University of Essex Online

MSc Artificial Intelligence

Report

Modelling Assignment

Submission date: 14.07.2025

Contents

Introduction

This report aims to design and document an ontology for an AI-driven job-matching service. The Protégé software will be used to set up the ontology.

The Job Matching Service

The proposed job matching service uses ontology and artificial intelligence (AI) techniques to optimise the matching process between job seekers and job opportunities. Traditional job portals often struggle to match complex profiles. They rely heavily on static keywords and criteria that do not take nuances in candidate skills, preferences, and experiences into account (Kenthapadi et al., 2017). In contrast to this, an ontology-based approach enables structured and semantically meaningful representation of job-related knowledge. This helps to facilitate the complex process of job-matching (Fernández-López & Gómez-Pérez, 2002).

By employing ontology-based AI systems, job searches become more effective and personalised, significantly improving user satisfaction and the quality of outcomes (Shafique & Qaiser, 2014). This approach supports complex queries and semantic reasoning, which cannot be achieved using traditional methodswith static keyword and criteria (Staab & Studer, 2009). With ontology-based systems users experience enhanced job recommendations. This can lead to higher engagement and potentially accelerating employment outcomes. Furthermore, the adaptability and scalability of ontology-based models ensure applicability across all job sectors and easily covers evolving candidate profiles (Horridge et al., 2004).

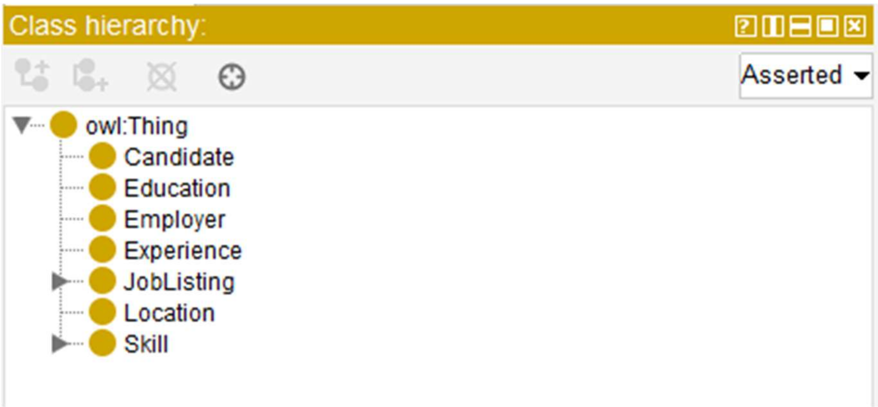
Ontology

Design Decisions and Building

The ontology for the job matching service was designed considering the fundamental aspects of employment opportunities and candidate profiles. The design aims at helping the service to achieve precise job matching. The following shows the main rationales for the ontology.

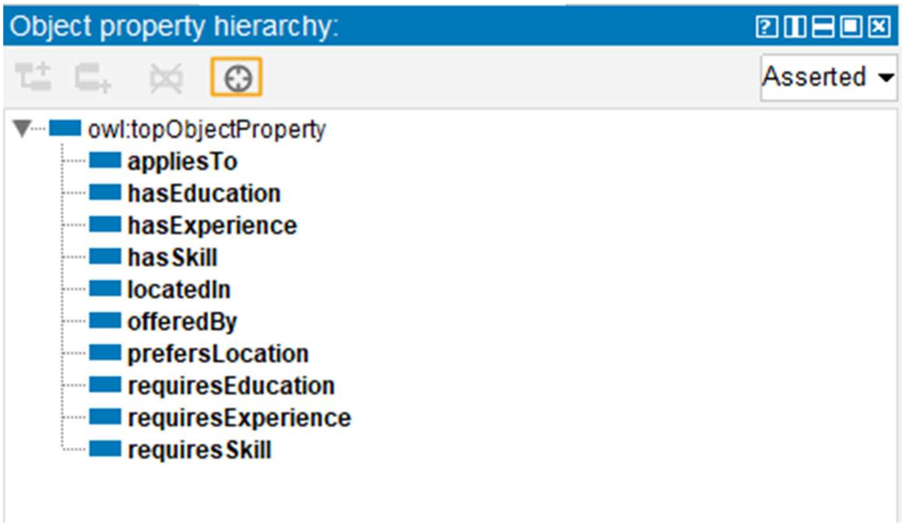
Main Classes

1. **Candidate:** Attributes include skills, working experience, education level, and job preferences.
2. **Job Listing**: Attributes include required skills, role, location, salary range, and employer.
3. **Skill**: Documents skills required by the job or possessed by candidates.
4. **Experience**: Represents previous employment roles and tenure.
5. **Education**: Details educational qualifications.
6. **Employer**: Contains information about companies offering jobs.
7. **Location**: Geographical location data related to job listing and candidate preferences.



Object Properties

1. Candidate **hasSkill** Skill
2. Candidate **hasExperience** Experience
3. Candidate **hasEducation** Education
4. Candidate **appliesTo** JobListing
5. JobListing **requiresSkill** Skill
6. JobListing **requiresExperience** Experience
7. JobListing **requiresEducation** Education
8. JobListing **offeredBy** Employer
9. JobListing **locatedIn** Location
10. Candidate **prefersLocation** Location

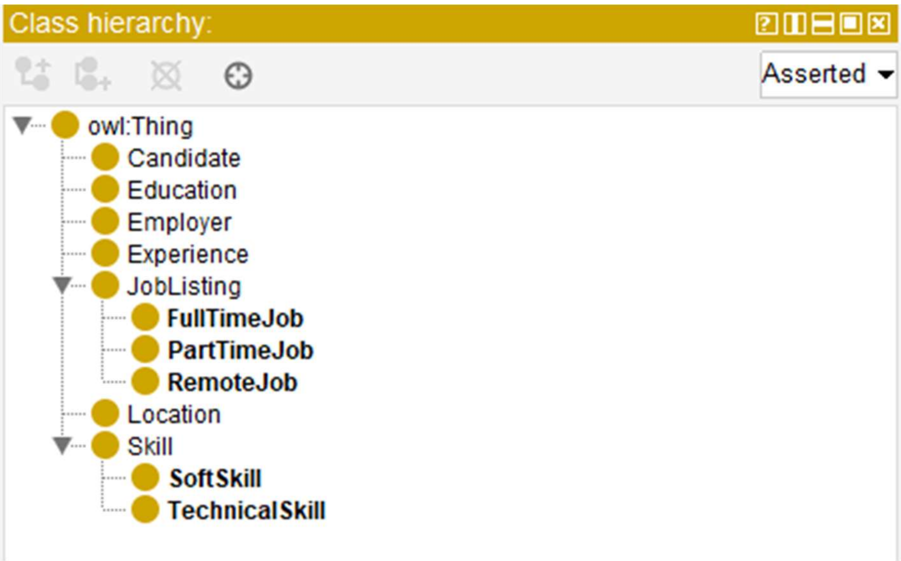


Object property hierarchy:

owl:topObjectProperty
- appliesTo
- hasEducation
- hasExperience
- hasSkill
- locatedIn
- offeredBy
- prefersLocation
- requiresEducation
- requiresExperience
- requiresSkill

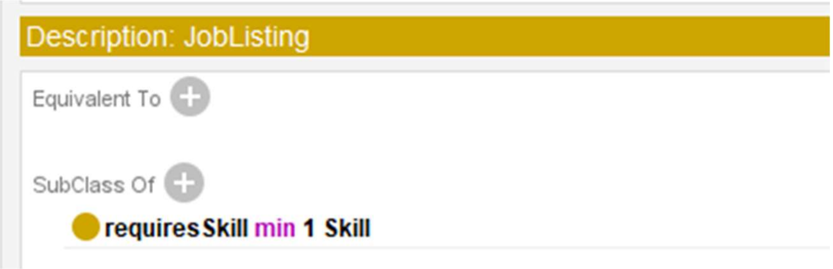
Class Hierarchy (Excerpt)

- JobListing
  - Subclasses: **FullTimeJob, PartTimeJob, RemoteJob**
- Skill
  - Subclasses: **TechnicalSkill, SoftSkill**



Constraints (Restrictions)

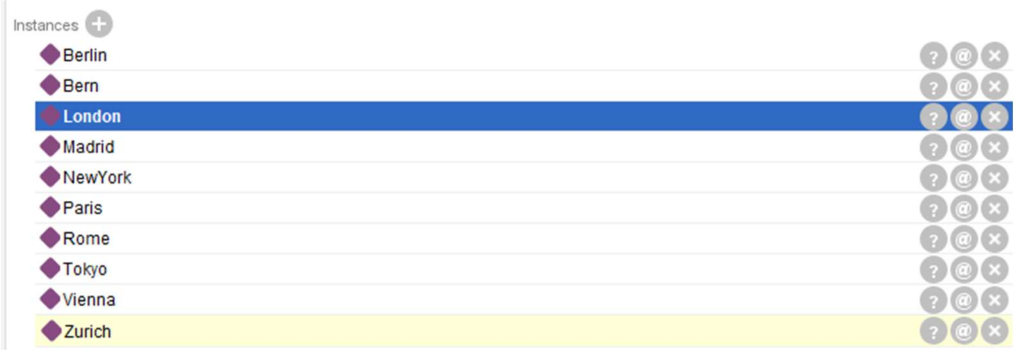
- JobListing must have at least one requiredSkill.
- A Candidate must have at least one hasSkill relation.
- Education levels follow a hierarchical relationship (Bachelor < Master < PhD)



requiresSkill min 1 Skill

Creating the Ontology in Protégé required us to perform the following steps:

1. We build ontology in Protégé by adding the classes, properties, class hierarchy and constraints.
2. We then add realistic instances which means we will add realistic examples of locations, job seekers, job opportunities and others.



Instances

- Berlin
- Bern
- London
- Madrid
- NewYork
- Paris
- Rome
- Tokyo
- Vienna
- Zurich

3. Reasoning was used to infer new knowledge from the ontology. For example, a candidate possessing a skill from a general class may be considered suitable for jobs requiring more specific subclasses of that skill. The ontology was tested using both the Pellet and HermiT reasoners, and no inconsistencies or logical errors were detected.

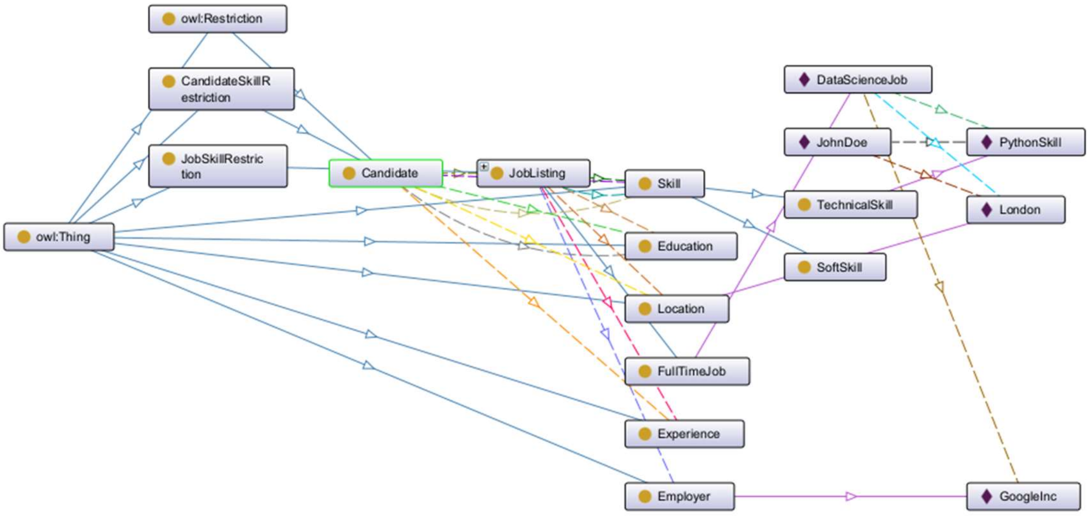
Testing

Additionally, the DL Query tab was used to test the ontology by issuing semantic queries and verifying that the expected results appeared.

Test Query	Expected Result	Testing Result
Candidate	Shows all candidate instances	OK

Candidate and (hasSkill some TechnicalSkill)	Find all candidates with any technical skill	OK
Candidate and (hasEducation some BScComputerScience)	Shows all Candidates with a BSc in Computer Science	OK
Candidate and (prefersLocation some Zurich)	Shows all candidates that have a preferesLocation relation to Zurich	OK
JobListing and (locatedIn some Bern)	Shows all job listings that are located in Bern	OK
JobListing and (requiresSkill some Teamwork)	Shows all jobs that require teamwork	OK
Candidate and (hasExperience some ProjectManager)	Shows all candidates with project manager experience	OK
Locations	Shows all location instances	OK

Conceptual Model Diagram of the Ontology Design



Business Justification

The ontology designed supports the job matching service providing a comprehensive knowledge structure enhancing the service by enabling context representation of job listings as well as candidates. By modeling relevant attributes and relationships, the ontology allows complex semantic queries that allow personalized job matching for each candidate. In summary the ontology allows the system to better achieve its goals of matching candidates and job listings.

Application for the Job Matching Case

When it comes to scalability and adaptability the ontology demonstrates significant potential for both. It allows job matching across different job sectors and with diverse candidate profiles. The architecture ontology is very structured yet flexible, which allows for easy updates and expansions. This ensures scalability and flexibility in case new information needs to be considered by the ontology due to an ever-evolving job market. The ontology fits well even for various job sectors and candidate profiles and can be expanded whenever there is something missing for a specific use—case.

Conclusion

In conclusion, this report shows how ontology-driven design can significantly enhance the quality of AI services, using the job-matching system as an example. By introducing semantic modelling and reasoning, the ontology facilitates intelligent, personalised, and scalable matches that overcome the limitations of traditional static keyword based systems. This approach offers a promising way forward for addressing the growing complexity of future labour markets.

References

Kenthapadi, K., Le, B. and Venkataraman, G., 2017, August. Personalized job recommendation system at linkedin: Practical challenges and lessons learned. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 346-347).

Fernández-López, M. and Gómez-Pérez, A., 2002. Overview and analysis of methodologies for building ontologies. *The knowledge engineering review*, *17*(2), pp.129-156.

Shafique, U. and Qaiser, H., 2014. A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), pp.217-222.

Staab, S. and Studer, R. eds., 2013. *Handbook on ontologies*. Springer Science & Business Media.

Horridge, M., Knublauch, H., Rector, A., Stevens, R. and Wroe, C., 2004. A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools edition 1.0. *University of Manchester*.