Developing an ontological foundation for an AI-based job matching service

1. Introduction

Ontologies play a crucial role in artificial intelligence applications by providing structured knowledge representation that enables semantic reasoning beyond simple data processing (Gruber, 1993). They serve as formal specifications of shared conceptualizations, allowing AI systems to understand relationships between entities and infer implicit knowledge (Studer, Benjamins and Fensel, 1998). This assignment explores the development of a prototype ontology as the foundation for an AI-driven job-matching service.

2. Business context and justification for technical approach

The current job market faces significant challenges in connecting qualified candidates with suitable positions, with the proliferation of social-media recruiting and automated application systems paradoxically making efficient matching more difficult (Cardoso, Mourão and Rocha, 2021). Three main critical problems persist despite technological advances: 1) skill misalignment, where employers struggle to identify candidates whose capabilities truly match requirements; 2) inefficient matching processes that overlook qualified candidates due to terminology variations; and 3) increasing specialization across industries, demanding more nuanced understanding of skill transferability.

Traditional keyword matching systems operate on lexical similarity rather than semantic understanding, failing to recognize synonymous terms or related concepts (García-Sánchez et al., 2006). Alternative approaches like statistical machine learning require extensive training data and produce opaque decisions, while rule-based systems prove rigid and difficult to maintain as domains evolve. By contrast, ontologies provide crucial context by establishing meaningful connections between concepts, enabling the system to infer, for example, that "Python" is a "programming language" applicable to "data science" (Mochol, Oldakowski and Heese, 2004).

This semantic foundation supports sophisticated matching based on conceptual understanding rather than exact terminology, dramatically improving accuracy by uncovering hidden connections between qualifications and requirements (Colucci et al., 2003). Furthermore, the ontology approach offers explainable reasoning with transparent knowledge structures and flexible representation that evolves with domain changes, addressing fundamental limitations of alternative technologies in the recruitment space.

3. Ontology design and implementation

The proposed ontology design was developed following the Methontology framework (Fernandez, Gomez-Pearez and Juristo, 1997). This method was selected for its versatile approach to ontology creation (Yun et al., 2021), and encompasses specification, conceptualization, formalization, implementation, and maintenance phases. The ontology specification supports an AI-driven job-matching service by modeling relationships between candidates, postings, skills, experiences, and preferences. Knowledge acquisition based on personal experience and exploration of publicly-available job postings on LinkedIn identified key elements: industry, location, qualification, experience, and skills.

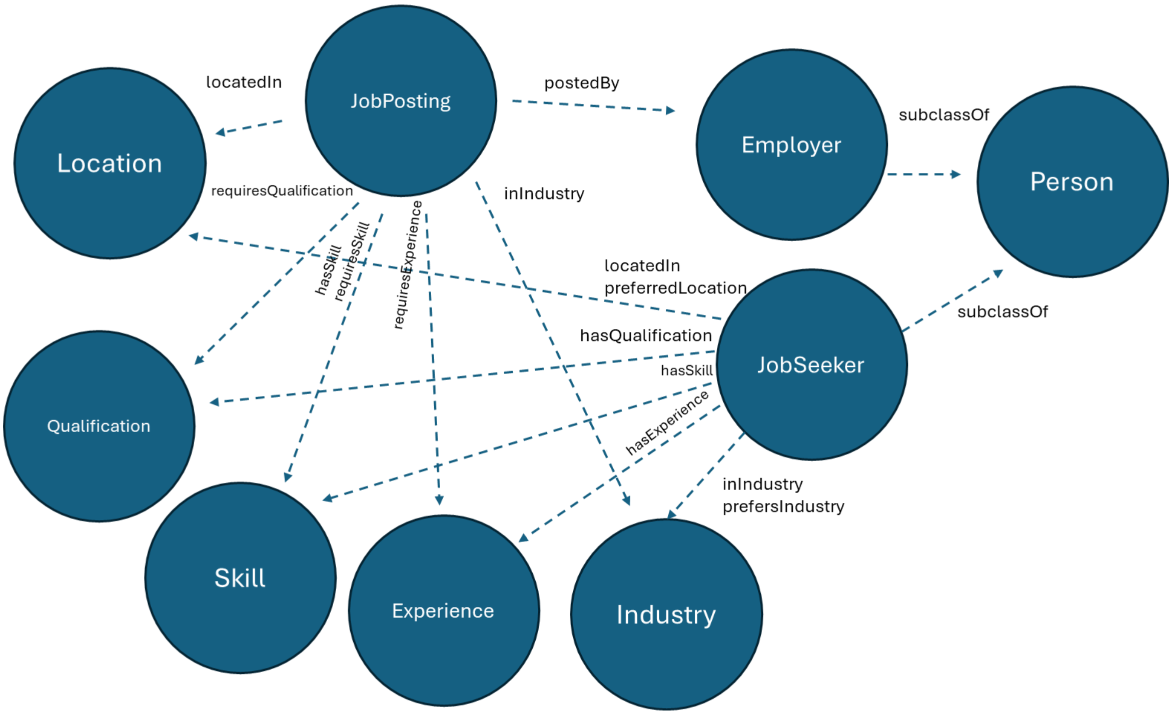
The structure comprises three main components (Figure 1):

- Class hierarchy: core classes include JobSeeker, Employer, JobPosting, Skill, Industry, Location, Qualification, and Experience, representing rigid properties enabling reasoning beyond terminology matches (Guarino and Welty, 2002).

- Object properties: these create semantic connections between entities, with hierarchical relationships enabling inference about implicit knowledge, distinguishing between factual information and preferences.

- Data properties: quantifiable attributes enable sophisticated compatibility assessment beyond binary matching.

Figure 1 - Conceptualizaton of a simple job-search ontology



Implementation was performed in Protégé v5.6.6 with manual property assignments (Debellis, no date). Defined classes were employed to support inference, with the Pellet reasoner ensuring coherence. The ontology was populated with 20 fictional entities, demonstrating functionality across multiple industries, skills, and qualification levels (Figure 2, Tables 1-3).

Figure 2 - OntoGraph layout of the ontology structure

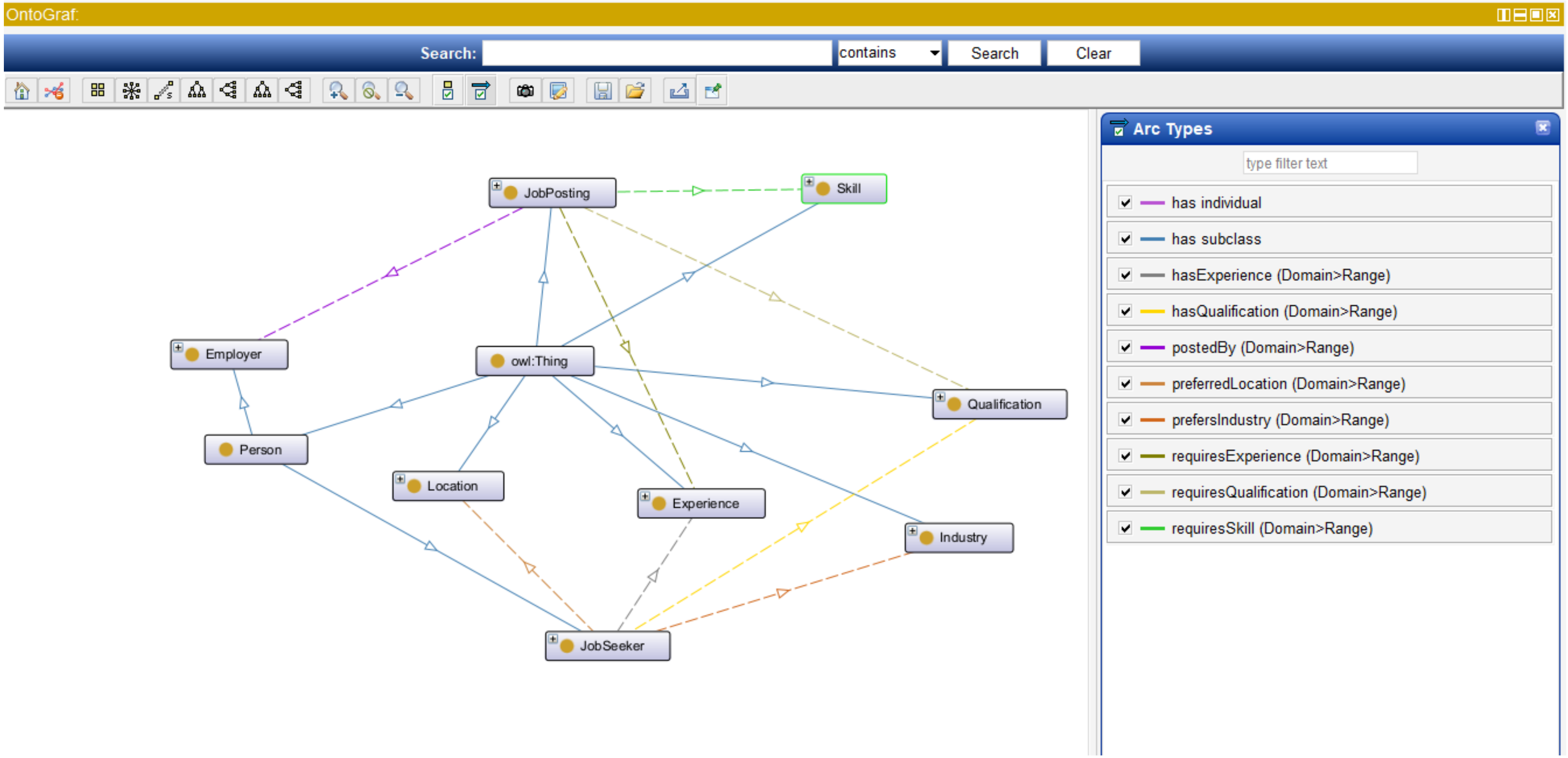


Table 1 - Summary of classes and instances included in the ontology

Core class	**Number of instances**	**Example instances**
Person		
- Employer	5	EduInstitute, FinanceGroup, HealthOrg, RetailChain, TechCorp
- Jobseeker	20	JobSeeker1-20
JobPosting	20	Job1-20
Experience	5	EntryLevel, JuniorLevel, MidLevel, SeniorLevel, ExecutiveLevel
Location	8	Berlin, London, NewYork, Remote, SanFrancisco, Sydney, Tokyo, Toronto
Skill	15	Accounting, CloudComputing, Communication, ContentWriting, CustomerService, DataAnalysis, GraphicDesign, HumanResources, Leadership, MachineLearning, Marketing, ProblemSolving, Programming, ProjectManagement, Sales
Industry	8	Consulting, Education, Finance, Healthcare, Manufacturing, Media, Retail, Technology
Qualification	5	HighSchoolDiploma, BachelorsDegree, AssociateDegree, MastersDegree, PhDDegree

Table 2 - Object properties implemented

Object property	**Domain**	**Range**
hasExperience	JobSeeker	Experience
hasQualification	JobSeeker	Qualification
hasSkill	JobPosting or JobSeeker	Skill
- requiresSkill	JobPosting	Skill
inIndustry	JobPosting or JobSeeker	Industry
- prefersIndustry	JobSeeker	Industry
locatedIn	JobPosting or JobSeeker	Location
- preferredLocation	JobSeeker	Location
postedBy	JobPosting	Employer
requiresExperience	JobPosting	Experience
requiresQualification	JobPosting	Qualification

Table 3 - Data properties implemented

Data property	**Domain**	**Range**
degreeLevel	Qualification	xsd:string
description	JobPosting	xsd:string
expectedSalary	JobSeeker	xsd:integer
jobTitle	JobPosting	xsd:string
name	Person	xsd:string
salary	JobPosting	xsd:integer
skillLevel	Skill	xsd:string
yearsOfExperience	Experience	xsd:integer

4. Model testing and analysis

Unlike other AI domains (e.g. machine learning) where formal, quantitative model evaluation metrics are well-established and widely-used, no standard evaluation framework exists for ontologies. Instead, ontology evaluation approaches can be considered from the perspective of intrinsic (i.e. of the design itself) versus extrinsic value (for a given purpose), and grouped by broad methodological approach (gold-standard, task-, corpus-, and criteria-based) (Raad and Cruz, 2015). For this assignment, a task-based method was employed, focusing on adaptability, efficiency, and consistency to assess whether the ontology could adequately model the specified domain.

Two evaluation approaches were implemented. First, defined classes were created for specific categories (e.g., technical jobs in the US, candidates for remote software development) using OWL-based specifications (Figures 3-4) (Horridge et al., 2006). The reasoner's classification combined with manual verification confirmed adequate specification.

Figure 3 - Job search using OWL statements to create defined classes (technical jobs in the US)

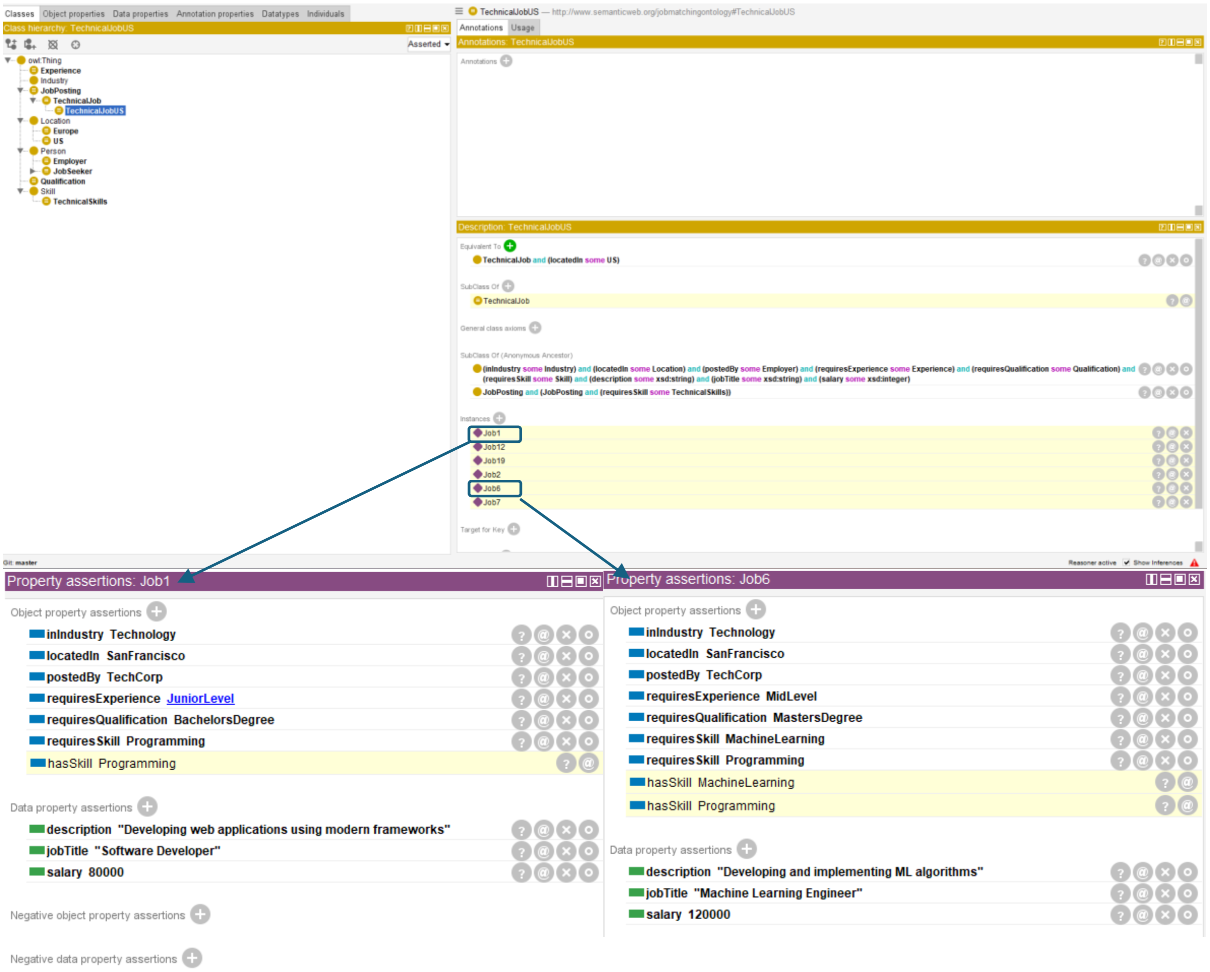
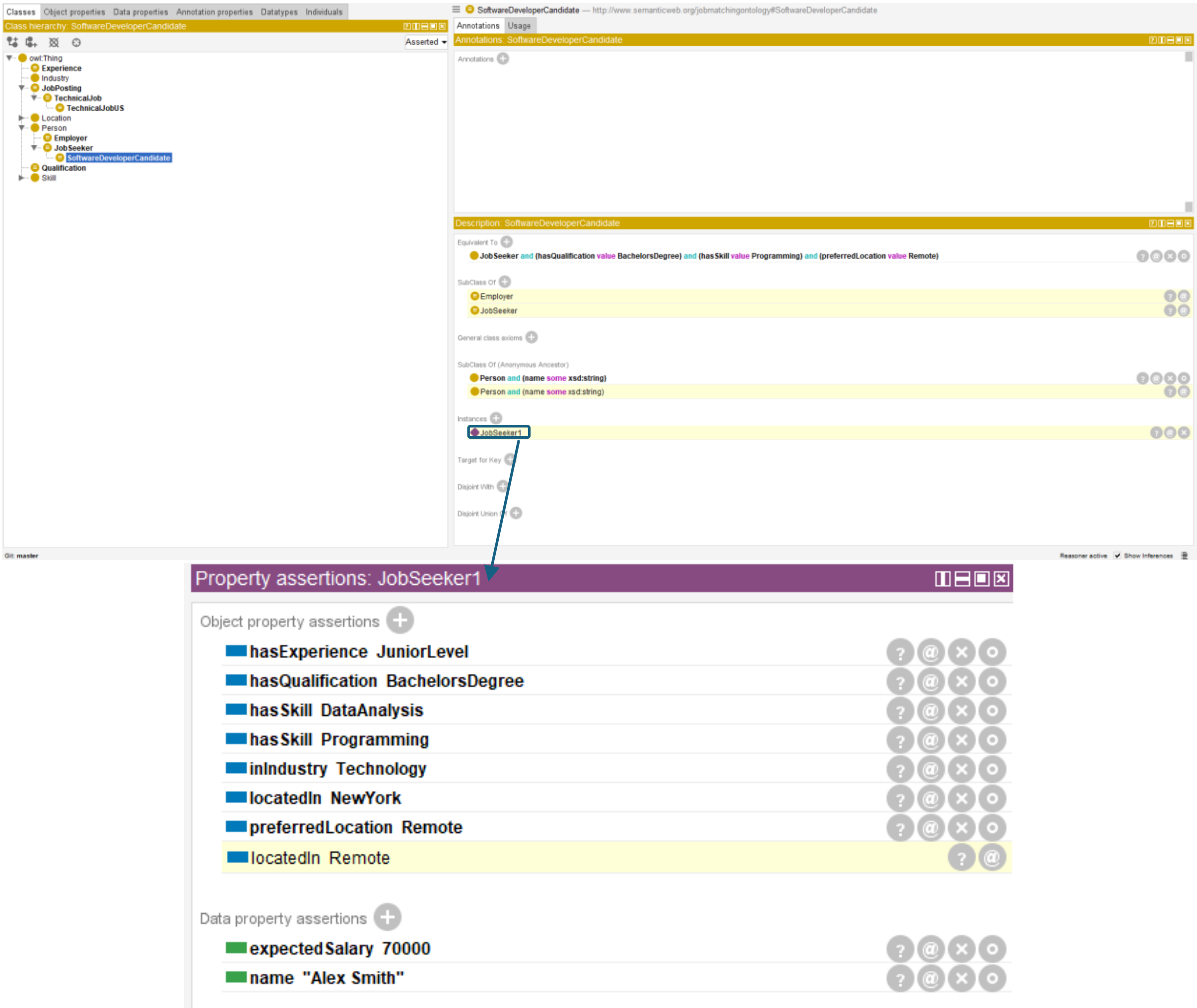
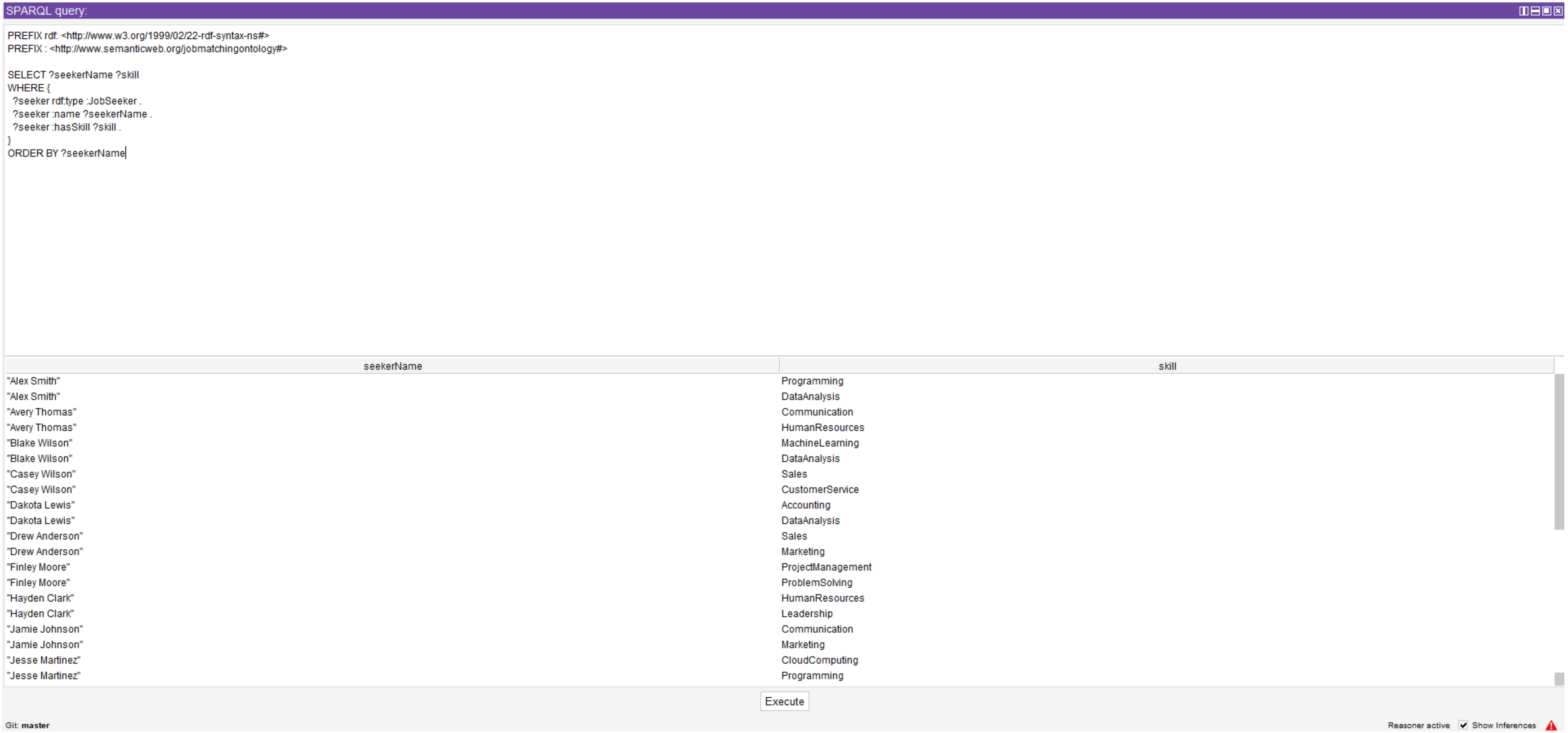


Figure 4 - Candidate search using OWL statements to create defined classes (software developer candidates for remote jobs)



Second, SPARQL queries were developed to select candidates or postings matching specific criteria (Figures 5-8). SPARQL was chosen over alternatives due to its superior retrieval and path handling capabilities (Perez, Arenas and Gutierrez, 2006). The testing results demonstrate the ontology's ability to handle increasingly complex queries that mirror real-world job matching scenarios. In particular, the query in Figure 9 successfully identified candidates matching multiple criteria simultaneously (skills, experience level, location preferences, and salary expectations), simulating how an actual recruitment process would filter candidates across multiple dimensions. This comprehensive matching capability represents a significant advancement over traditional keyword-based systems.

Figure 5 - Simple search for all candidates and skills (using SPARQL query)




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

SELECT ?seekerName ?skill
WHERE {
  ?seeker rdf:type :JobSeeker .
  ?seeker :name ?seekerName .
  ?seeker :hasSkill ?skill .
}
ORDER BY ?seekerName
```

seekerName	skill
"Alex Smith"	Programming
"Alex Smith"	DataAnalysis
"Avery Thomas"	Communication
"Avery Thomas"	HumanResources
"Blake Wilson"	MachineLearning
"Blake Wilson"	DataAnalysis
"Casey Wilson"	Sales
"Casey Wilson"	CustomerService
"Dakota Lewis"	Accounting
"Dakota Lewis"	DataAnalysis
"Drew Anderson"	Sales
"Drew Anderson"	Marketing
"Finley Moore"	ProjectManagement
"Finley Moore"	ProblemSolving
"Hayden Clark"	HumanResources
"Hayden Clark"	Leadership
"Jamie Johnson"	Communication
"Jamie Johnson"	Marketing
"Jesse Martinez"	CloudComputing
"Jesse Martinez"	Programming

Figure 6 - Find matching jobs and candidates based on specific skills

SPARQL query:

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

SELECT ?seekerName ?jobTitle ?skill
WHERE {
  ?seeker rdf:type :JobSeeker .
  ?seeker :name ?seekerName .
  ?seeker :hasSkill ?skill .

  ?job rdf:type :JobPosting .
  ?job :jobTitle ?jobTitle .
  ?job :requiresSkill ?skill .
}
ORDER BY ?seekerName ?jobTitle
```

seekerName	jobTitle	skill
"Alex Smith"	"Cloud Engineer"	Programming
"Alex Smith"	"Data Analyst"	DataAnalysis
"Alex Smith"	"Financial Analyst"	DataAnalysis
"Alex Smith"	"Machine Learning Engineer"	Programming
"Alex Smith"	"Marketing Analyst"	DataAnalysis
"Alex Smith"	"Research Scientist"	DataAnalysis
"Alex Smith"	"Senior Software Engineer"	Programming
"Alex Smith"	"Software Developer"	Programming
"Avery Thomas"	"Content Writer"	Communication
"Avery Thomas"	"HR Specialist"	Communication
"Avery Thomas"	"HR Specialist"	HumanResources
"Avery Thomas"	"Sales Representative"	Communication
"Avery Thomas"	"Teacher"	Communication
"Blake Wilson"	"Data Analyst"	DataAnalysis
"Blake Wilson"	"Financial Analyst"	DataAnalysis
"Blake Wilson"	"Machine Learning Engineer"	MachineLearning
"Blake Wilson"	"Marketing Analyst"	DataAnalysis
"Blake Wilson"	"Research Scientist"	MachineLearning
"Blake Wilson"	"Research Scientist"	DataAnalysis
"Casey Wilson"	"Customer Service Representative"	CustomerService

Figure 7 - Find remote jobs with salary above a threshold

SPARQL query:

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

SELECT ?jobTitle ?employerName ?salary
WHERE {
  ?job rdf:type :JobPosting .
  ?job :jobTitle ?jobTitle .
  ?job :salary ?salary .
  ?job :postedBy ?employer .
  ?employer :name ?employerName .
  ?job :locatedIn :Remote .
  FILTER(?salary >= 80000)
}
ORDER BY DESC(?salary)
```

jobTitle	employerName	salary
"Senior Software Engineer"	"TechCorp Inc."	"130000"^^<http://www.w3.org/2001/XMLSchema#integer>

Figure 8 - Find job seekers with specific qualifications and expected salary

SPARQL query:

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

SELECT ?seekerName ?expSalary
WHERE {
  ?seeker rdf:type :JobSeeker .
  ?seeker :name ?seekerName .
  ?seeker :expectedSalary ?expSalary .
  ?seeker :hasQualification :MastersDegree .
   FILTER(?expSalary  <80000).
}
ORDER BY ?seekerName
```

seekerName	expSalary
"Reese Taylor"	"70000"^^<http://www.w3.org/2001/XMLSchema#integer>

Figure 9 - Find suitable jobs for applicants, based on experience, skills, qualifications, preferred location, and salary

SPARQL query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/jobmatchingontology#>

SELECT ?jobSeeker ?jobPosting ?seekerName ?jobTitle
WHERE {
 # Basic typing
 ?jobSeeker rdf:type :JobSeeker .
 ?jobPosting rdf:type :JobPosting .

 # Get names for readability
 ?jobSeeker :name ?seekerName .
 ?jobPosting :jobTitle ?jobTitle .

 # Skill matching
 ?jobSeeker :hasSkill ?skill .
 ?jobPosting :requiresSkill ?skill .

 # Qualification matching
 ?jobSeeker :hasQualification ?qual .
 ?jobPosting :requiresQualification ?qual .

 # Experience level matching
 ?jobSeeker :hasExperience ?seekerExp .
 ?jobPosting :requiresExperience ?requiredExp .
 ?seekerExp :yearsOfExperience ?seekerYears .
```

jobSeeker	jobPosting	seekerName	jobTitle
JobSeeker1	Job1	"Alex Smith"	"Software Developer"
JobSeeker1	Job1	"Alex Smith"	"Software Developer"
JobSeeker10	Job1	"Sam Rodriguez"	"Software Developer"
JobSeeker10	Job1	"Sam Rodriguez"	"Software Developer"
JobSeeker11	Job1	"Jesse Martinez"	"Software Developer"
JobSeeker11	Job1	"Jesse Martinez"	"Software Developer"
JobSeeker12	Job1	"Drew Anderson"	"Software Developer"
JobSeeker12	Job1	"Drew Anderson"	"Software Developer"
JobSeeker13	Job1	"Blake Wilson"	"Software Developer"
JobSeeker13	Job1	"Blake Wilson"	"Software Developer"
JobSeeker14	Job1	"Reese Taylor"	"Software Developer"
JobSeeker14	Job1	"Reese Taylor"	"Software Developer"
JobSeeker15	Job1	"Finley Moore"	"Software Developer"
JobSeeker15	Job1	"Finley Moore"	"Software Developer"
JobSeeker16	Job1	"Hayden Clark"	"Software Developer"
JobSeeker16	Job1	"Hayden Clark"	"Software Developer"
JobSeeker17	Job1	"Dakota Lewis"	"Software Developer"
JobSeeker17	Job1	"Dakota Lewis"	"Software Developer"
JobSeeker18	Job1	"Skyler King"	"Software Developer"
JobSeeker18	Job1	"Skyler King"	"Software Developer"

Execute

Git: **master**

Reasoner state out of sync with active ontology ☑ Show Inferences

5. Critical analysis

The proposed design transforms job matching from lexical comparison to semantic reasoning, addressing skill misalignment, inefficient matching, and personalization needs. This fundamentally alters knowledge representation from isolated records to interconnected semantic networks, enabling inference beyond explicit matches.

The ontology's strengths derive from its semantic relationship structure that models nuanced connections between job market entities (Tarus et al., 2018). The separation of core concepts from domain implementations also enables consistent reasoning while allowing field-specific specialization. This modular approach facilitates evolution through different combinations of skills and qualifications without structural modification, which is crucial in evolving job markets (Blomqvist, 2014). The system accommodates both mandatory criteria and preferences, with quantifiable attributes enabling precise matching.

The system was evaluated through logical consistency (reasoner checks), expressiveness (complex queries), and adaptability (addition of new concepts). The separation of factual properties from preferences demonstrates how ontological approaches address real-world complexity beyond traditional database schemas.

However, the proposed design has a number of limitations. These include, first of all, the simple design, which lack expressiveness for related skills and transferable competencies, and was not built with scalability or data acquisition automation as priorities. Skills are represented as binary entities rather than graded attributes with proficiency levels, and the system lacks temporal context and distinction between "essential" and "desirable" skills. Moreover, like most OWL ontologies, it cannot model uncertainty despite job matching being inherently uncertain (Lukasiewicz and Straccia, 2008). Addressing these limitations would require more sophisticated constructs incorporating fuzzy logic, temporal reasoning, and complex skill hierarchies alongside NLP techniques (Wimalasuriya and Dou, 2010).

6. Conclusions

In sum, this project demonstrates the application of ontology-based knowledge representation to address limitations in traditional job matching systems, namely how semantic relationships and reasoning capabilities can enhance job matching beyond keyword-based approaches. The ontology successfully models job market entities and their relationships, enabling inference about suitable matches through both defined classes and SPARQL queries, across multiple matching scenarios. While the prototype has limitations in handling uncertainty and skill relationships, it provides a foundation for more sophisticated implementations. Future improvements could tackle the identified limitations through more complex modelling constructs and integration with natural language processing techniques.

References

Blomqvist, E. (2014) 'The use of Semantic Web technologies for decision support–a survey', Semantic Web, 5(3), pp. 177-201.

Cardoso, A., Mourão, F. and Rocha, L. (2021) 'The matching scarcity problem: When recommenders do not connect the edges in recruitment services', Expert Systems with Applications, 175(C), 01 August. Available at: https://doi-org.uniessexlib.idm.oclc.org/10.1016/j.eswa.2021.114764 (Accessed: 13 July 2025).

Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., Mongiello, M. and Mottola, M. (2003) 'A Formal Approach to Ontology-Based Semantic Match of Skills Descriptions', JUCS - Journal of Universal Computer Science, 9(12), pp. 1437–1454. Available at: https://doi.org/10.3217/jucs-009-12-1437.

Debellis, M. (no date) Protégé OWL Tutorial. Available at: https://www.michaeldebellis.com/post/protege-owl-tutorial (Accessed: 11 July 2025).

Fernandez, M., Gomez-Pearez, A. and Juristo, N. (1997) 'Methontology: From Ontological Art Towards Ontological Engineering', in AAAI-97 Spring Symposium on Ontological Engineering. Stanford University, pp. 33-40.

García-Sánchez, F., Martínez-Béjar, R., Contreras, L., Fernández-Breis, J.T. and Castellanos-Nieves, D. (2006) 'An ontology-based intelligent system for recruitment', Expert Systems with Applications, 31(2), pp. 248–263. Available at: https://doi.org/10.1016/j.eswa.2005.09.023.

Gruber, T.R. (1993) 'A translation approach to portable ontology specifications', Knowledge Acquisition, 5(2), pp. 199–220. Available at: https://doi.org/10.1006/knac.1993.1008.

Guarino, N. and Welty, C. (2002) 'Evaluating ontological decisions with OntoClean', Communications of the ACM, 45(2), pp. 61-65. Available at: https://doi-org.uniessexlib.idm.oclc.org/10.1145/503124.503150 (Accessed: 13 July 2025).

Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R. and Wang, H.H. (2006) 'The manchester OWL syntax', in CEUR Workshop Proceedings. Workshop on OWL: Experiences and Directions, OWLED 2006, RWTH Aachen University. Available at: https://research.manchester.ac.uk/en/publications/the-manchester-owl-syntax (Accessed: 11 July 2025).

Lukasiewicz, T. and Straccia, U. (2008) 'Managing uncertainty and vagueness in description logics for the Semantic Web', Journal of Web Semantics, 6(4), pp. 291-308.

Mochol, M., Oldakowski, R. and Heese, R. (2004) 'Ontology based recruitment process'. Available at: https://scispace.com/pdf/ontology-based-recruitment-process-2rpuxlfr08.pdf.

Perez, J., Arenas, M. and Gutierrez, C. (2006) 'Semantics and Complexity of SPARQL'. arXiv. Available at: https://doi.org/10.48550/arXiv.cs/0605124.

Raad, J. and Cruz, C. (2015) 'A Survey on Ontology Evaluation Methods', in Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015). Lisbon, Portugal, pp. 179-186. Available at: https://doi-org.uniessexlib.idm.oclc.org/10.5220/0005591001790186 (Accessed: 13 July 2025).

Studer, R., Benjamins, V.R. and Fensel, D. (1998) 'Knowledge engineering: Principles and methods', Data & Knowledge Engineering, 25(1), pp. 161–197. Available at: https://doi.org/10.1016/S0169-023X(97)00056-6.

Tarus, J.K., Niu, Z. and Mustafa, G. (2018) 'Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning', Artificial Intelligence Review, 50(1), pp. 21-48.

Wimalasuriya, D.C. and Dou, D. (2010) 'Ontology-based information extraction: An introduction and a survey of current approaches', Journal of Information Science, 36(3), pp. 306-323.

Yun, W., Lee, D., Park, C., Kim, H. and Min, O. (2021) 'Knowledge modeling: A survey of processes and techniques', International Journal of Intelligent Systems, 36(4), pp. 1686–1720. Available at: https://doi.org/10.1002/int.22357.