

**Università degli Studi di Camerino**

Corso di laurea triennale

Informatica per la comunicazione digitale

Anno Accademico 2023/24

*ST1414 Modellazione e Gestione della Conoscenza*

**Configuratore di Laptop**

Progettazione Ontologica e Implementazione Software

Fabio Grelloni

Sommario

[Ambito applicativo e del dominio 3](#_Toc177295029)

[Concetti e proprietà modellate nell’ontologia 3](#_Toc177295030)

[Scenari di utilizzo e query associate 6](#_Toc177295031)

[Responsabilità nello sviluppo dell’applicazione 7](#_Toc177295032)

[Interfacce e classi utilizzate 7](#_Toc177295033)

[Uso dell’applicazione nel dominio considerato 8](#_Toc177295034)

# Ambito applicativo e del dominio

L’idea di progettare un **configuratore di laptop** è nata con l'obiettivo di offrire agli utenti la possibilità di personalizzare il proprio portatile, consentendo loro di selezionare vari componenti, periferiche e accessori sulla base delle proprie esigenze.

Il dominio di riferimento è quello della **tecnologia informatica** e in particolare la **configurazione di hardware per laptop**.

Attraverso un’interfaccia interattiva, l'utente può scegliere tra vari elementi disponibili, come componenti (RAM, CPU etc…), periferiche (dispositivi esterni etc…) e molti altri.

Il configuratore offre un’esperienza semplificata che sfrutta un’ontologia sottostante per gestire e validare le scelte effettuate dagli utenti.

Il dominio tecnologico è particolarmente adatto all’uso di un'ontologia in quanto i componenti di un laptop possono essere rappresentati in maniera strutturata, con relazioni e proprietà ben definite, permettendo anche di garantire la validità delle configurazioni finali.

# Concetti e proprietà modellate nell’ontologia

Nell’ontologia sviluppata per il configuratore, i concetti principali sono i seguenti:

* **Laptop**: rappresenta il dispositivo principale e aggrega tutti i componenti e le periferiche.
* **Componenti**: include CPU, RAM, Storage, Display, Scheda Grafica, e Sistema Operativo. Questi sono rappresentati come classi ontologiche con proprietà quali nome, velocità (per la CPU), capacità (per la RAM e Storage), risoluzione (per il display), ecc.
* **Periferiche**: comprendono dispositivi esterni come Monitor Esterni, Tastiere, Mouse, Speaker Esterni, e Webcam, ciascuno modellato con proprietà specifiche.
* **Porte**: Ethernet, USB e HDMI, che collegano il laptop alle periferiche esterne.
* **Sicurezza**: rappresenta i concetti di Antivirus e Funzionalità di Protezione.
* **Garanzia**: una classe che rappresenta i termini di garanzia associati al laptop.
* **Batteria**: rappresenta l’unità in funzione quando il laptop non è alimentato a corrente.
* **Sistema di raffreddamento**: si occupa di gestire la temperatura del laptop in modo da preservare i vari componenti.
* **Sistema audio**: indica il sistema di out-put del laptop, che comprende tutti i componenti quali speakers, sound cards etc…
* **Colore**: corrisponde alla colorazione del dispositivo.

Ogni concetto nell'ontologia è modellato attraverso l'uso di **data property** e **object property**. Le **data property**, come *hasSpeed*, *hasCapacity*, e *hasResolution*, sono utilizzate per rappresentare le caratteristiche chiave dei vari componenti, descrivendo attributi numerici o testuali come la velocità di elaborazione, la capacità di memoria o la risoluzione del display.

Le **object property**, invece, sono impiegate per definire le relazioni tra i concetti ontologici. Ad esempio, la relazione *hasComponent* permette di collegare un'istanza di *Laptop* a una *CPU*, mentre *hasPeripheral* collega un *Laptop* a un *ExternalMonitor*.

Queste relazioni semantiche, che interconnettono concetti tra loro, garantiscono una rappresentazione coerente e strutturata del dominio.

Grazie all'uso combinato di data property e object property, l'ontologia supporta la gestione delle configurazioni in modo semantico, garantendo non solo la descrizione accurata dei singoli componenti, ma anche la validità delle configurazioni finali attraverso le relazioni tra di essi.

Si integra qui di seguito l’immagine che rappresenta lo schema ontologico del mio configuratore di Laptop.

Lo schema è stato generato tramite il plug-in OWLviz all’interno dell’ambiente di sviluppo Protégé e mostra concetti e relazioni dell’ontologia descritti più dettagliatamente nel file del progetto *LaptopConfigModellazione.rdf* :

Immagine che contiene testo, linea, diagramma

Descrizione generata automaticamente

# 

# Scenari di utilizzo e query associate

Gli scenari di utilizzo principali dell’applicazione includono:

* **Configurazione di un nuovo laptop**: L'utente può selezionare componenti hardware per assemblare un laptop secondo le proprie esigenze.
* **Validazione della configurazione**: L'applicazione può verificare che una configurazione sia valida, ad esempio assicurandosi che ci sia compatibilità tra i componenti o che tutte le periferiche scelte siano supportate dal sistema.
* **Visualizzazione della configurazione finale**: Una volta completate tutte le selezioni, l'utente può visualizzare un riepilogo della configurazione del laptop attraverso un interfaccia ordinata ed intuibile.

Esempi di query SPARQL per il recupero di informazioni dall'ontologia:

* **Query per ottenere tutte le CPU disponibili:**

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX laptop: <http://www.semanticweb.org/fabio/ontologies/2024/7/LaptopConfigModellazione#>

SELECT ?cpu ?speed WHERE {

?cpu rdf:type laptop:CPU .

?cpu laptop:hasCPUSpeed ?speed .

}

* **Query per ottenere le periferiche associate a un determinato laptop:**

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX laptop:<http://www.semanticweb.org/fabio/ontologies/2024/7/LaptopConfigModellazione#>

SELECT ?peripheralName WHERE {

?laptop rdf:type laptop:Laptop ;

laptop:hasPeripheral ?peripheral .

?peripheral laptop:hasPeripheralName ?peripheralName .

}

# Responsabilità nello sviluppo dell’applicazione

L'applicazione è stata sviluppata suddividendo il lavoro in diverse responsabilità:

* **Caricamento e gestione dell'ontologia**: Il componente OntologyLoader si occupa di caricare il file RDF/OWL contenente i dati dell'ontologia e di fornire un'interfaccia per accedere a questi dati tramite query.
* **Inferenza con Pellet**: Grazie a Pellet, l'applicazione può eseguire inferenze logiche, ad esempio per determinare automaticamente le configurazioni valide sulla base delle regole ontologiche.
* **Esecuzione di query SPARQL**: Il componente SPARQLQueryExecutor si occupa di gestire l'interazione tra l'ontologia e l’applicazione, eseguendo query SPARQL per recuperare dati specifici.
* **Configurazione e visualizzazione**: L'interfaccia utente è gestita tramite JavaFX, offrendo un'interazione intuitiva che permette di configurare il laptop attraverso scelte interattive e visualizzare la configurazione finale.

# Interfacce e classi utilizzate

L'applicazione è strutturata utilizzando diverse classi che gestiscono le principali responsabilità:

* **LaptopConfiguratorApp**: La classe principale che gestisce l'interfaccia utente, permette all'utente di selezionare i componenti e visualizzare la configurazione finale.
* **OntologyLoader**: Questa classe si occupa di caricare l'ontologia da un file RDF/OWL e di inizializzare il modello Jena. Fornisce il modello di ontologia utilizzato dall'intera applicazione.
* **PelletInference**: Implementa l'inferenza tramite il reasoner Pellet per determinare relazioni e configurazioni valide all'interno dell'ontologia.
* **SPARQLQueryExecutor**: Questa classe è responsabile dell'esecuzione di query SPARQL sull'ontologia, permettendo il recupero delle informazioni necessarie alla configurazione del laptop.
* **LaptopComponentQueryService**: Un servizio che agisce come un’interfaccia tra l’applicazione e l’ontologia, incapsulando le query SPARQL e fornendo metodi specifici per il recupero di componenti (CPU, RAM, Periferiche, ecc.).

# Uso dell’applicazione nel dominio considerato

L'applicazione **Configuratore di Laptop** viene utilizzata principalmente in contesti dove è richiesta una configurazione personalizzata di un laptop, ad esempio in negozi di informatica, servizi di vendita online o anche in ambito educativo, per simulare configurazioni hardware.

Il processo di utilizzo è semplice e guidato:

1. **Selezione del componente**: L'utente sceglie tra le varie opzioni disponibili (ad esempio, Batteria, Colore, Sistema Audio, Periferiche, ecc.).
2. **Validazione della configurazione**: L'applicazione, attraverso l'ontologia e l’inferenza, verifica la compatibilità e la validità delle scelte.
3. **Visualizzazione finale**: Una volta completate tutte le selezioni, l'utente può visualizzare la configurazione completa del laptop, con tutti i dettagli sugli elementi scelti.

L'ontologia sottostante garantisce che i componenti siano compatibili tra loro e che la configurazione sia valida, offrendo una base solida per estendere l'applicazione a nuovi componenti o periferiche in futuro.

L'ontologia sottostante assicura che i componenti siano compatibili tra loro e che la configurazione risultante sia valida, fornendo una struttura flessibile e scalabile.

Grazie alla natura modulare dell'ontologia, è possibile **estendere** facilmente l'applicazione per includere nuovi componenti, periferiche o accessori in futuro, semplicemente aggiungendo nuovi concetti, data property e object property. Ad esempio, si potrebbe integrare ulteriori concetti come schede grafiche di ultima generazione, nuovi standard di porte di connessione, o tecnologie avanzate di raffreddamento.

Oltre a garantire la coerenza semantica delle configurazioni, l'ontologia potrebbe essere ulteriormente ampliata per supportare un **sistema di preventivazione automatica**. Attraverso l'aggiunta di data property come hasPrice, l'applicazione potrebbe calcolare in tempo reale il costo complessivo del laptop in base ai componenti selezionati, fornendo un preventivo preciso e dettagliato.

Questo approccio semantico renderebbe il configuratore non solo uno strumento di personalizzazione, ma anche una piattaforma integrata per la gestione delle scelte economiche e tecniche, in grado di evolversi dinamicamente insieme al mercato dei laptop.