Business Context and Justification of Approach

In the current environment, the process of matching skills to a job description is a time-consuming exercise for both the recruiter and the job seeker. The manual review of candidates' CVs during the recruitment process or even the use of keywords as a mechanism to filter the candidates lacks accuracy, is cumbersome, is highly subjective, and in turn results in unsuitable matches (Negi and Kumar, 2014).

That being said, there is a significant need for recruitment services that can match job descriptions and CVs and in turn optimize resource allocation for companies and enhance the recruitment process in terms of accuracy and efficiency.

Job matching systems are intelligent systems that help companies to identify the best candidates for a specific role, but also help candidates to identify the job roles that are more suitable to them and in turn have more chances to be offered the job.

Compared to other job matching systems, such as the Content-based system that provides recommendations based on the attributes and characteristics of the CVs or the Collaborative filtering system that provides recommendations based on the behavior and preferences of multiple users, the ontology-based matching system is preferred as it combines job specificities, industry standards, and qualifications to provides its recommendations.

In particular, the ontology-based matching system, through semantic understanding of data, ensures that matches are done based relevance of all concepts and not merely based on exact skills matching (Grimm et al., 2011). In addition, flexibility of such systems allows for adding new skills or qualification as the market evolves in order to identify the most suitable candidates for the companies and the most suitable job roles for the job seekers (Tang et al., 2023; Zhao and Meersman, 2005).

To facilitate the recruitment process and given the benefits of an ontology-based systems in terms of effectiveness and personalised job searches as describes before, such a prototype system was developed using Protégé (DeBellis, 2021). The design follows a domain-driven structure that includes core classes (Table 1), object and data properties (Table 2), and a matching logic implemented through the use of SWRL rules (Figure 1).

Table 1. Core classes

Core class	**Description**
Job Seeker	Refers to individuals looking for a job
Job	Refers to open job opportunities
Skill	Refers to an abstract category used to match job requirements and candidates' CV
Qualification	Refers to an abstract category used to match job requirements and candidates' CV
Experience	Refers to an abstract category used to match job requirements and candidates' CV

Table 2. Data and object properties

Job Seeker	
Data properties	hasName, hasYearsExperience, hasPreferredLocation
Object properties	hasSkill, hasQualification, hasExperienceLevel
Job	
Object properties	requiresSkill, requiresQualification, requiresExperienceLevel, hasLocation

Figure 1. SWRL rule

```
JobSeeker(?js) ^ Job(?job) ^ hasSkill(?js, ?skill) ^ requiresSkill(?job, ?skill)->
matchesJob(?js, ?job)
```

Such design was adopted as it includes key variables of a recruitment process, such as qualification, skills, experience, and preferences, as well as because it provides the flexibility of adding further requirements such as language preference due to its semantic richness.

Finally, the use of SWRL rule enables the AI-driven goal of the service and it facilitates the automated reasoning in order job seekers to be provided roles more suitable to their profiles and companies to be able to shortlist the most suitable profiles for the roles they are seeking to fill.

Analysis of Outputs with Evidence of Testing

After populating the ontology with example individuals, such as JobSeekerJohnStan and SeniorProcurementOfficer, assigning appropriate skills, qualifications, and experience, SWRL rule was applied to infer job matches. Running the Pellet reasoner as well, this rule correctly matched the job seeker to a job role and such relationship was appeared in the relevant section (Object Property Assertions) confirming that that the system is working properly.

Testing was performed during the development of the ontology systems and included the application of data and object property assertions to sample individuals, running of reasoning (Pellet) at each stage to ensure the ontology is able to infer relationships without errors, and use of SWRL Tab in order to define relevant matching rules. The system throughout the testing process provided evidence of effectiveness and rule-based reasoning for job matching. Appendix 1 includes screenshots of the outputs from Protégé.

As already mentioned, this ontology is not a theoretical framework but is a practical solutions that can help for companies and job seekers during the recruitment process. It is actually a semi-automated screening process that can be used to flag matches between CVs and job roles based on pre-defined rules.

The ontology that has been developed is a simple prototype and the objective is to present the capabilities of such system. However, considering the flexibility of ontology-based systems, this ontology can expand and include new job sectors, diverse candidates' CVs, additional rules with pre-defined weights, as well as additional preferences.

Additionally, given that the system does not perform the matching based on job titles or keywords but relies on semantic relationships, it can be used even in cases where job descriptions are continuously changing or where regional criteria may be applied.

Conclusion

This prototype ontology supports intelligent job matching and could result in effectiveness and efficiency gains during the recruitment process, providing several benefits for both the companies and job seekers. It provides a foundation for further development and subsequent deployment in an HR environment. Its reasoning capabilities and flexible design, contribute to the scalability and explainability of the systems, essential elements of an automated job matching system.

References

DeBellis , M. (2021). *A Practical Guide to Building OWL Ontologies. Using Protégé 5.5 and Plugins.*

Grimm, S., Abecker, A., Völker, J., & Studer, R. (2011). Ontologies and the Semantic Web. In *Handbook of Semantic Web Technologies* (pp. 507–579). Berlin, Heidelberg: Springer Berlin Heidelberg.

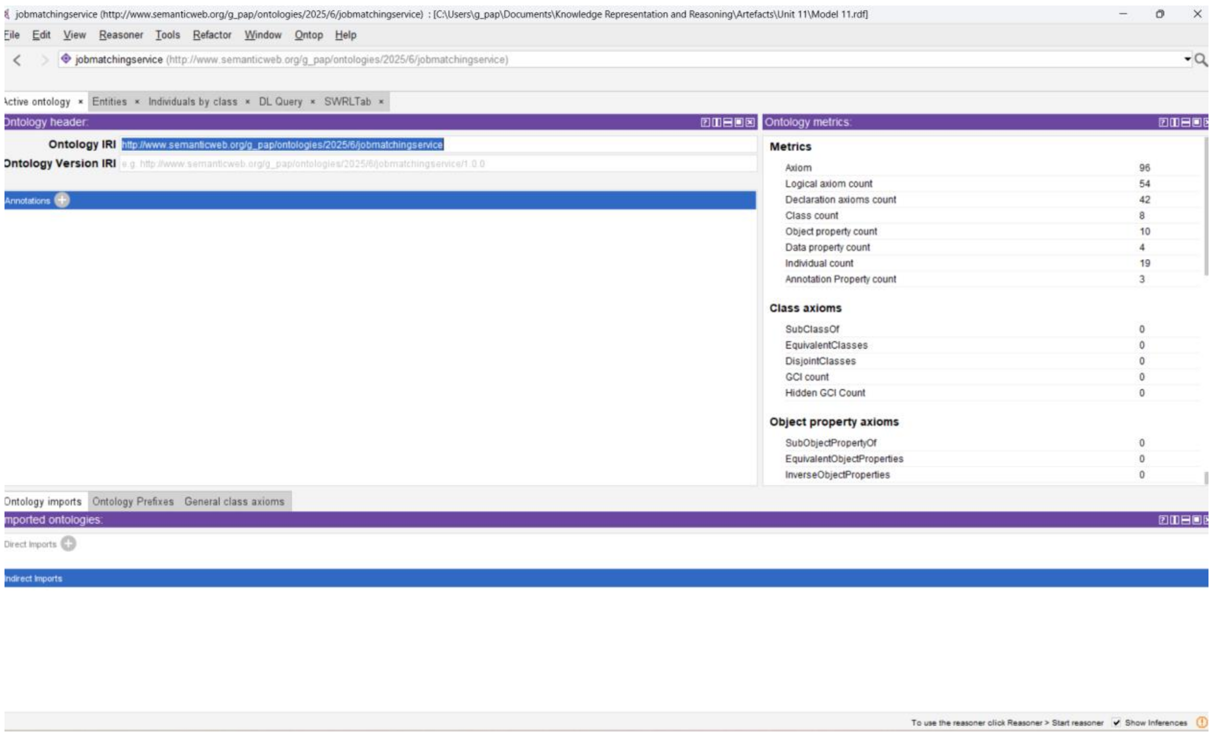
Negi, Y., & Kumar, S. (2014). A comparative analysis of keyword- and semantic-based search engines. In D. Mohapatra, & S. Patnaik, *Intelligent Computing, Networking, and Informatics* (pp. 727–736). New Delhi: Springer India.

Tang, X., Feng, Z., Xiao, Y., Wang, M., Ye, T., Zhou, Y., . . . Zhang, D. (2023). Construction and application of an ontology-based domain-specific knowledge graph for petroleum exploration and development. *Geoscience Frontiers, 14*(5), 101426.

Zhao, G., & Meersman, R. (2005). Architecting ontology for scalability and versatility. In R. Meersman, & Z. Tari, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE* (pp. 1605–1614). Berlin, Heidelberg: Springer Berlin Heidelberg.

Model 11.rdf Model 11.properties

Appendix 1. Screenshots of Progete outputs


jobmatchingservice (http://www.semanticweb.org/g_pap/ontologies/2025/6/jobmatchingservice) : [C:\Users\g_pap\Documents\Knowledge Representation and Reasoning\Artefacts\Unit 11\Model 11.rdf]
File Edit View Reasoner Tools Refactor Window Ontop Help
jobmatchingservice (http://www.semanticweb.org/g_pap/ontologies/2025/6/jobmatchingservice)
Active ontology Entities Individuals by class DL Query SWRLTab
Ontology header:
Annotations
Ontology imports Ontology Prefixes General class axioms
Imported ontologies:
Direct Imports
Indirect Imports
To use the reasoner click Reasoner > Start reasoner
Show Inferences

Ontology IRI http://www.semanticweb.org/g_pap/ontologies/2025/6/jobmatchingservice

Ontology Version IRI e.g. http://www.semanticweb.org/g_pap/ontologies/2025/6/jobmatchingservice/1.0.0

Ontology metrics:

Metrics

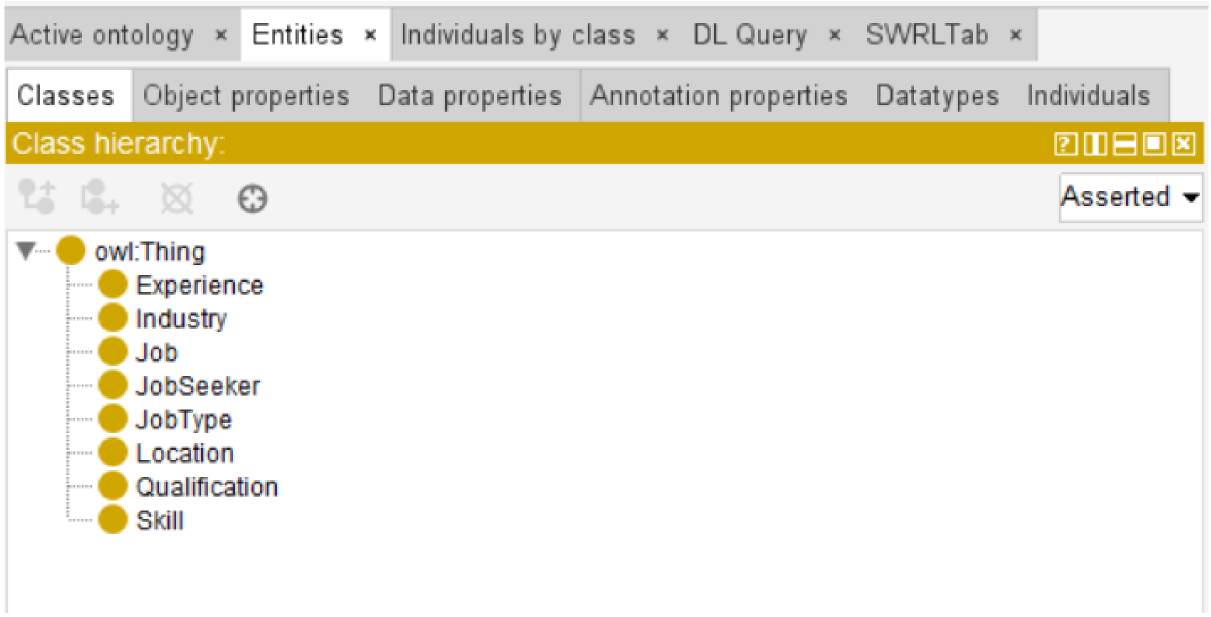
Axiom	96
Logical axiom count	54
Declaration axioms count	42
Class count	8
Object property count	10
Data property count	4
Individual count	19
Annotation Property count	3

Class axioms

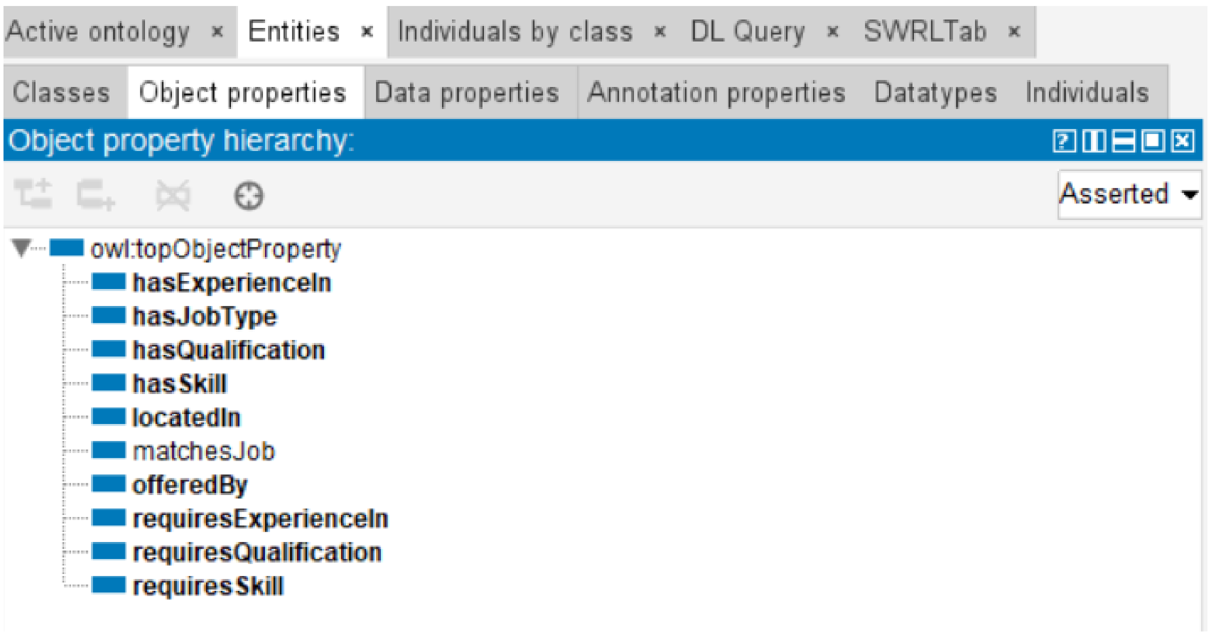
SubClassOf	0
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0

Object property axioms

SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	0

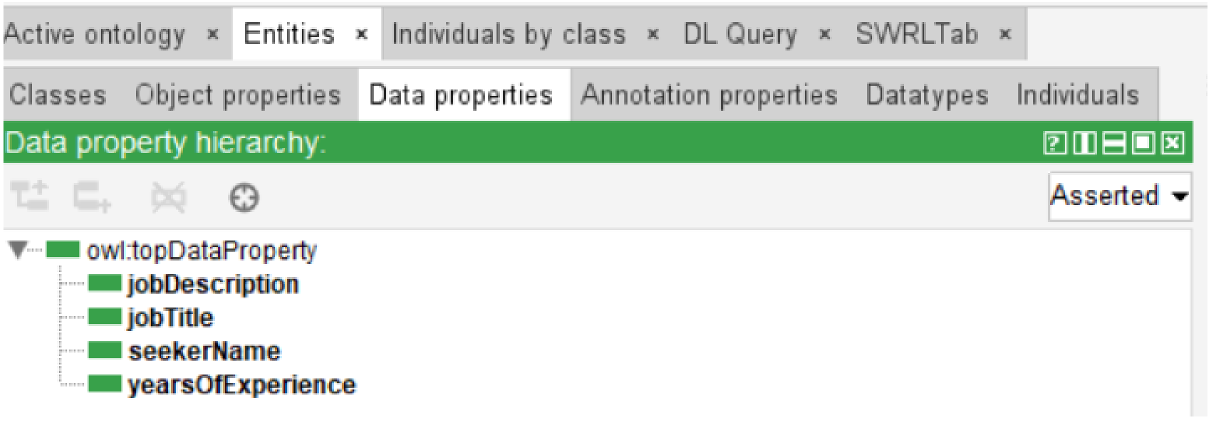


owl:Thing
- Experience
- Industry
- Job
- JobSeeker
- JobType
- Location
- Qualification
- Skill



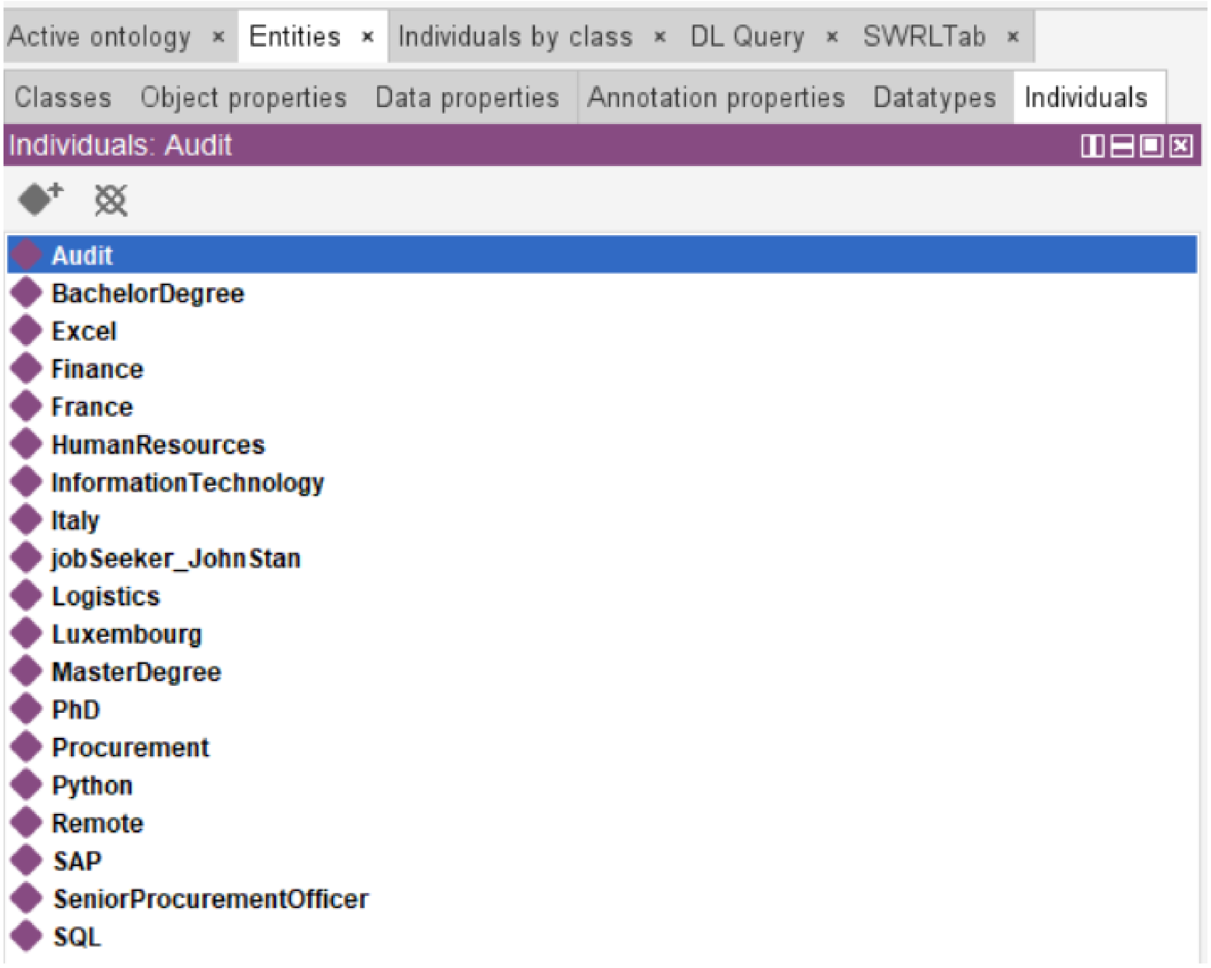
Object property hierarchy:

- owl:topObjectProperty
  - hasExperienceIn
  - hasJobType
  - hasQualification
  - hasSkill
  - locatedIn
  - matchesJob
  - offeredBy
  - requiresExperienceIn
  - requiresQualification
  - requiresSkill



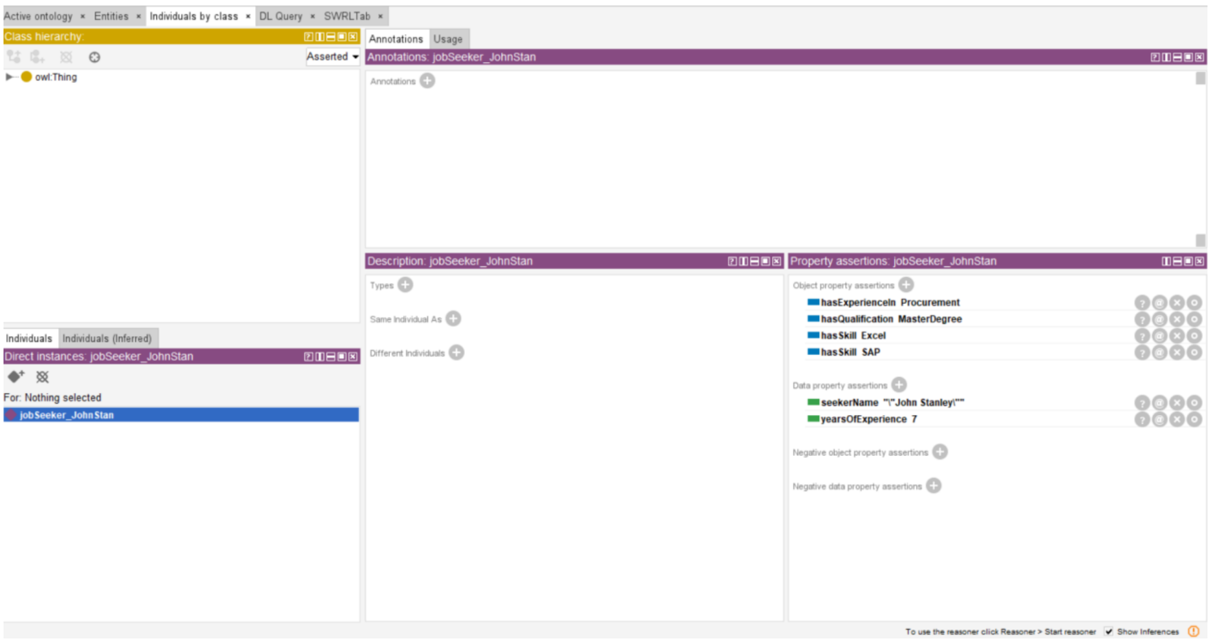
Data property hierarchy:

- owl:topDataProperty
  - jobDescription
  - jobTitle
  - seekerName
  - yearsOfExperience



Individuals: Audit

Audit
BachelorDegree
Excel
Finance
France
HumanResources
InformationTechnology
Italy
jobSeeker_JohnStan
Logistics
Luxembourg
MasterDegree
PhD
Procurement
Python
Remote
SAP
SeniorProcurementOfficer
SQL



Annotations: jobSeeker_JohnStan

Annotations

Direct instances: jobSeeker_JohnStan

For: Nothing selected

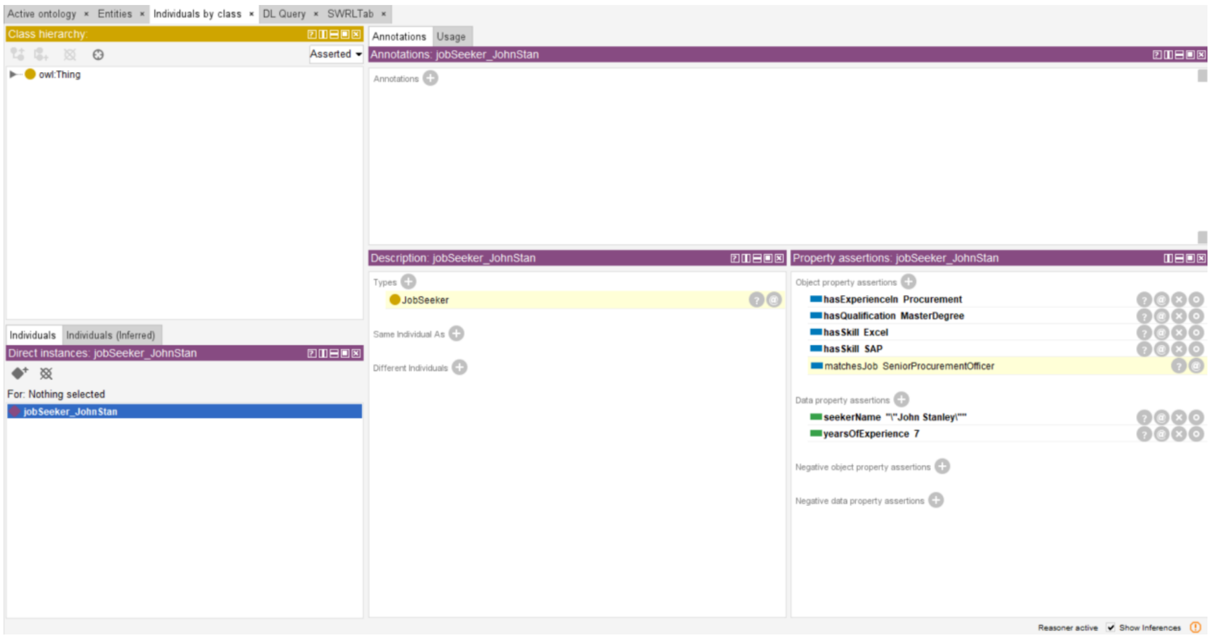
jobSeeker_JohnStan

Description: jobSeeker_JohnStan

Property assertions: jobSeeker_JohnStan

Object property assertions
hasExperienceIn Procurement
hasQualification MasterDegree
hasSkill Excel
hasSkill SAP

Data property assertions
seekerName "\"John Stanley\""
yearsOfExperience 7


Active ontology × Entities × Individuals by class × DL Query × SWRLTab ×
Class hierarchy:
Asserted
owl:Thing
Annotations Usage
Annotations: jobSeeker_JohnStan
Annotations
Individuals Individuals (Inferred)
Direct instances: jobSeeker_JohnStan
For: Nothing selected
jobSeeker_JohnStan
Description: jobSeeker_JohnStan
Types
Same Individual As
Different Individuals
Property assertions: jobSeeker_JohnStan
Reasoner active
Show Inferences

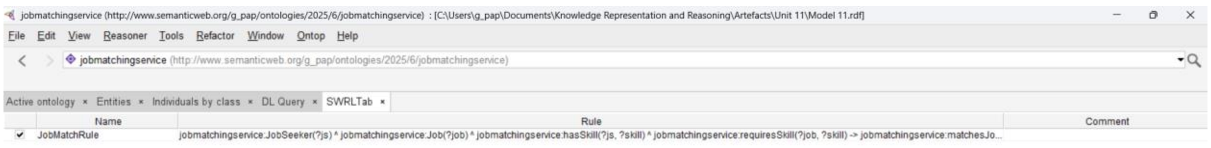
Types: JobSeeker

Object property assertions
- hasExperienceIn Procurement
- hasQualification MasterDegree
- hasSkill Excel
- hasSkill SAP
- matchesJob SeniorProcurementOfficer

Data property assertions
- seekerName "\"John Stanley\""
- yearsOfExperience 7

Negative object property assertions

Negative data property assertions


File Edit View Reasoner Tools Refactor Window Ontop Help
Active ontology × Entities × Individuals by class × DL Query × SWRLTab ×

jobmatchingservice (http://www.semanticweb.org/g_pap/ontologies/2025/6/jobmatchingservice) : [C:\Users\g_pap\Documents\Knowledge Representation and Reasoning\Artefacts\Unit 11\Model 11.rdf]

jobmatchingservice (http://www.semanticweb.org/g_pap/ontologies/2025/6/jobmatchingservice)

	Name	Rule	Comment
✓	JobMatchRule	jobmatchingservice:JobSeeker(?js) ^ jobmatchingservice:Job(?job) ^ jobmatchingservice:hasSkill(?js, ?skill) ^ jobmatchingservice:requiresSkill(?job, ?skill) -> jobmatchingservice:matchesJo...	