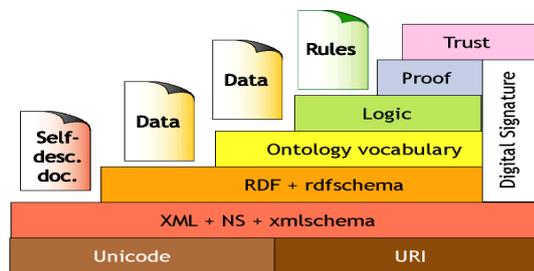


Figura 4 Pila di tecnologie per il Semantic Web

Nonostante un notevole sforzo, e nonostante il successo nella realizzazione di molte componenti del disegno, la concretizzazione del paradigma del Semantic Web è ancora limitata da un numero di complessità inerenti, tra cui due almeno ci appaiono rilevanti ai fini del presente ragionamento. Un primo problema ha natura pratica e tecnologica e riguarda la necessità di associare una caratterizzazione semantica alla immensa mole di informazione che esiste nel Web attuale: tale problema è grande sia qualitativamente che quantitativamente, ed è tanto più difficile quanto meno strutturata è l'informazione; mentre è possibile immaginare meccanismi di mapping per sorgenti strutturate quali i database relazionali, il trattamento di risorse non regolari richiede tecniche di intelligenza artificiale e di elaborazione del linguaggio naturale che difficilmente oggi possono rispondere alla scala del problema. Un secondo problema ha invece natura più concettuale, ma proprio per questo costituisce una preconditione alla soluzione del primo: affinché siano possibili elaborazioni automatiche su dati condivisi occorre accordo non solo sul formato e la sintassi dei meta-dati, ma anche sulla loro semantica⁹; nella pratica questo si traduce nella necessità di

⁹ A. HAJMOOSAEI, S. ABDUL-KAREEM. An Approach for Mapping of Domain-

condividere modelli ontologici, e questo non può che avvenire attraverso il governo di processi di sviluppo iterativi incrementali e partecipati, capaci di rispondere ai requisiti della concorrenza nello sviluppo e della evoluzione dei bisogni.

Come prodotto intermedio, lo sforzo mirato allo sviluppo del Semantic Web ha però prodotto un sistema di linguaggi, protocolli, componenti e standards che permettono di rispondere a requisiti applicativi che aggirano o mitigano le complessità menzionate. In particolare, i prodotti concepiti per la realizzazione del Semantic Web appaiono sufficienti e adeguati alla realizzazione di sistemi informativi organizzati secondo il concetto del paradigma ontologico.

Innanzitutto sono disponibili linguaggi di specifica: per la definizione di modelli ontologici, per la formulazione di espressioni di query, e per la specificazione di regole di inferenza¹⁰. I tre linguaggi possono convenientemente essere OWL¹¹, SPARQL¹² e SWRL¹³. A sua volta, OWL poggia su altri linguaggi di rappresentazione tra cui RDF¹⁴ (Resource Description Framework), XML (Extensible Markup Language), URI e UNICODE (Fig.5). Sono poi disponibili Reasoner, quali ad esempio Pellet, in grado di eseguire le interrogazioni e le inferenze sui modelli ontologici. Nel loro insieme i componenti concreti di Fig.5 realizzano tutti i moduli concettuali dello schema di Fig.4.

based Local Ontologies. *Complex, Intelligent and Software Intensive Systems*, 2008.

¹⁰ C. TEMPICH, H.S. PINTO, S. STAAB. Ontology engineering revisited: an iterative case study with diligent. *The Semantic Web: Research and Applications: 3rd European Semantic Web Conference*, 2006,

¹¹ W3C. Owl web ontology language. <http://www.w3.org/TR/owl-features>, 2004

¹² W3C. Simple Protocol and RDF Query Language. <http://www.w3.org/TR/rdf-sparql-query/>, 2008

¹³ W3C. Semantic Web Rule Language. <http://www.w3.org/Submission/SWRL/>, 2004

¹⁴ W3C. Resource Description Framework. <http://www.w3.org/RDF/>, 2004

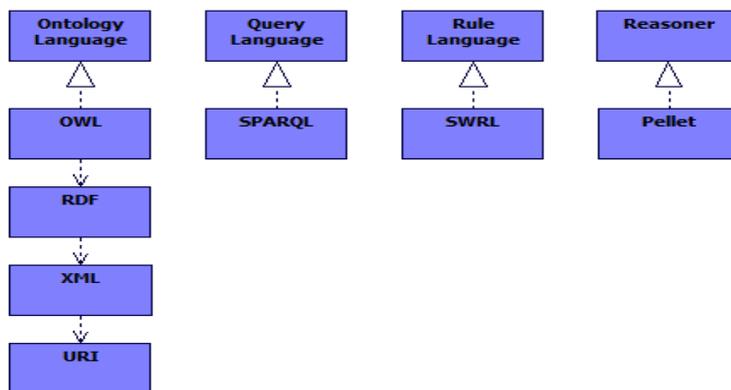


Figura 5 Linguaggi per la Rappresentazione dell'Informazione nel Paradigma

4. ARCHITETTURE SW ONTOLOGICHE

Come già menzionato, un modello ontologico può rappresentare l'informazione di un dominio ma non può codificarne gli algoritmi ed i processi di calcolo. Combinando la capacità di rappresentazione e classificazione delle ontologie con la capacità di elaborazione di un linguaggio ad oggetti è però possibile superare il limite e realizzare quello che chiamiamo una *architettura SW ontologica*, la quale trae dall'uso delle ontologie un numero di pregevoli caratteristiche.

In generale, un sistema SW complesso è organizzato secondo una architettura che distribuisce le responsabilità in tre strati: il domain layer codifica la logica applicativa del sistema ed è responsabile di implementarne le elaborazioni; il presentation layer espone una o più interfacce sul dominio; il data layer è responsabile di dare persistenza ai dati su cui opera la logica di dominio. Ad esempio, nella organizzazione di un Centro Unificato di Prenotazione: il presentation layer espone interfacce verso gli utenti e gli operatori del servizio sanitario; il data layer compone (federa) uno o più archivi nei quali sono mantenuti i dati delle strutture sanitarie coinvolte; il domain layer ha la responsabilità di gestire le operazioni di prenotazione delle prestazioni.

In una architettura convenzionale, la logica applicativa del domain layer è realizzata con oggetti di un linguaggio di programmazione (e.g. Plain Old Java Objects – POJO). Questi implementano un modello di

specifica, che è usualmente progettato in modo semi-formale tramite i class-diagrams di UML¹⁵, e forniscono una soluzione capace di rappresentare in modo efficace logiche di dominio anche di elevata complessità. Il data layer archivia l'informazione persistente in uno o più databases relazionali, spesso preesistenti all'applicazione e soggetti a vincoli di legacy. La distanza concettuale che esiste tra le astrazioni degli oggetti e delle relazioni (noto in letteratura come object relational impedence mismatch¹⁶) è risolta attraverso uno strato di Object Relational Mapping. In questo schema, che comunque rappresenta una pratica evoluta nello stato dell'arte, la logica di dominio è abilitata a manipolare i records del database ma non la struttura delle tabelle che lo compongono. Ad esempio, la logica di dominio può interrogare il data layer per sapere che Firenze è il capoluogo della Toscana, ma non farlo per sapere che una Regione ha un Capoluogo. In sostanza, il data layer espone la parte estensionale della base di conoscenza, mentre la rappresentazione della parte intensionale rimane a carico del domain layer. Il limite può essere mitigato tramite schemi architetturali avanzati, quali quello del pattern Reflection¹⁷, i quali comunque comportano la necessità di fornire una implementazione ad oggetti del modello concettuale dei dati.

¹⁵ M. FOWLER. *Patterns of Enterprise Application Architecture*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002

¹⁶ S. AMBLER. *Agile Database Techniques: Effective Strategies for the Agile Software Developer*. Wiley, 2003

¹⁷ D. C. SCHMIDT, H. ROHNERT, M. STAL, D. SCHULTZ. *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. Wiley. 2000

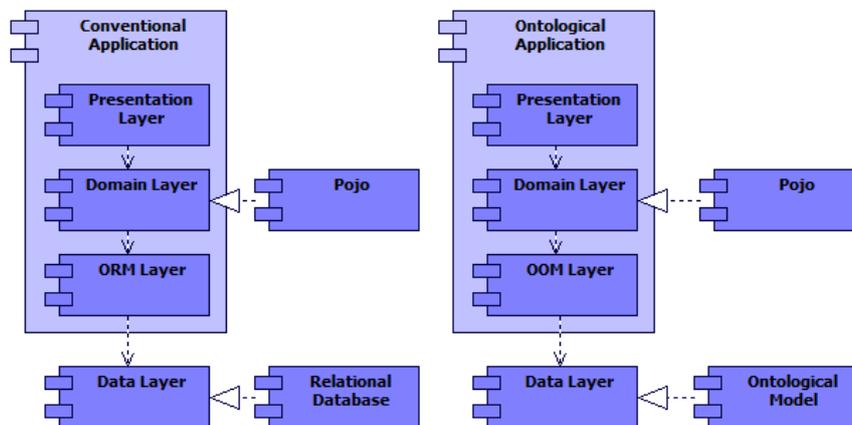


Figura 6 Architetture SW evolute che usano RDB o Ontologie

In una architettura ontologica, il domain layer è ancora realizzato per il tramite di oggetti, ma questi accedono al data layer attraverso un'interfaccia che espone la astrazione di una ontologia. La logica di dominio può quindi delegare al data layer non solo la rappresentazione dei dati ma anche quella delle loro concettualizzazioni, costituiti in questo caso rispettivamente dalle istanze e dalle classi del modello ontologico. Così facendo, diventa possibile generalizzare la logica di dominio in modo da adattarla non solo al variare delle istanze ma anche al variare dei concetti rispetto ai quali esse sono astratte.

Questo fornisce una leva formidabile per affrontare i requisiti di manutenibilità e evolvibilità che segnano qualsiasi applicazione SW. In particolare, questo assume una straordinaria importanza nella gestione di sistemi di cooperazione applicativa, che per loro natura evolvono nel tempo e sono soggetti a sviluppo incrementale, distribuito e spesso concorrente.

5. CONCLUSIONI

L'*ontologia* fornisce una potente astrazione per la organizzazione concettuale di una base di conoscenza combinando dati e concetti in un modello omogeneo, sufficientemente espressivo per la rappresentazione di domini complessi, e sufficientemente formalizzato per l'applicazione di tecniche di inferenza e ragionamento automatico. Questo delinea un

paradigma ontologico nella rappresentazione dell'informazione che può essere concretamente declinato costruendo sul sistema di linguaggi, protocolli e strumenti sviluppati nel disegno del Web Semantico. L'accesso a un tale sistema di rappresentazione tramite linguaggi di programmazione ad oggetti abilita la creazione di *architetture SW ontologiche* nelle quali la logica applicativa si basa su una astrazione ontologica dell'informazione. In questa architettura la logica applicativa può delegare allo strato di persistenza non solo i dati ma anche le loro concettualizzazioni, fornendo una leva straordinaria per raggiungere requisiti che hanno in generale una maggiore rilevanza e che diventano cruciali nello sviluppo di applicazioni di cooperazione applicativa: la agilità nello sviluppo incrementale e nell'evoluzione; la federazione di basi di conoscenza concreta e concettuale distribuite; la capacità di conciliare differenze e mantenere consistenza nella composizione; l'innesto in un contesto tecnologico di crescente maturità.