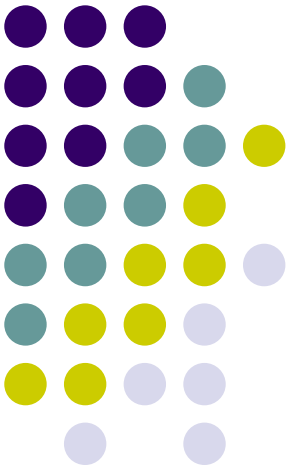
Freie Universität Berlin

# Improving the recruitment process through ontology-based querying

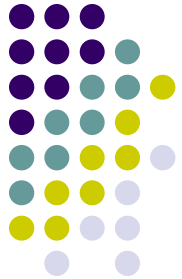
Malgorzata Mochol[1], Holger Wache[2], Lyndon J. B. Nixon[1]
holger@cs.vu.nl

[1]Free University of Berlin
Networked Information Systems
http://ag-nbi.de

[2]Vrije Universiteit Amsterdam
Artificial Intelligence Section
http://www.cs.vu.nl/



# Outline



- Motivation
- Project Context
- e-Recruitment Nowadays
- e-Recruitment Use Case
  - Requirements Analysis
  - Semantic Web-based Prototype
  - Extension of the Semantic Web-based Prototype
- Conclusion & Future Work

# Motivation

- Online Recruitment the main recruitment channel
  - 47% of German internet users (28% of European users) read the online job postings*
  - Over 50% of future employment procurement in Germany is expected to occur onlinE

- Online personnel marketing = cost cutting and efficiency
- Maintenance of an overview of so many portals is a formidable task → visiting every job exchange site next to impossible
  - many websites and online portals financed by publishing fees
  - various business websites
  - portal set up by the state job centre

* “Daten für die Online-Rekrutierung I/2005” wwjGmbH

# Project Context

- **Knowledge Nets** (Wissensnetze)
  - Analysis of typical scenarios for the deployment of Semantic Web technologies
  - Prediction of the economic impact of Semantic Web technologies on e-Business
  - Combination of business and technology-driven analysis

- **European Network of Excellence Knowledge Web**
  - Goal is to achieve technology transfer from academia to industry
  - An Industry Area collects business use cases and promotes Semantic Web based solutions
  - Industry Portal http://knowledgeweb.semanticweb.org/o2i

# Today's e-Recruitment

- Job postings in the form of **free text** using **uncontrolled vocabularies**
- Meta-search engines – search on a **full text basis**
- Open positions published on employer **websites** → meta-job portals collect information from different sites
- **General** & **specialized** search engines as main tool in job search
  - 74% of internet users use various search engines as main tool for information retrieval*
  - 41,1% of internet users use Google*

- Problem: Search engines index job postings imprecisely:
  - Problem to recognize a job posting on the Web
  - Problem to extract relevant keywords (job title, skills, …) using linguistic methods
  - Search results limited in their ability to provide offers that match the precise needs
- Solution: **Semantic annotation** of job postings

*"Online-Rekrutierung I/2005" wwjGmbH

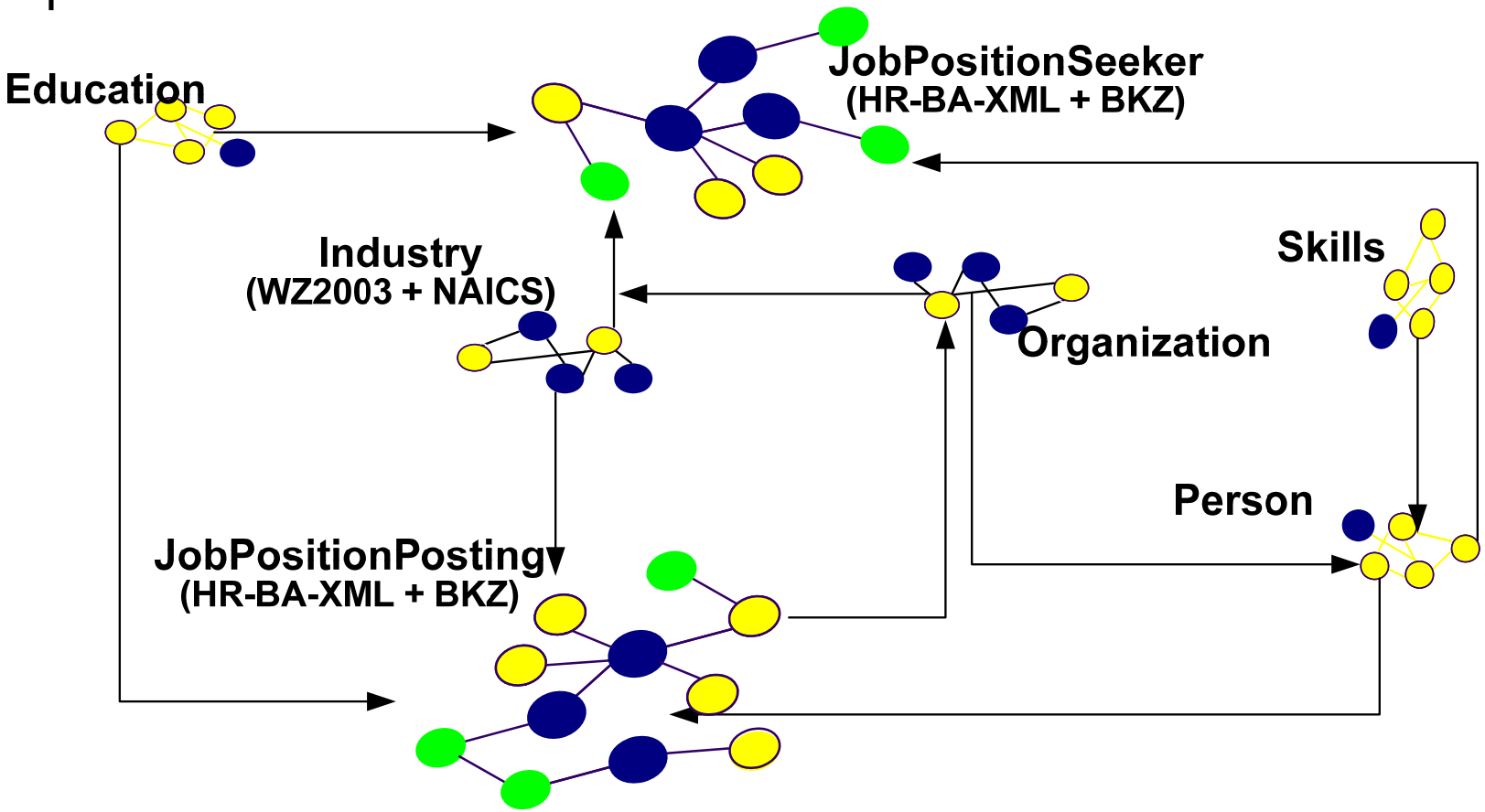
# Use Case-based Requirement Analysis

- We are looking for a person which:
  - has a degree in computer science
  - wants to work in software consulting and development,
  - is an expert in C, Java, PHP, UML, .Net and WindowsNT,
  - has worked for at least 5 years in an industrial and 5 year in a research project,
  - should have experience as project or team manager,
  - should not be older then 25

*"Online-Rekrutierung I/2005" wwjGmbH

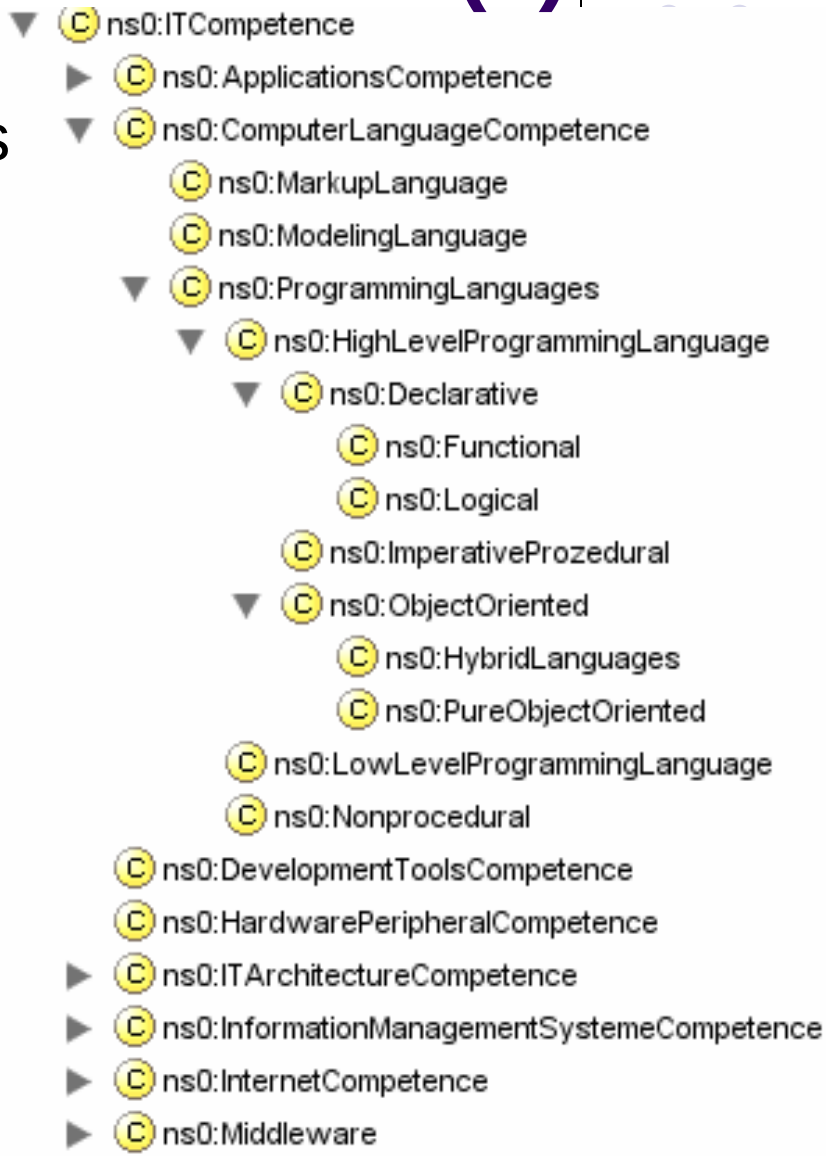
# Semantic Web-based solution (I)

- *Human Resource Ontology* incorporates set of controlled shared vocabularies for employers, job applicants and job portals

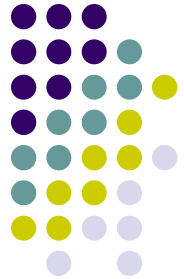


# Semantic Web-based solution (II)

- Semantic matching - combines annotations using controlled vocabularies with background knowledge about the HR domain
- Semantic search engines
  - use the annotation to identify job postings
  - could reliably crawl and index job postings
  - offer semantic matching services to increase the precision of matching of open positions and applications



# Advantages of Semantic Web-based solution



- Employers could use semantic matching algorithms to automate the pre-selection of candidates
- Job seekers could profit by increased transparency in the market

# Semantic Web-based solution (III)

Details zu den Tätigkeiten (88.0% Ähnlichkeit, gewichtet mit 0.2)

software consulting and development

| | Stellenausschreibung | Bewerber | Ähnlichkeit |
|---|---|---|---|
| **Wirtschaftszweig:** | Softwareberatung und -entwicklung | Sonstige Softwareentwicklung | 100.0% |
| **Berufskennziffer:** | Dipl.-Informatiker/in (Uni) | Dipl.-Informatiker/in (FH) - Softwaretechnik | 76.0% |

degree in computer science

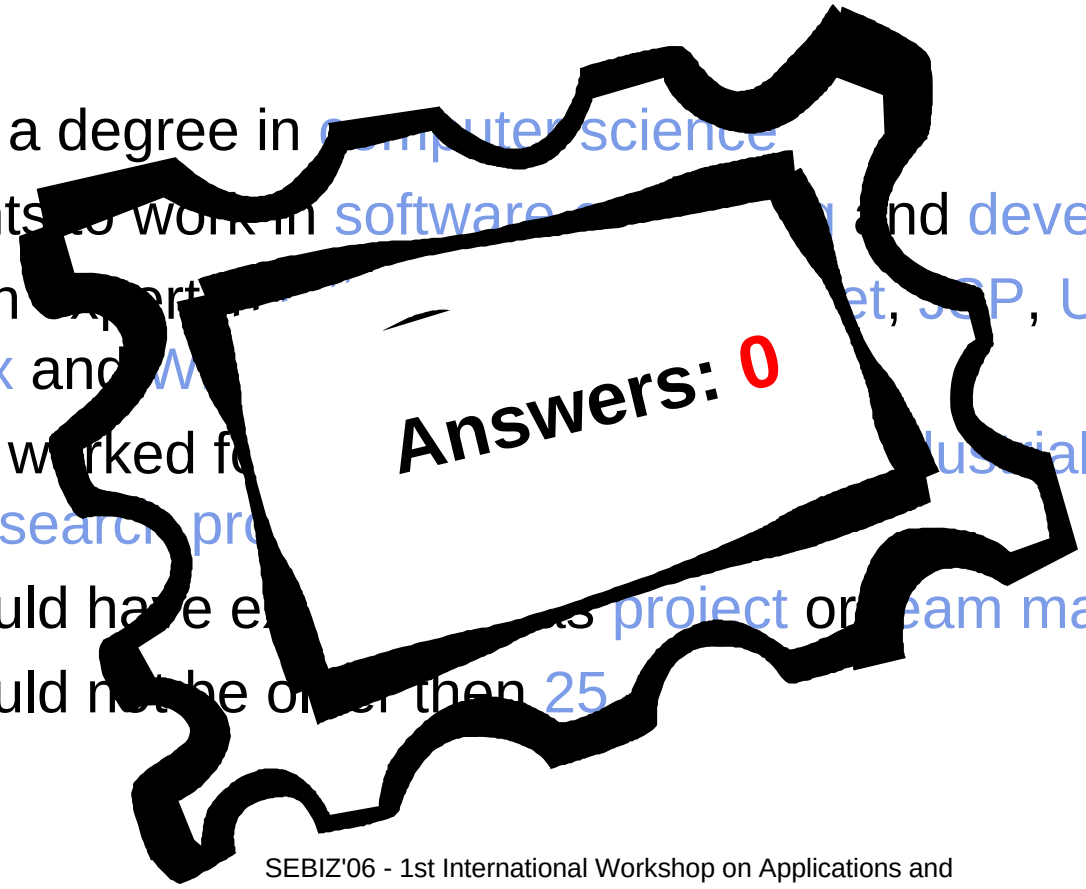
Erforderliche Kompetenzen (90.7% Ähnlichkeit, gewichtet mit 0.6)

| Erforderliche Kompetenzen | Vorhandene Kompetenzen | Ähnlichkeit |
|---|---|---|
| C# (Experte) | Java(Mittlere Kenntnisse) | 89.7% |
| Java(Experte) | Java(Mittlere Kenntnisse) | 92.0% |
| C++ (Experte) | Java(Mittlere Kenntnisse) | 86.1% |
| Unix(Experte) | Linux(Experte) | 92.1% |
| Servlet(Experte) | JSP(Experte) | 92.1% |
| UML(Experte) | UML(Mittlere Kenntnisse) | 92.0% |
| DotNET(Experte) | DCOM(Anfaenger) | 80.1% |
| JSP(Experte) | JSP(Experte) | 100.0% |
| WindowsNT(Experte) | Linux(Experte) | 92.1% |

skills with competence level

# Some (still) open issues

- We are looking for a person which:
  - has a degree in computer science,
  - wants to work in software [illegible] and development,
  - is an expert in [illegible] et, J2EP, UML, .Net , Unix and W[illegible],
  - has worked f[illegible] dustrial and 5 year in a research pr[illegible],
  - should have e[illegible] as project or team manager,
  - should not be older then 25.



# How to relax the query

- We are looking for a person which:
  - has a degree in computer science
  - want[illegible] work in software consulting and development,
  - [illegible]xpert in C#, Java, C++ , Servlet, JSP, UML, .Net , [illegible]nd WindowsNT,
  - has worked for at least 5 years in an industrial and 5 year in a research project,
  - should have experience as project or team manager,
  - should not be older then 25

Drop the degree requirement (3)

Another OO-language would also be okay (1)

3 years may also be enough (2)

need not to have leading experiences but then he should be younger than 21 (4)

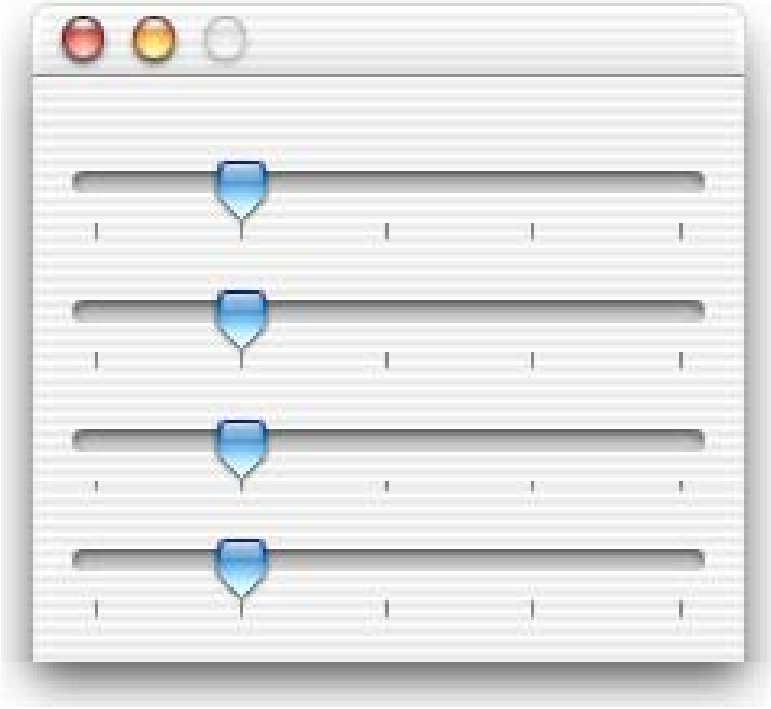
# How to specify these relaxations

Skills

Experiences

Function

Age



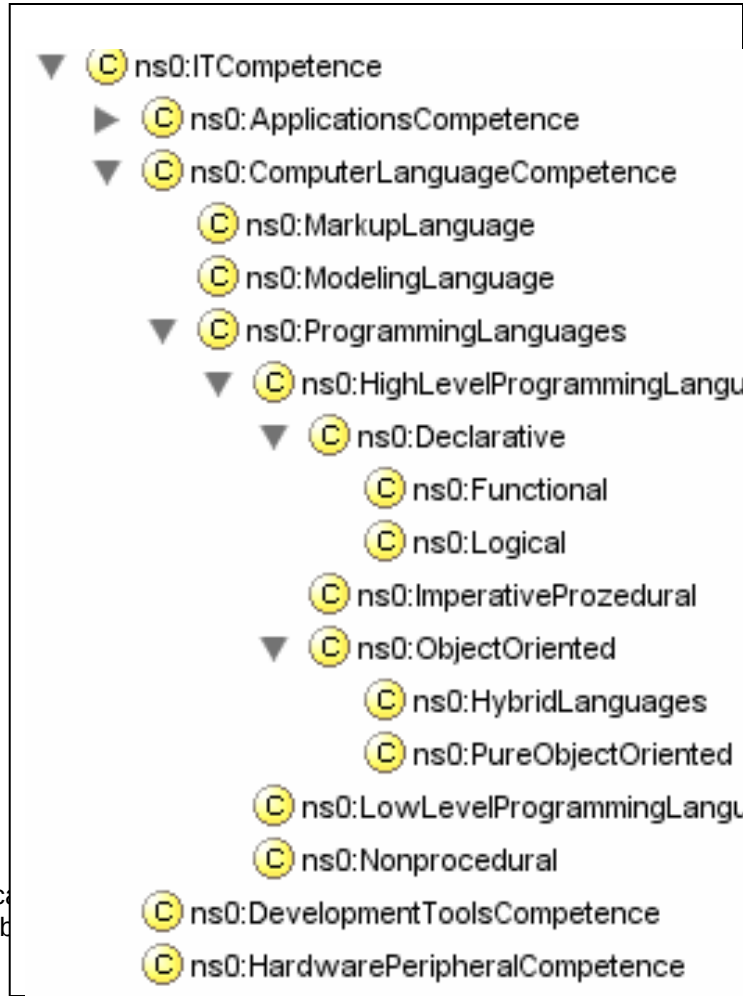
# How to specify these relaxations

**IF** experience = JAVA
**THEN** experience = OO-Language

**Rule 1**

**IF** experience = X
**THEN** experience = Y
**WITH** Y is father of X

**Rule 2**



# Rewriting Rules (I)

When you find all patterns in a query ....

... then substitute the patterns with these replacements ...

**IF** $pattern_1$, $pattern_2$, …
**THEN** $replacement_1$, $replacement_2$, …
**WITH** $condition_1$, $condition_2$, …

... but do the replacements only if the conditions are satisfisfied.

**Very simple but expressive relaxation technique!**

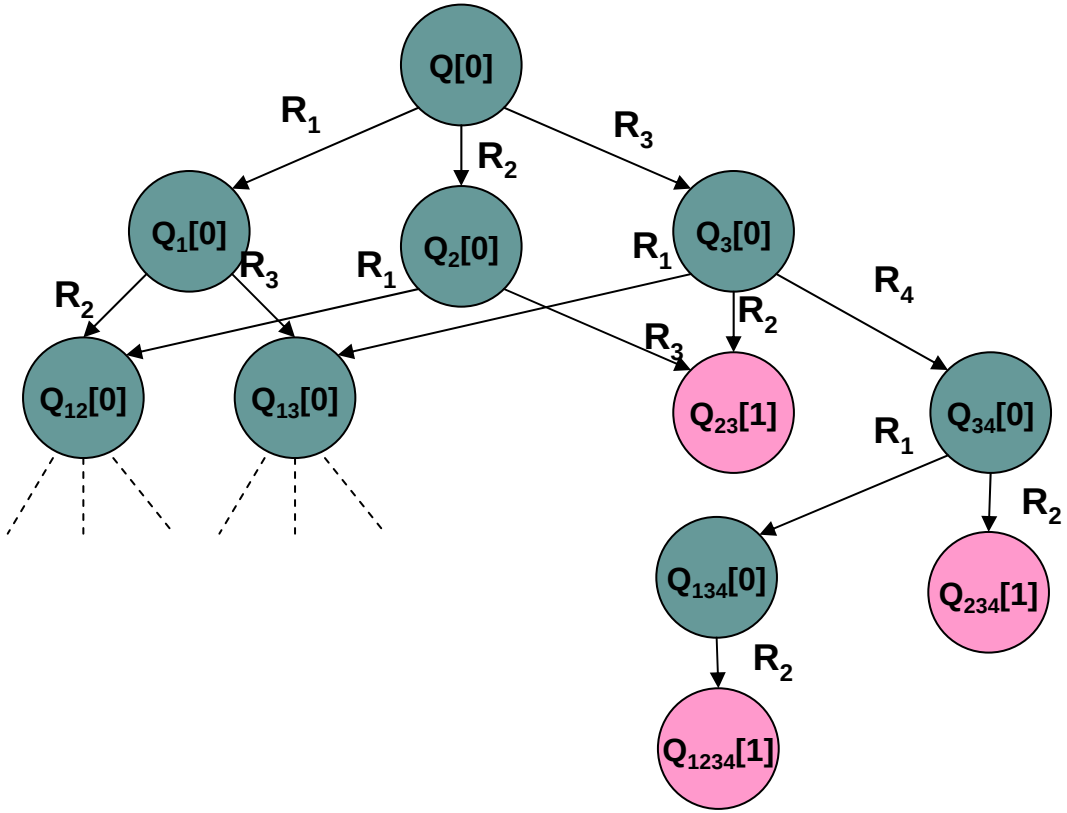
# Replacing OWL queries

```
<owl:Class rdf:ID="Query">
   <rdfs:subClassOf>
     <owl:Class rdf:ID="Perso
   </rdfs:subClassOf>
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:someValuesFrom>
         <owl:Class>
           <owl:intersectionO
             <owl:Class rdf:a
             <owl:Class rdf:a
             <owl:Class rdf:a
             <owl:Class rdf:a
           </owl:intersection
         </owl:Class>
       </owl:someValuesFrom>
       <owl:onProperty>
         <owl:ObjectProperty
       </owl:onProperty>
     </owl:Restriction>
   </rdfs:subClassOf>
   ...
 </owl:Class>
```

**IF** <owl:Class rdf:about="Java"/>
**THEN** <owl:Class rdf:about="PureObjectOrientedLanguages"/>
**WITH** true

**IF**
```
<owl:Restriction>
    <owl:onProperty rdf:resource="#hasDuration"/>
    <owl:someValuesFrom>
        <owl:Class rdf:ID="FiveYearsOrMore"/>
    </owl:someValuesFrom>
</owl:Restriction>
```
**THEN**
```
<owl:Restriction>
    <owl:onProperty rdf:resource="#hasDuration"/>
    <owl:someValuesFrom>
        <owl:Class rdf:ID="TwoYearsOrMore"/>
    </owl:someValuesFrom>
</owl:Restriction>
```
**WITH** true

# Search tree


Q[0]
R1
R2
R3
Q1[0]
Q2[0]
Q3[0]
R3
R1
R1
R2
R4
R2
R3
Q12[0]
Q13[0]
Q23[1]
Q34[0]
R1
R2
Q134[0]
Q234[1]
R2
Q1234[1]

# Advantages and Disadvantages

## Sliders

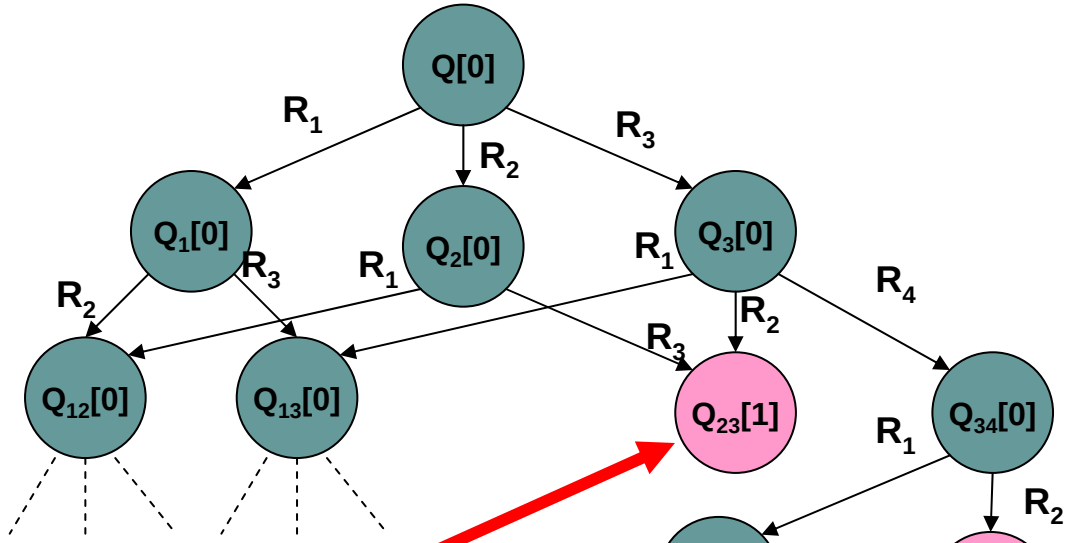
- ✚ "Estimates" the distance between my query and one result (provides a ranking)
- ▬ Sliders are difficult to use; relaxation steps have to be translated in an unnatural way

## Rewriting Rules

- ✚ Easy to formulate the options how to relax a query.
- ▬ Returns a set of unordered results, i.e. the results can't be ranked

**Combine both**

# Proposal for Combination




Apply Slider technology in order to rank
- the clusters and
- the results in one cluster

# Conclusion

- e-Recruitment is a natural application area for Semantic Web techniques
  - Job postings and offerings are annotated with metadata
  - Ability to search with complex and precise queries
- Over specified queries (normal case!) have to be relaxed
  - Sliders are inappropriate for representing relaxation steps but rank all results
  - Rewriting rules more human-oriented but can not rank the results
  - Combination of both most promising