



# DS-DIAGNOSIS SUPPORT

A MEDICAL SUPPORT FOR DISEASE DIAGNOSIS

---

*Icon caso di studio  
mat. 655602 Poli Giovanni Federico*

# DS-DIAGNOSIS SUPPORT

---

**DS-Diagnosis Support** è un'applicazione con interfaccia utente utile ad individuare possibili malattie a partire dai sintomi che esse presentano.

L'applicazione acquisisce conoscenza dall'ontologia DO: Disease Ontology.

Un'ontologia standardizzata allo scopo di fornire alla comunità biomedica descrizioni coerenti, riutilizzabili e sostenibili dei termini delle malattie umane e dei relativi concetti di malattie del vocabolario medico.

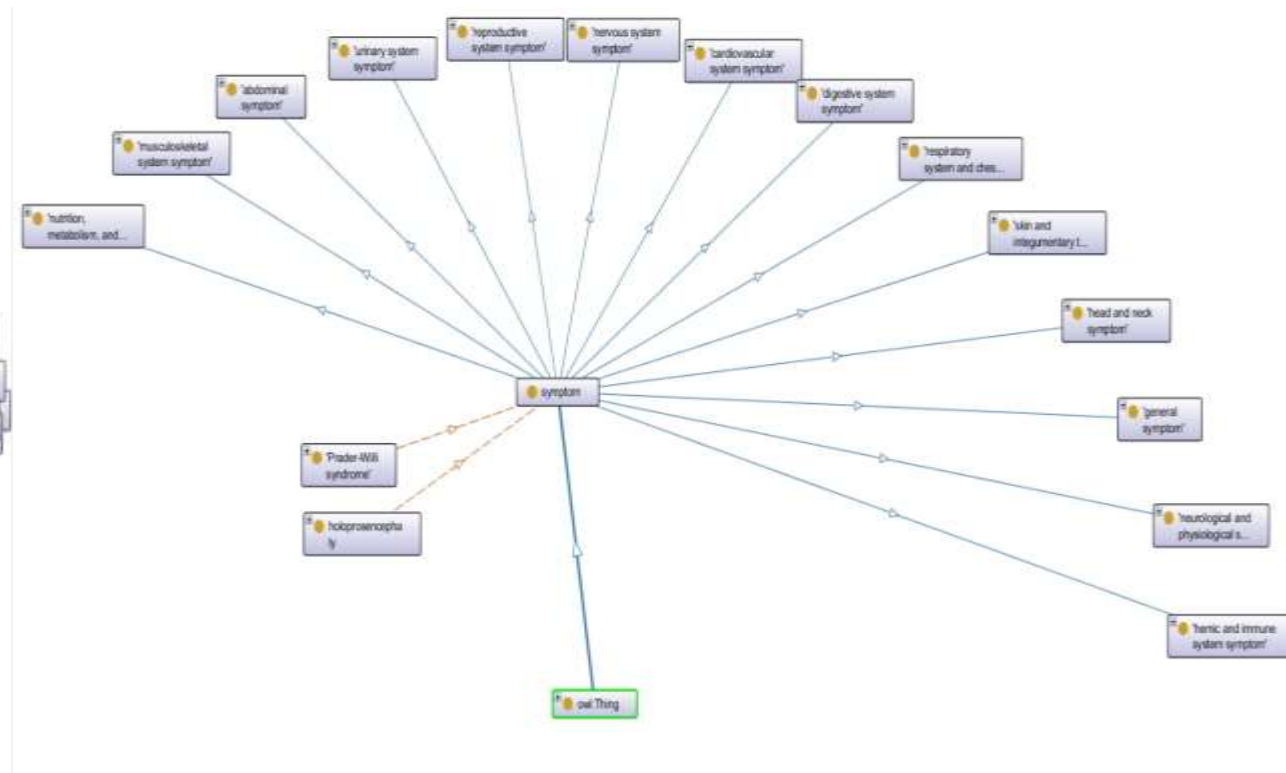
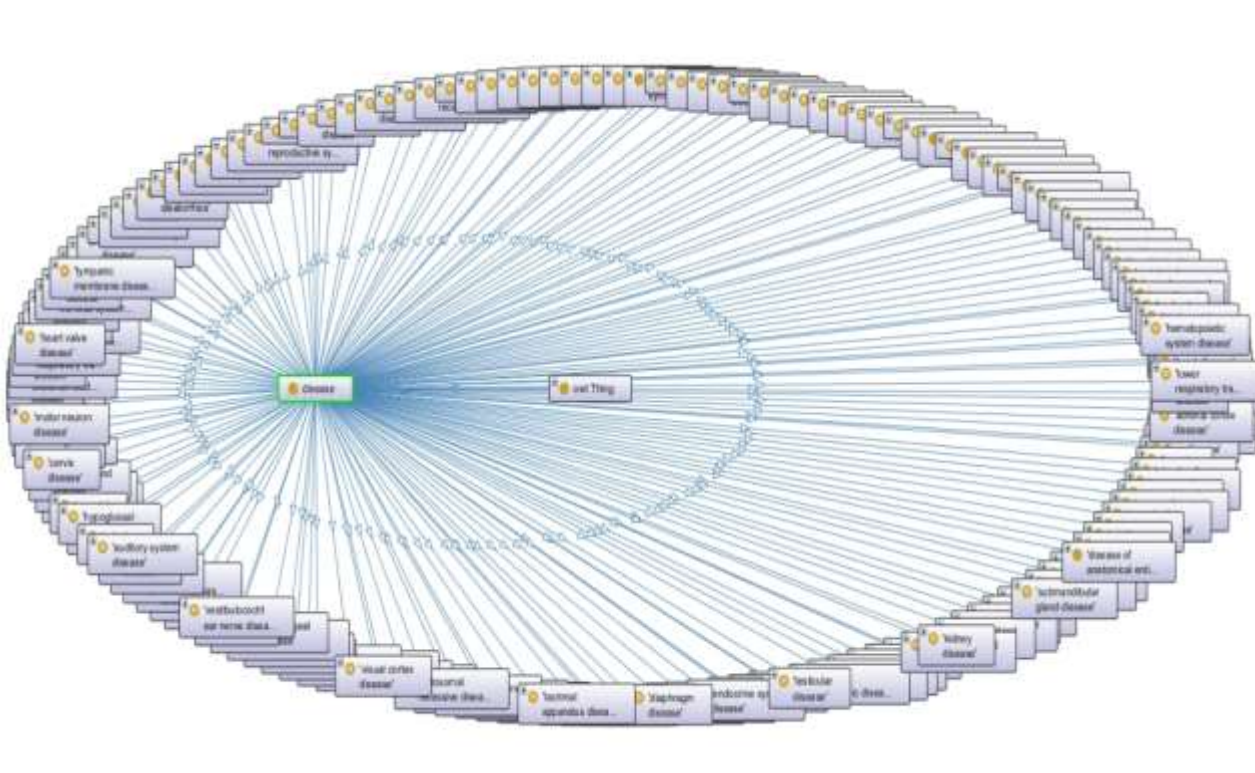
Il modello alla base è un grafo orientato all'interno del quale:

- i **nodi** sono costituiti da malattie e sintomi
- gli **archi** escono da un nodo malattia e convergono su un nodo sintomo

# Diagramma delle classi

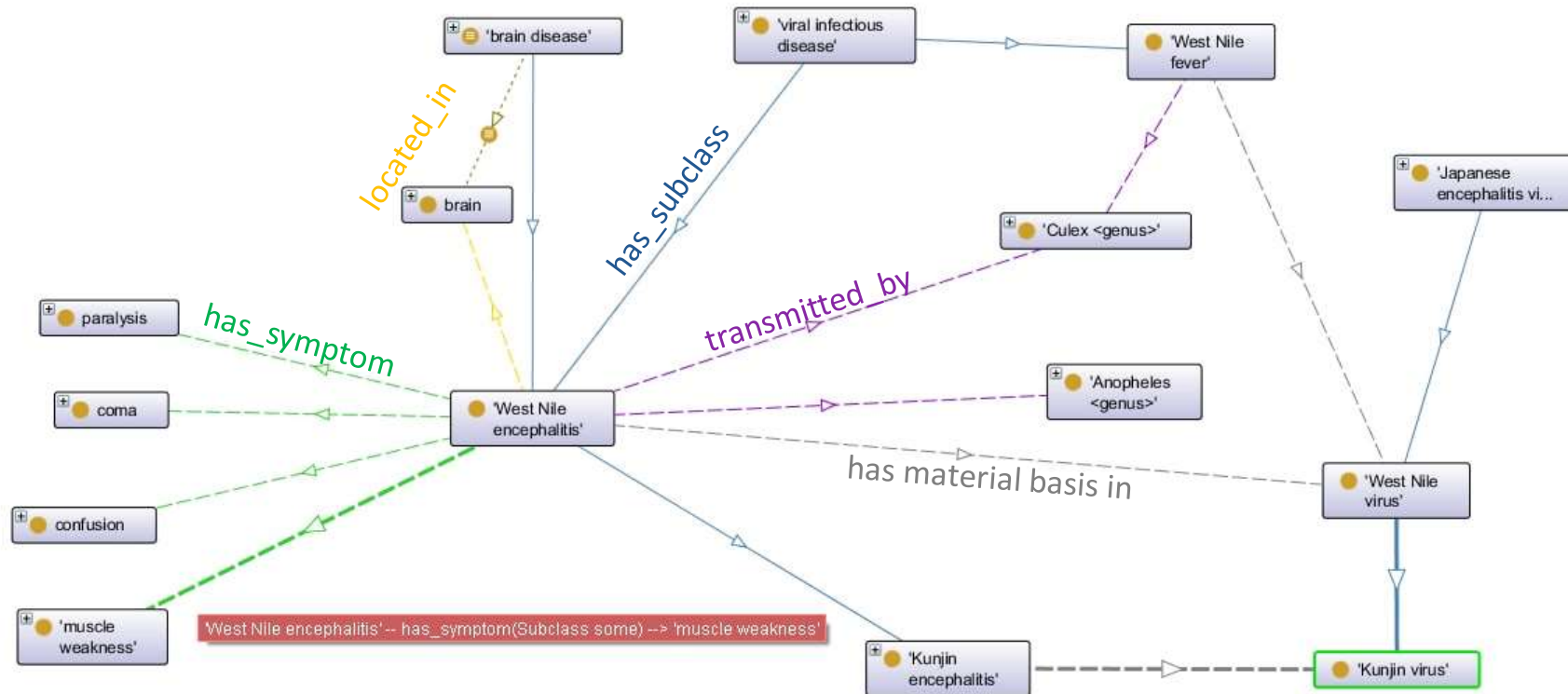
La Knowledge-Base generata prende in considerazione due classi.

La classe *Disease* e la classe *Symptom*, legate attraverso la proprietà *has\_symptom*





# Esempio della relazione *has\_symptom* (in verde) per la malattia «*West Nile encephalitis*»



# Qual è lo scopo di DS?

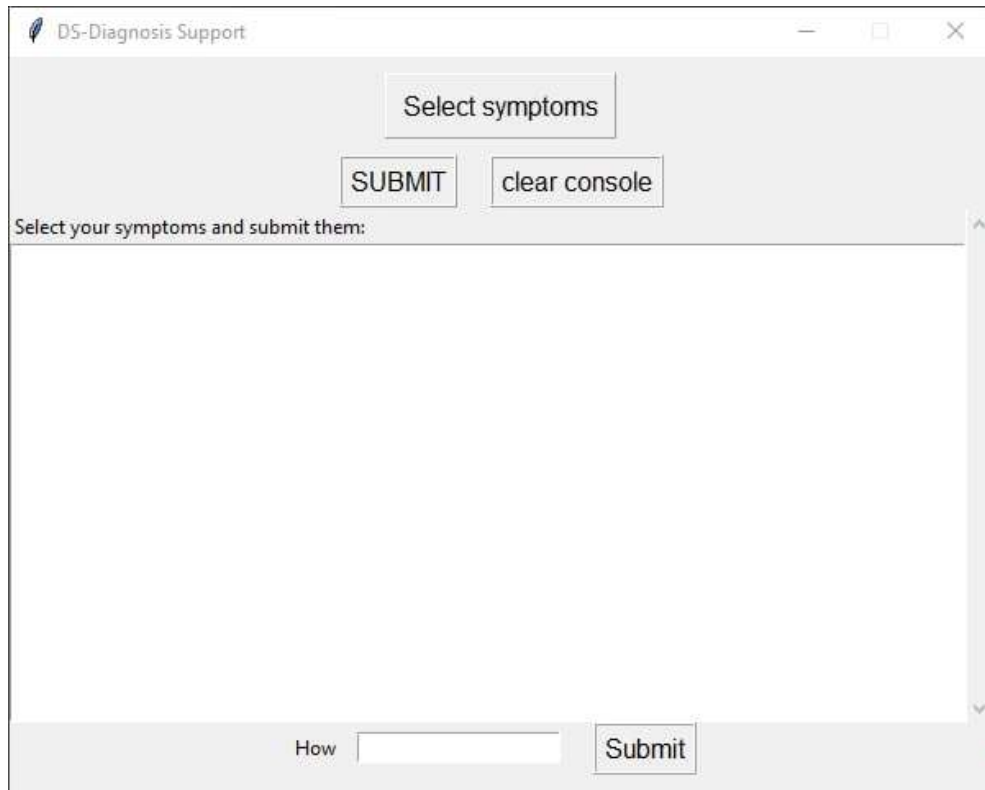
---

DS ha due principali compiti:

1. Visualizzare le malattie più probabili a seconda dei sintomi inseriti dall'utente
2. Dare una spiegazione del risultato

# GUI

---



L'interfaccia utente è semplice ed essenziale:  
Tramite il pulsante «Select symptoms» viene proposta all'utente la lista di tutti i possibili sintomi all'interno della KB, dai quali sceglierà quelli d'interesse.

Quelli scelti, appariranno all'interno della console, ed una volta scelti tutti, l'utente può:

- premere il pulsante «clear console» per resettare la console
- premere il pulsante «SUBMIT» per far partire l'inferenza e dunque visualizzare il risultato sulla console.

# Caso d'uso:

The screenshot shows a web application titled "DS-Diagnosis Support". It has a "Select symptoms" button at the top. Below it are "SUBMIT" and "clear console" buttons. A text input field contains the text "coma-confusion-muscle weakness-paralysis-". Below the input field is a "RESULT" section with a table of diseases and their probabilities.

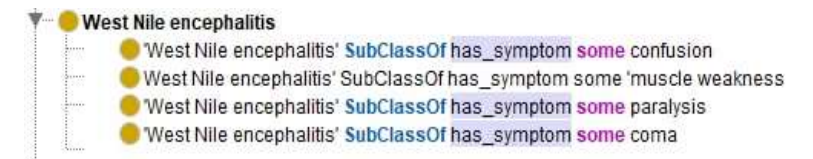
disease	%
West Nile encephalitis	26.67%
St. Louis encephalitis	13.33%
La Crosse encephalitis	13.33%
Japanese encephalitis	13.33%
Western equine encephalitis	6.67%
tropical spastic paraparesis	6.67%
bulbospinal polio	6.67%
staphyloenterotoxemia	6.67%
bagassosis	6.67%

At the bottom, there is a "How" label, an empty input field, and a "Submit" button.

L'utente inserisce come sintomi:  
coma, confusion, muscle weakness e paralysis.

Il sistema restituisce le malattie più probabili in base a quei sintomi. In questo caso la «West Nile encephalitis» ha la più alta probabilità di essere quella di cui soffre l'utente perché come abbiamo visto prima, questa malattia presenta esattamente i sintomi che l'utente ha inserito.

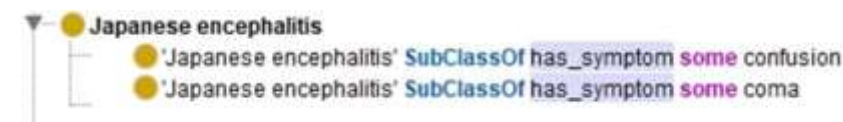
Infatti  
nell'ontologia:



# How question

L'inferenza del risultato è prodotta tramite un ragionamento invisibile agli occhi dell'utente, il quale però potrebbe essere interessato a scoprire più informazioni riguardanti la soluzione.

Per questo motivo è stata implementata anche la domanda HOW, che risponde all'esigenza dell'utente di sapere come una clausola è stata restituita.



The screenshot shows the 'DS-Diagnosis Support' window. The 'Select symptoms' section contains the text 'coma-confusion-muscle weakness-paralysis-'. Below this, the 'RESULT' section displays a list of diseases and their associated percentages:

disease	%
West Nile encephalitis	26.67%
St. Louis encephalitis	13.33%
La Crosse encephalitis	13.33%
Japanese encephalitis	13.33%
Western equine encephalitis	6.67%
tropical spastic paraparesis	6.67%
bulbospinal polio	6.67%
staphyloenterotoxemia	6.67%
bagassosis	6.67%

At the bottom, the 'How' section shows 'West Nile encephalitis' selected, and a 'Submit' button is visible.

The screenshot shows the 'DS-Diagnosis Support' window. The 'Select symptoms' section contains the text 'coma-confusion-muscle weakness-paralysis-'. Below this, the 'RESULT' section displays a list of diseases and their associated percentages:

disease	%
West Nile encephalitis	26.67%
St. Louis encephalitis	13.33%
La Crosse encephalitis	13.33%
Japanese encephalitis	13.33%
Western equine encephalitis	6.67%
tropical spastic paraparesis	6.67%
bulbospinal polio	6.67%
staphyloenterotoxemia	6.67%
bagassosis	6.67%

At the bottom, the 'How' section shows 'Japanese encephalitis' selected, and a 'Submit' button is visible.



# Architettura #1

---

Per gestire la conoscenza come un insieme di clausole definite sono state utilizzate le classi create da *David L. Poole* e *Alan K. Mackworth* prese dal sito *aipython.org*.

Ovvero le classi *logicProblem.py* e *logicBottomUp.py* che rispettivamente definiscono la KB come un insieme di clausole definite, e implementano la procedura di inferenza bottom-up.

Per la lettura dell'ontologia è stato utilizzato il modulo python *owlready2*, che fornisce tutte le funzioni necessarie per la manipolazione di ontologie.

A causa di problemi con il parser però l'ontologia è stata modificata tramite l'utilizzo di **Protégé** (in particolare lo splitter in *owlready2* non riusciva a ritrovare correttamente le triple a causa delle entità e relazioni all'interno dell'ontologia catalogate come “*obsolete*” da *DO* stessa).

Tramite il reasoner di *Protégé*, è stato dunque possibile esportare l'ontologia rimodellata nel file *do\_inferred* il quale viene poi utilizzato all'interno del programma.

# Architettura #2

---

La KB quindi viene creata a partire dalla classe *logicProblem.py*, che definisce le **clausole definite**, un **atomo askable**, la risposta **yes**, e la **KB** stessa.

Però, tramite questa logica, la KB avrebbe dovuto avere una conoscenza *completa*, perché una malattia ha un insieme ben definito di sintomi, e solo quei sintomi possono manifestarsi per far sì che quella malattia venga identificata.

Il problema però è che questa implementazione avrebbe prodotto un sistema eccessivamente selettivo, che avrebbe rilevato malattie solo nel caso in cui l'utente avesse inserito gli esatti sintomi, senza alcuna tolleranza (come per esempio un sintomo *falso positivo*).

Per questo motivo, il sistema sfrutta un **modello probabilistico**, che assegna ad ogni malattia appartenente al *punto fisso* della KB uno score (per ogni malattia appartenente al punto fisso lo score è tale che sia compreso tra 0 ed 1 e la somma degli score delle malattie appartenenti al punto fisso sia uguale ad 1). Questo score indica la probabilità che l'utente ha di essere affetto da quella malattia.

La probabilità viene calcolata sul numero dei sintomi in comune tra la malattia presa in considerazione ed i sintomi selezionati dall'utente.

Per esempio se definiamo  $S(u)$  l'insieme dei sintomi dell'utente, e con  $S(m)$  l'insieme dei sintomi che caratterizzano la malattia, allora la probabilità che l'utente abbia contratto la malattia  $m$  è denotata da:

$$\frac{|S(m)|}{|S(u)|}$$

Dunque se sono stati inseriti tutti e soli i sintomi della malattia  $m$

-nonostante questo non escluda il fatto che possano essere estratte anche altre malattie- la probabilità di aver contratto la malattia  $m$  sarà sicuramente maggiore o al più uguale alla probabilità di aver contratto un'altra malattia del punto fisso.