



## **Università degli Studi di Camerino**

informatica per la comunicazione digitale

*ST1414 Modellazione e Gestione della Conoscenza*

Pigliacampo Leonardo, matricola 119405

### **Configuratore di Laptop**

*Progettazione di un'ontologia e implementazione Software*

#### **Descrizione dell'ambito applicativo e del dominio**

Il progetto sviluppato mira a trasformare il processo di personalizzazione di un laptop in un'esperienza accessibile e intuitiva, sfruttando una piattaforma basata su Java con interfaccia grafica.

L'idea alla base del configuratore è di permettere agli utenti, sia essi consumatori finali o professionisti IT, di assemblare un dispositivo che rispecchi precisamente le loro esigenze di uso quotidiano, professionale o di intrattenimento. Gli utenti possono selezionare e combinare vari componenti, visualizzando in tempo reale le modifiche ottimizzando così le performance del dispositivo finale in base alle specifiche desiderate.

La configurazione avviene tramite un'ontologia che modella il dominio tecnico dei laptop, definendo una struttura dati che rappresenta non solo i componenti singoli, ma anche le loro interdipendenze e compatibilità.

La tecnologia dei laptop, con i suoi componenti complessi e interconnessi, si presta particolarmente all'applicazione di un'ontologia che organizza e definisce queste parti

in modo chiaro e strutturato, contribuendo a verificare la coerenza e la funzionalità delle configurazioni finali scelte dagli utenti.

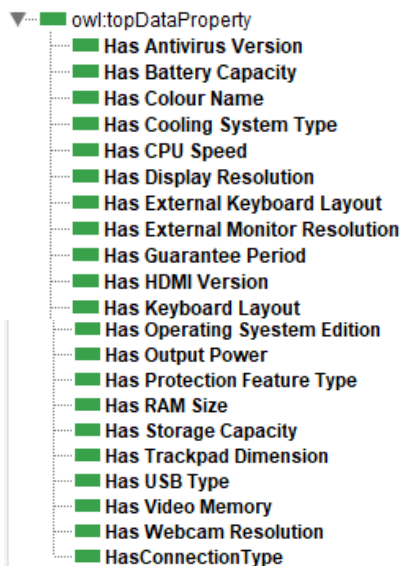
## Concetti e proprietà modellate nell'ontologia

Nell'ontologia i concetti principali sono:

- **Laptop:** Rappresenta l'entità principale, il prodotto finale che sarà configurato dall'utente.
- **Componenti Hardware:** include CPU, Display, Scheda Grafica, Keyboard, RAM, Storage, trackpad e Webcam. Questi componenti sono modellati come classi ontologiche, ciascuna con proprietà specifiche, come il nome, la velocità per la CPU, la capacità per la RAM e lo storage, la risoluzione per il display e webcam, il layout per la tastiera e le dimensioni per il trackpad.
- **Accessori:** include dispositivi esterni quali mouse, tastiere e monitor, ognuno dei quali è modellato con proprietà specifiche come, ad esempio, il tipo di connessione, il layout per le tastiere e la risoluzione per il monitor esterno.
- **Porte:** Questa classe ontologica rappresenta le porte (USB e HDMI), tramite le quali vengono collegati gli accessori come monitor esterni ecc..al laptop.
- **Sicurezza:** comprende gli elementi di Antivirus e Funzionalità di Protezione come Scanner di impronte digitali, riconoscimento facciale e otturatore webcam.
- **Garanzia:** Questa classe ontologica rappresenta i termini di garanzia associati al laptop.
- **Sistema operativo:** Classe che rappresenta il sistema operativo del laptop comprende proprietà come l'edizione.
- **Sistema di raffreddamento:** Classe che rappresenta i meccanismi e le tecnologie di raffreddamento del laptop, come ventole e sistemi di raffreddamento a liquido.
- **Batteria:** Classe ontologica che modella le caratteristiche della batteria del laptop, come la capacità misurata in mAh.
- **Sistema audio:** Classe che rappresenta il sistema di output sonoro del laptop comprende proprietà come la potenza di output del sistema.
- **Colore:** corrisponde alla colorazione della scocca del dispositivo.

Nell'ontologia, i concetti sono delineati tramite l'uso di *data property* e *object property*.

**Le data property** specificano attributi quantitativi o descrittivi dei componenti, quali "Has CPU Speed" per la velocità di elaborazione, "Has RAM Size" per la capacità di memoria, "Has Display Risoluzione" per la risoluzione dello schermo, "Has Battery Capacità" per la capacità della batteria, e "Has Antivirus Version" per la versione dell'antivirus. Queste proprietà consentono di attribuire valori numerici o testi che dettagliano le caratteristiche essenziali dei componenti del laptop.



Le **Object Property** nell'ontologia collegano i vari componenti e caratteristiche del laptop, stabilendo relazioni tra di loro.

Ad esempio, "Has Accessory" collega un laptop ai suoi accessori, "Has Colour" specifica il colore del laptop, mentre "Is Hardware Component Of" indica che un elemento specifico è parte dell'hardware di un laptop. Queste object property permettono di definire le connessioni strutturali e funzionali all'interno dell'ontologia, facilitando l'organizzazione delle informazioni in modo logico e interconnesso.



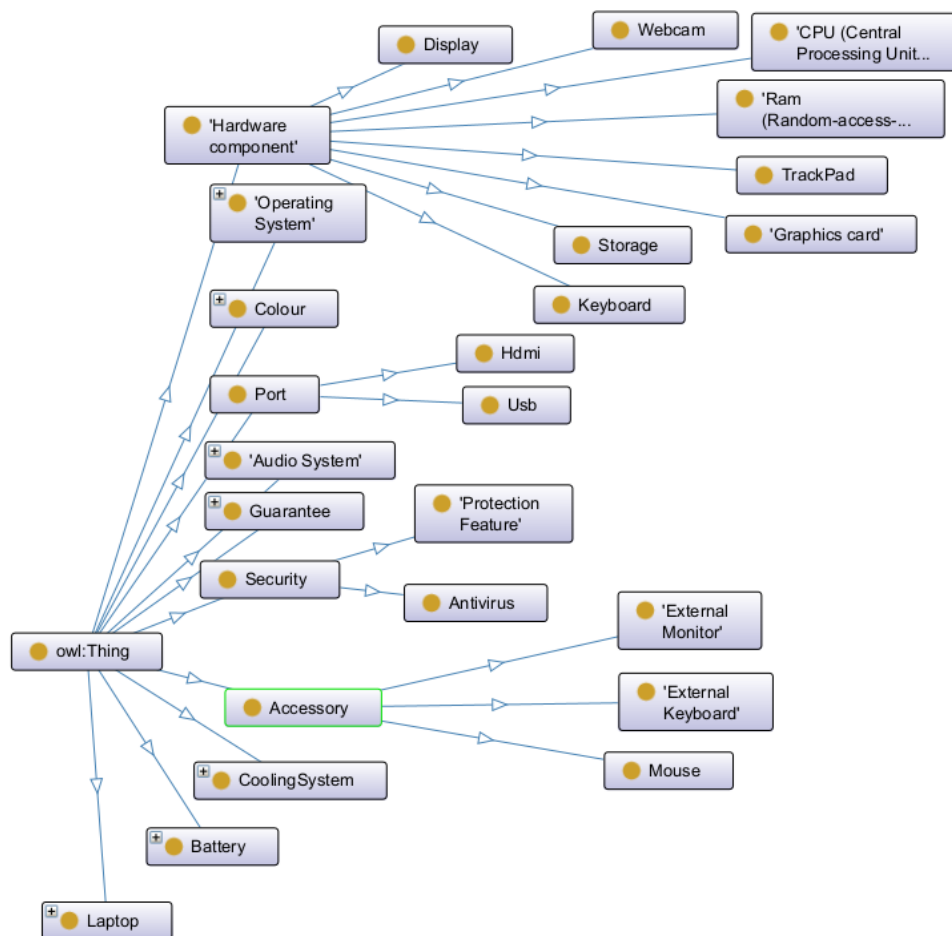
Queste relazioni semantiche collegano i concetti in modo che riflettano una struttura coerente e dettagliata del dominio. Utilizzando sia le data property che le object property, l'ontologia fornisce supporto semantico per la gestione delle configurazioni. Ciò permette non solo una descrizione precisa dei componenti individuali, ma anche l'assicurazione che le configurazioni finali siano valide attraverso le interconnessioni tra i vari elementi.

Si integra qui di seguito l'immagine che rappresenta lo schema ontologico del LaptopConfigurator, generato tramite il plug-in OntoGraf all'interno dell'ambiente di sviluppo Protégé.

L'ontologia si può esaminare dettagliatamente nel file:

*LaptopConfiguratorModellazioneGestioneConoscenza.rdf* che si trova nel seguente path:

PigliacampoLeonardo119405\LaptopConfigurator\app\src\main\resources



## Scenari di utilizzo

I principali scenari di utilizzo dell'applicazione comprendono:

- **Configurazione di un nuovo laptop:** L'utente ha la possibilità di scegliere tra diversi componenti per assemblare un laptop che soddisfi le proprie necessità.
- **Visualizzazione della configurazione finale:** Dopo aver effettuato tutte le selezioni, l'utente può visualizzare un riepilogo dettagliato della configurazione del laptop attraverso un'interfaccia chiara e intuitiva.

## Query SPARQL

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

### Sistemi operativi disponibili con le relative edizioni:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX laptop: <http://www.semanticweb.org/leona/ontologies/2024/9/LaptopConfiguratorModellazioneGestioneConoscenza#>
SELECT ?operatingSystem ?edition
WHERE {
  ?operatingSystem rdf:type laptop:OperatingSystem .
  ?operatingSystem laptop:HasOperatingSystemEdition ?edition .
}
```

### CPU con la relativa velocità (CPUSpeed):

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX laptop:
<http://www.semanticweb.org/leona/ontologies/2024/9/LaptopConfiguratorModellazioneGestioneConoscenza#>
SELECT ?cpu ?speed
WHERE {
  ?cpu rdf:type laptop:Cpu .
  ?cpu laptop:HasCPUSpeed ?speed .
}
```

### Sistemi di raffreddamento con la relativa tipologia:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX laptop: <http://www.semanticweb.org/leona/ontologies/2024/9/LaptopConfiguratorModellazioneGestioneConoscenza#>
SELECT ?coolingSystem ?type
WHERE {
  ?coolingSystem rdf:type laptop:CoolingSystem .
  ?coolingSystem laptop:HasCoolingSystemType ?type .
}
```

Altre query SPARQL si trovano all'interno del QueryService

## Responsabilità considerate nello sviluppo dell'applicazione

- **Caricamento dell'ontologia:** Assicurare che la struttura di conoscenza relativa ai componenti e alle relazioni del dominio sia disponibile e accessibile per l'applicazione, facilitando l'utilizzo e la consultazione dei dati ontologici.
- **Inferenza ontologica:** Applicare deduzioni logiche sui dati dell'ontologia per arricchire la conoscenza disponibile, abilitando nuove correlazioni e garantendo la coerenza delle informazioni attraverso operazioni di inferenza.
- **Esecuzione di query SPARQL:** Consentire l'accesso a informazioni specifiche dell'ontologia tramite l'esecuzione di query SPARQL, permettendo di estrarre dati e relazioni in modo mirato e dinamico, in base alle esigenze dell'applicazione.
- **Configurazione e visualizzazione:** Fornire un'interfaccia utente intuitiva che permetta agli utenti di configurare i componenti del laptop tramite selezioni guidate e di visualizzare chiaramente la configurazione finale.

## Interfacce e classi utilizzate

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

- **LaptopConfiguratorModellazioneGestioneConoscenzaApp:**

La classe principale che gestisce l'interfaccia utente, implementata con JavaFX, fornisce un'interazione intuitiva, consentendo agli utenti di configurare il laptop mediante selezioni interattive e di visualizzare la configurazione finale in modo dettagliato e dinamico.

- **OntologyLoader:**

La classe OntologyLoader è incaricato dell'importazione del file RDF, utilizzando il framework Jena per caricare e inizializzare la struttura ontologica nel sistema. Fornisce il modello di ontologia utilizzato dall'intera applicazione.

- **PelletInference:**

La classe PelletInference si occupa di eseguire operazioni di inferenza sull'ontologia caricata, utilizzando il reasoner Pellet. La sua responsabilità principale è applicare regole logiche e deduzioni automatiche sui dati ontologici per arricchire la conoscenza estraibile. Questo consente di derivare nuove informazioni implicite dai dati esistenti, supportando query avanzate e garantendo la coerenza delle configurazioni.

- **SPARQLQueryExecutor:**

La classe SPARQLQueryExecutor si occupa di gestire l'interazione tra ontologia e applicazione, eseguendo query SPARQL per recuperare dati.

- **QueryService:**

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, Accessory, ecc.).

## **Applicazione del Configuratore di Laptop nel Settore**

Il "Laptop Configurator" è pensato per ambienti che richiedono configurazioni personalizzate, come negozi di elettronica o piattaforme di e-commerce. L'applicazione guida l'utente attraverso un percorso chiaro e interattivo:

1. **Scelta dei Componenti:** L'utente ha a disposizione una selezione di elementi personalizzabili (batteria, colore, sistema audio, periferiche, ecc.) per adattare il laptop alle proprie necessità.
2. **Anteprima della Configurazione:** Al termine delle selezioni, viene mostrata una visualizzazione completa della configurazione scelta, evidenziando ogni componente e le specifiche rilevanti.

L'ontologia alla base garantisce la compatibilità tra i vari componenti, assicurando la validità delle configurazioni e la loro flessibilità, consentendo di ampliare facilmente l'applicazione con l'integrazione di nuovi componenti. La natura modulare dell'ontologia permette un'espansione dinamica: basta aggiungere nuovi elementi e proprietà.

Oltre a garantire una configurazione accurata, l'ontologia può essere ampliata per includere un sistema di calcolo del prezzo. Aggiungendo questa semplice proprietà, l'applicazione potrebbe calcolare il prezzo complessivo della configurazione scelta, consentendo di visualizzare preventivi aggiornati e personalizzati. Questo approccio trasforma il configuratore in uno strumento completo per gestire sia scelte tecniche che valutazioni economiche, rendendolo pronto ad adattarsi alle nuove esigenze e ai trend del mercato laptop.