TÉCNICO LISBOA

Bottom-up Ontology for IT Skills

Pedro Miguel Nunes Cunha

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Miguel Leitão Bignolas Mira da Silva
Prof. Rúben Filipe de Sousa Pereira

Examination Committee

Chairperson: Prof. Pedro Tiago Gonçalves Monteiro
Supervisor: Prof. Miguel Leitão Bignolas Mira da Silva
Member of the Committee: Prof. Helena Sofia Andrade Nunes Pereira Pinto

March 2021

Acknowledgments

First of all, I would like to express gratitude to my supervisors, Professor Miguel Mira da Silva and Professor Rúben Pereira, for all the guidance, knowledge shared, and all the time spent, which without a doubt were crucial for the success of this thesis.

Secondly, I also want to thank the company that initially propelled this work, and that despite its premature termination, allowed me to take advantage of its data for the conclusion of this thesis.

To those closest to me, my parents, grandmother, brother, and niece, for all the encouragement and motivation they gave me and especially for the support in a more demanding moment during the development of this thesis, which forced me to suspend the work for a few months. I want to emphasize my tremendous gratitude to my parents for allowing me and giving me all the conditions I needed to reach this stage.

To my girlfriend Matilde, for all her patience, support, motivation, for all the right words at the right time, for all the calls for attention so I wouldn't get distracted by less important things, for reading the entire thesis and finding ways to improve it, an enormous thank you.

I want to end by showing my appreciation to all those who have been part of my journey through the past years at the Instituto Superior Técnico, and especially to those who have contributed positively to help me get to this point.

Abstract

The recruitment process carried out by companies specialized for the purpose has been automated over the years to reduce time and costs while improving its accuracy. Approaches, such as ontologies, have been used to represent skills, position categories and other information necessary for the matching process between candidates and job positions. However, existing ontological developments do not take advantage of the data used by these companies daily, but rather, by the data available from other sources, which may not correctly represent the domain in which the company operates.

Using the Design Science Research Methodology to guide the work and the ”Ontology Development 101” Methodology to guide the development, a bottom-up ontology was developed, with terms gathered from the database (DB) of a recruitment company. The focus was on skills and job categories in the IT field, on the skills hierarchy, on the relationships between skills and job categories, and the relations among skills themself. For an exhaustive evaluation process, four approaches were conducted: Reasoners, Competency Questions, Data-Driven and by Comparing With Other IT Skills Ontology.

The results obtained through the analysis of the evaluation process demonstrate that the developed IT Skills Ontology is indeed a tool that proves to be useful in the recruitment process, more specifically, in the match between candidates with job categories.

Keywords

Ontology; Bottom-up Ontology; IT Skills Ontology; Ontology Development; Ontology Evaluation; IT Recruitment Match

Resumo

O processo de recrutamento levado a cabo por empresas especializadas para o efeito tem sido automatizado ao longo dos anos para reduzir o tempo e os custos, melhorando ao mesmo tempo a sua precisão. Abordagens, tais como ontologias, têm sido utilizadas para representar competências, categorias de cargos e outras informações necessárias para o processo de correspondência entre candidatos e cargos. Contudo, os desenvolvimentos ontológicos existentes não tiram partido dos dados utilizados diariamente por estas empresas, mas sim dos dados disponíveis de outras fontes, que podem não representar correctamente o domínio em que a empresa opera.

Utilizando a metodologia do Design Science Reserch para orientar este trabalho e a metodologia ”Ontology Development 101” para orientar o desenvolvimento em si, foi desenvolvida uma ontologia de bottom-up, com termos recolhidos da base de dados (DB) de uma empresa de recrutamento. O foco foi nas competências e categorias de trabalho na área das IT, na hierarquia de competências, nas relações entre competências e categorias de trabalho, e nas relações entre as competências. Para um processo de avaliação exaustivo, foram executadas quatro abordagens: Reasoners, Competency Questions, Data-Driven e Comparison with other IT Skills Ontology.

Os resultados obtidos através da análise do processo de avaliação da Ontologia de IT Skills, demonstram que a ontologia desenvolvida é de facto uma ferramenta que se revela útil no processo de recrutamento, mais especificamente, na correspondência entre candidatos com categorias e cargos de emprego de IT.

Palavras Chave

Ontologia; Ontologia Bottom-up; Ontologia de IT Skills; Desenvolvimento de Ontologia; Avaliação de Ontologia; Match no Recrutamento de IT

Contents

1 Introduction **1**

1.1 Research Methodology 5

1.2 Document Structure 6

2 Theoretical Background **7**

2.1 Skills 9

2.2 Ontologies Fundamentals 10

2.2.1 What are ontologies? 10

2.2.2 Ontological benefits 11

2.3 Ontology Engineering 12

2.4 Bottom-up Ontology Development 14

2.5 Skills Ontologies 17

3 Research Problem **19**

4 Proposal **23**

4.1 Objective 25

4.2 Ontology Development Method 25

4.3 Proposal Implementation and Development Tools 28

5 First Iteration **31**

5.1 Description 33

5.2 Design and Development 33

5.2.1 Design 33

5.2.2 Development 37

5.3 Demonstration 39

5.4 Evaluation 42

6 Second Iteration **45**

6.1 Description 47

6.2 Design and Development 47

6.3 Demonstration 50
6.4 Evaluation 52
6.4.1 Competency Questions 53
6.4.2 Reasoners 53
6.4.3 Data-Driven with Candidates and Job Advertisements 54
6.4.4 Evaluation Assessment 55
7 Conclusion **57**
7.1 Contributions 59
7.2 Limitations 60
7.3 Future Work 60
A Appendix A **67**

List of Figures

1.1 DSRM process model (adapted from [1]) 6

4.1 Ontology Development 101 Method [2] 26
4.2 Ontology development 101 Method (adapted from [3]) 28

5.1 Class hierarchy for skills 36
5.2 Rule developed to map the skills spreadsheet to the ontology 38
5.3 Ontology obtained from applying the Cellfie plug-in rules to the excel sheets 39
5.4 Definition of the *hasKnowledgeOf* property 40
5.5 Example of the *DataScientist* category 40
5.6 Example of the match between a candidate and a job advertisement 41
5.7 Example of the match between a job advertisement and a category 41
5.8 Example of a candidate matching with two categories 42
5.9 Example of a job advertisement matching no category (1) 42
5.10 Example of a job advertisement matching no category (2) 43
5.11 Example of a candidate matching no category 43

6.1 Representation of the *isFrameworkOf* property 48
6.2 Cellfie rules to map instances to the ontology 49
6.3 Sample of the excel sheet used to map the instances 49
6.4 Example of the usage of the *isFrameworkOf* property 50
6.5 Rules for the *isFrameworkOf* property 50
6.6 Instance of the candidate 9164 51
6.7 Instance of the candidate 9164 without the Python skill 51
6.8 Instance of the candidate 7565 51
6.9 Job example that does not match a category but a candidate match's the job and a category 52
6.10 Rules inferred after compilation 54

A.1 Excel sample 68

List of Tables

2.1 Methodologies for ontology development 13
2.2 Criteria, levels and approaches for evaluation 14

5.1 Glossary of the class hierarchy and categories used in the ontology 35
5.2 Properties 36

6.1 Instances sample 48
6.2 Competency Questions result interpretation 53

1

Introduction

Contents

1.1 Research Methodology 5
1.2 Document Structure 6

Nowadays, we live in a time of technological advances in which one of the main objectives is to make life easier for people. Such simplifications are expected by a generation, entitled as generation Z [4]. This generation presents itself as highly educated, technologically savvy, innovative and creative, and can be characterized as having an interest in new technologies that are easy to use. As the interest in new technologies grows, organizations also look for new forms, methods and tools to simplify tasks and processes.

A vast amount of people, not only from this Z generation, are often looking for jobs. There may be several reasons for that, from finishing college to looking for new challenges. At the same time, companies are looking for persons to fill their needs. To this end, and to help both companies and people in this selection and recruitment process, specialized companies were founded.

The selection and recruitment process of candidates for job positions, conducted by recruitment companies, is done less automatically than what they would possibly want, or could do [5], therefore taking more time and with higher costs. This work usually falls on human resources workers [6].

It is also important to highlight that since the industry is in constant change [7, 8], it becomes more demanding to be an expert, and most of the time, recruiters are not specialists in the field for which they are searching or evaluating [9]. For that reason, they usually follow traditional or ad-hoc techniques to conduct the ”match” between candidates and positions, in which, if the candidate fulfills the requirements, then he or she should be contacted to move forward in the process of recruitment.

The recruitment company that drove this thesis work specializes in the IT field, which means most of the job advertisements (for job positions) and candidates they work with are of the field. The selection process carried out by the company can follow several paths. For this thesis work, we focused on the standard case, in which the candidate using the platform applies for a published job, and a recruitment specialist evaluates the possible match.

These recruitment specialists are someone that sits between the external and internal personal of the company, who possesses a more broad knowledge regarding IT skills (or at least it is expected). They have the function of evaluating the information from both the candidate and the requirements for the job. The result of this assessment is then made available to the company that published the job.

In general, the way the recruitment process, and more specifically the skills assessment, between job advertisement requirements and the candidate's skills needs a unified mechanism. A mechanism that more easily identify what is associated with a competence (from now on referred to as skill) or category in the IT field.

Therefore, there is a need for a solution that can help automate and conduct this process more precisely. Providing recruitment companies the means to do it, even when resorting to people not qualified for it and in addition to this, allowing the unambiguous identification and definition of each concept with which it is necessary to interact during the recruitment process, will help to simplify it. One

solution to implement these changes, fitting with the requirements in need, are ontologies.

The state of the art study conducted for this theses shows a vast amount of ontologies, in the most diverse fields, including in the IT field [10]. The more relevant for our work are the ones that represent the IT domain, and more specifically, the ones that identify, qualify, and categorize IT skills. Although ontologies have been developed specifically for this purpose or even adapted from other objectives or domains, they lack being developed based on existing data (possibly in recruitment companies DB) but rather from other sources [11, 12]. This process of using empirical data for the development of ontologies is formally named bottom-up ontology development.

In this thesis, guided by the Design Science Research Methodology (DSRM), we developed a bottom-up ontology to identify IT skills and categorize them into well-defined categories in the context of IT. The development itself followed the Ontology Development 101 is comprised of seven steps.

Those steps were performed in two iterations. The first one aimed specially to the identification of the ontology purpose, its main components, classes and relations, and the second one aimed to improve the initial ontology after the analyses of the obtaining the results from the evaluations approaches taken at the time.

The ontology content comes from an IT recruitment company's DB, particularly data referring to categories and skills used to describe the candidates and job advertising requirements. There was a need to treat the data, since many of the terms that it was compose of were not related to the field. The carried out ontology evaluation process used five approaches: Reasoners, Competency Questions, Data-Driven, by Comparing With Other IT Skills Ontology and following a Theoretical Criteria/Levels assessment. The obtained results are very encouraging of the ontology's validity as a solution for the identified problem.

The ontology suffered a severe change regarding its use after the beginning of the development. With this change, although it was a process initiated by a specific recruiting company, it is no longer a development for that company. However, this change does not alter or harm the final result in terms of the product produced, since based on the evaluation conducted, it continues to be a viable solution to a problem identified in the domain of IT Skills Ontologies.

With this solution, we aim to serve both recruitment companies and candidates. Companies that will possess a method, possibly more effective than the ones they use, to conduct the skills assessment process (in the recruitment process). For candidates, who with this solution can know which category fits best, based on their skills.

1.1 Research Methodology

Design Science Research Methodology (DSRM) was the methodology chosen to help guide this research work. DSRM provides a process model to conduct researches in the Information Systems field [13]. The goal of DSRM is to produce artifacts to serve as solutions for identified organizational problems. The method is an iterative cycle focused on the development of said artifacts, by getting feedback on the work developed, it allows the researcher to improve the artifact quality. These artifacts can be constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems) [13].

The DSRM process (Figure 1.1) is carried out in six steps:

- **Problem identification and Motivation:** The research problem is defined and the value of a solution is justified, and by that, it motivates both the researcher and the audience to pursue the solution and accept the results. In our thesis it relies on the fact that ontologies developed for IT Skills, don't take advantage of empirical data from recruitment companies;
- **Define the objectives for a solution:** Starting from the problem definition, indicate the objectives of the solution as well as what is possible and likely to be accomplished. We aim to clarify and automatize the matching between candidates and job position;
- **Design and Development:** Creation of the proposed artifact, determining the desired functionality and architecture. Or goal was to develop a bottom-up ontology for IT skills, based on data from a recruitment company's DB;
- **Demonstration:** Demonstrate the use of the artifact to solve one or more instances of the problem, thus proving that the idea works. By using our developed ontology in real scenarios with candidates and job positions available in the company's pool;
- **Evaluation:** Assess how good a solution is the artifact developed in the face of the problem. This activity includes comparing the objectives of the solution with the results obtained from the application of the artifact in the demonstration. We followed five approaches to evaluate our ontology: Comparing it with another IT Skills Ontology, Data-driven evaluation, Reasoners, Theoretical Criteria/Levels assessment and Competency Questions;
- **Communication:** Expose the problem and its importance, the artifact produced utility and novelty, the rigor of its design and effectiveness to researchers and relevant audiences. Our communication will be performed by discussing and publishing this thesis.

The development process following the DSRM recognizes the possibility/need to perform iterations on it to obtain a higher quality artifact. Thus, to develop our artifact, we performed two iterations of the Demonstration and Evaluation phases.

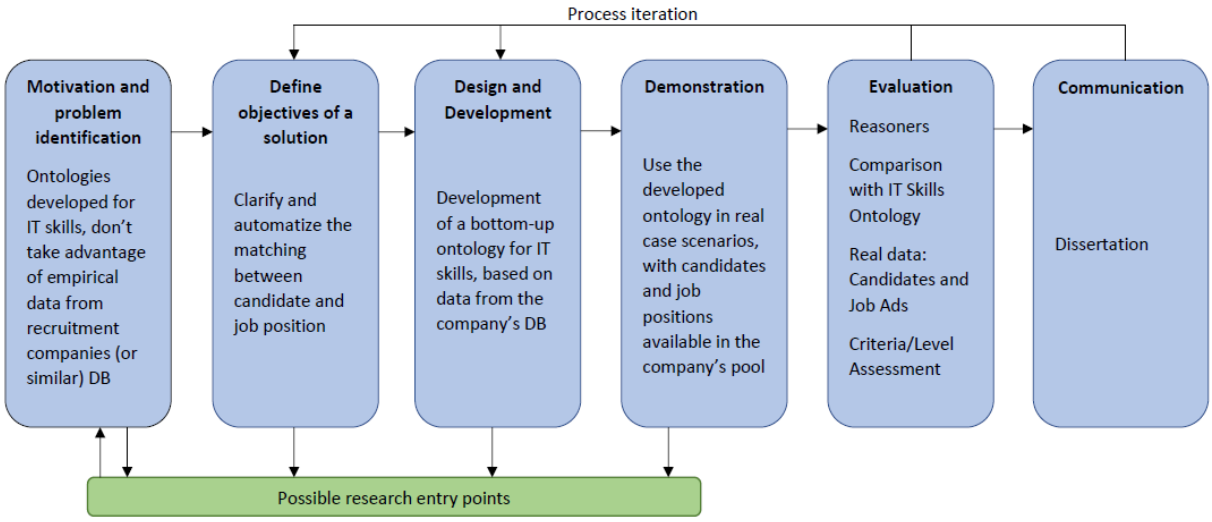

Figure 1.1: DSRM process model (adapted from [1])

1.2 Document Structure

This document follows the structure of the chosen research methodology described in Section 1.1. It starts by introducing the theoretical background that supported this master's thesis in Chapter 2 with the state of the art of Skills in Section 2.1, Ontologies in Section 2.2, 2.3, 2.4, and Skills Ontologies in Section 2.5.

Chapter 3 describes the problem and motivation for this thesis, followed by the proposal description for the artifact in Chapter 4.

Chapters 5 and 6 portray the iterations performed to develop the IT Skills Ontology, comprising both Demonstration and Evaluation steps of the DSRM.

Finally, in Chapter 7, conclusions taken from the research are exposed.

2

Theoretical Background

Contents

2.1 Skills 9
2.2 Ontologies Fundamentals 10
2.3 Ontology Engineering 12
2.4 Bottom-up Ontology Development 14
2.5 Skills Ontologies 17

This Chapter provides the conducted background research regarding skills and ontologies that served as fundamental knowledge throughout this master's thesis. Divided into five sections, were the first one regards the concept of skill, the second one ontology fundamentals, the third one ontology engineering, the fourth concerns the use of empirical data to build ontologies (bottom-up ontology development), and previously developed IT skills ontologies in the fifth and last section.

2.1 Skills

The purpose of this section is to understand what defines the concept **SKILL**, so that we can more accurately identify which terms can be considered skills and understand what is the relation between the concepts of competence and skill.

A skill in its core definition [14], is the ability to do something well, implying understanding or knowledge. The word may be ambiguous since it is not clear either if it indicates mere adequacy or superior, extraordinary ability.

Some literature identifies a small semantic core, composed by the terms "ability", "aptitude", "capability", "competence", "effectiveness" and "skill" [15], defining competence in several fields, as a specialized system of an individual, with the necessary or sufficient requirements to achieve a specific objective.

Another identifies many levels at which competence can be defined [16], for example, describing competence at a functional level as the things that a person who works in a given occupational field should be able to do and be able to demonstrate.

The HR-XML Consortium workgroup proposes the following definition of competence: *"a specific, identifiable, definable, and measurable knowledge, skill, ability and/or other deployment-related characteristics (e.g. attitude, behavior, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context."*.

These competence definitions focus on the existence of some valence that allows an individual to perform a particular function. Although competence may incorporate a skill, it represents more than it. It includes fundamental abilities, behaviors, and knowledge that someone must possess to use a skill.

It is possible to see the skill and competence concepts as synonymous. Since this work focus on skills, it will give more emphasis to that concept and not so much to that of competence. However, a more competency-oriented approach is a possible topic for future research.

Based on this principle, we will use the concept skill to identify something that someone is good at or, at least, can perform it if necessary to achieve some goal. For example, a professional cyclist like Peter Sagan has the skill of riding a bicycle since he knows it and is good at it. An example in the IT context could be a front-end developer programming in Javascript.

Francis Green [17] also reported his understanding of the term skill and its similarity to other terms

used by various fields to refer to the concept. He ends up defining three key points that a skill must have, thus creating the *PES concept of skill*:

- Productive: using a skill at work brings a productive value;
- Expandable: a skill is subject to training and development resulting in an improvement;
- Social: skills are socially determined.

2.2 Ontologies Fundamentals

To portray our research study of ontologies, we will start by describing a little about why to use this type of technology, what an ontology is, and the most direct benefits of their usage.

Communication is fundamental for people, organizations, and software systems [18]. Differences amongst themselves, in how they communicate or how they demonstrate their viewpoints, results in a lack of a shared understanding, leading to less efficient communication. In the IT field, it can lead to difficulties in identifying requirements definitions of a system's specifications.

Therefore, there was a need to establish a unified framework to help reduce or eliminate conceptual and terminological confusion, leading to a shared understanding. To formally represent this shared understanding, this knowledge regarding some domain, a conceptualization can serve as the basis [19]. It is an abstract view of the domain we want to portray, a combination of objects, concepts, and the relations among them. A solution to represent this knowledge in the field of computer science was to use ontologies.

2.2.1 What are ontologies?

Several definitions for the term "ontology" have been given over the years. Each author provides their view over the subject, but the definitions provided are quite alike amongst themselves.

When describing ontologies to be used in the computer science field, Gruber discussed them as *"an explicit specification of a conceptualization. For knowledge-based systems, what "exists" is exactly that which can be represented."* [19]. Until the current days, this is one of the most quoted definitions, and based on it, many others were proposed. Uschold defined ontology as *"Ontology is the term used to refer to the shared understanding of some domain of interest which may be used as a unifying framework to solve the above problems in the above-described manner."* [18].

Corcho et al. identified the above mentioned and other definitions in the literature: *"Ontologies are defined as a formal specification of a shared conceptualization"*; *"a logical theory which gives an explicit, partial account of a conceptualization"* [20, 21]. For van der Vet et al. *"an ontology serves as a partial specification of the knowledge representation to be built in a later stage. The specification is partial*

because it supplies concepts in which states of affairs can be expressed but does not actually specify states of affairs." [22].

With the years passing, new and improved methods and technologies were adapted to store and portray objects, concepts, and relationships between them. One of the most used are *Relational Models*. These are primarily used and associated with Database Management Systems (DBMS) which is software designed to assist in maintaining and utilizing large collections of data [23]. It is the type of system that companies use nowadays to store data. Data that may correspond either to their information, used in the normal operations of the company, information about their clients or partners.

Gruber defined ontology as the level of abstraction of data models comparable to hierarchical and relational models but intended to modeling knowledge about individuals, their attributes, and how they are related to other individuals in the context of DB systems [24].

For this work, we will adopt the definition provided by Uschold, since it accurately describes what we want to achieve, this is, a *"shared understanding of some domain of interest which may be used as a unifying framework to solve the above problems"*. In our case, the domain of IT skills.

2.2.2 Ontological benefits

Ontologies can be beneficial for various purposes. Following the findings in this literature research, Uschold et al. identifies three levels of benefits in using ontologies [18]. At the *communication* level, they enable shared understanding and facilitate communication between people; *Inter-operability* since users need consistent tools that qualify the integration of various software tools and to exchange data; *Systems engineering* as to applications that support the design and development of software systems.

In addition to the usages mentioned before, more recently, two new levels of benefits have been highlighted [25]. *Understanding*, meaning that an ontology can serve as documentation for the conceptualization of a domain; *Reuse*, the existence of an ontology similar to the one intended to be developed, may lead to its use as a basis, but the necessary time to understand and integrate it, must be taken into account to verify if it is worthy.

At the *communication* level, the latter definition focuses more on interactions by means of sharing/sending ontologies between partners as opposing to the first one, which describes it in a more general context inside the organization. As for the *inter-operability* level, the definitions are very similar.

Additionally, doubt arises whether the usage of ontology can bring such significant benefits as some say. Therefore, the costs of implementation must be taken into account and verify if they are justifiable.

2.3 Ontology Engineering

This section aims to portray the research performed on the ontology engineering field. The methodologies (and approaches) for the development of ontologies, the representation of concepts of the ontology domain (and the relations between those concepts), including how the ontology engineer obtains the domain terms and how he constructs the relations, are points of interest of this section.

Initial research has proposed general criteria to guide the development of ontologies [19]. Other researches state-of-art, portray several approaches that have been devised since then for the process of design and development of ontologies [2, 20, 21, 26–28]. Table 2.1 resumes the ones identified in the mentioned studies and each of the phases that compose them. These methodologies are the most used or analyzed by those who study or develop ontologies, and the order in which they appear has been established based on their main characteristics.

Based on the methodologies mentioned in Table 2.1 we identified three phases that are common to the majority. The *determination of the domain and scope for the ontology*, *identification of concepts and relations between them* and *evaluation* are the most present.

For the first phase, the main goal is to identify the domain that the ontology will cover, what will it be used for, the competency questions it is supposed to answer, and who will be using and maintaining the ontology.

The second phase is the identification and definition of those that are the main components of ontologies, the classes, relations, and instances [34]. Next, we explain the importance of each component:

- **Classes:** Are used to represent concepts, normally organized in taxonomies, for example, the taxonomy of programming languages (Java, C, PHP);
- **Relations:** The association of these Classes (representing concepts) amongst themselves is represented with Relations, for example, ”Back-end Developer isProgrammerOf Java”, where Back-end Developer and Java are classes and isProgrammerOf in is the relation between them. These relations usage can also allow to identify the domain or range of some concepts;
- **Instances:** Used to represent elements or individuals in an ontology.

For the identification and definition of the ontology class hierarchy, there are three approaches that can be followed, as suggested in [3, 18]:

- **Top-down:** It starts with by defining the top-level concepts, the most general ones, and then their specialization. However, these can result in a less stable model leading to re-work. For example, starting with by creating the class Programming Languages, then specializing by creating subclasses: JAVA, C#, C. Further categorization can be applied to these subclasses, like versions;

Table 2.1: Methodologies for ontology development

Methodology	Stages	Characteristics
Methontology [29]	1) Plan 2) Specification 3) Conceptualization 4) Formalization 5) Integration 6) Implementation 7) Evaluation 8) Documentation 9) Maintenance	The most mature method. Includes details for re-engineering ontologies but lacks it in terms of pre-developement.
Ontology Development 101 [3]	1) Determine the domain and scope 2) Consider reusing existing ontologies 3) Enumerate important terms 4)Define classes and the class hierarchy 5)Define class properties 6)Define the facets of the properties 7)Create instances	Simplified method for ontology development. Details the pre and development process as does for the project initiation. Lacks integral and post-development processes.
On-To-Knowledge [30]	1) Kick-off 2) Refinement 3) Evaluation 4) Ontology maintenance	Details project management, integral, pre-development and development processes guidelines. Lacks post-development and training details.
Toronto Virtual Enterprise Method (TOVE) [31]	1) Motivating scenarios 2) Informal competency questions 3) Terminology specification 4) Formal competency questions 5) Axiom specification 6) Completeness theorems	High degree of formality. Only provides some detail as do the developement process. Does not include guidelines for re-engineering ontologies or a specified life cycle.
Uschold and King's [18]	1) Identify pupose and scope 2) Building the ontology 3) Evaluation 4) Documentation	Same omissions as the TOVE methodology, being even less detailed.
KACTUS [32]	1)Specification of the application 2)Preliminary ontology design 3)Refinement and structuring	Same omissions as the above method, and has not been used to build many ontologies and applications.
NeOn [33]	Differs from scenario to scenario	Does not imply a rigid workflow, but suggests a variety of pathways. The ontology engineer must identify the one that better fits the development he has to do.

- **Bottom-up:** It starts with the definition of the most specific classes and then grouping them into more general categories. Although this approach results in a very high level of detail, it increases overall effort and the risk of inconsistencies, which can lead to the need for re-work. For example, we start by defining the classes Bootstrap, Foundation, and Skeleton. We then create a common superclass for these classes named Frameworks (**this bottom-up, is associated with the definition of the class hierarchy, should not be confused with bottom-up ontology development**);

- **Middle-out:** It is the combination of the top-down and bottom-up approaches. It starts with defining the more salient concepts followed by generalizing and specializing them appropriately, if necessary, which leads to less re-work and overall effort. For example, we can start with a few top-level concepts such as Data Scientist and some specific concepts, such as MySQL. Then, we can relate them to a middle-level concept, such as DBs.

Finally, the third phase corresponds to the evaluation of the ontology. From the literature, ontology evaluation can be divided into three topics [21, 28, 35–38]. These are identified further specified in Table 2.2.

Table 2.2: Criteria, levels and approaches for evaluation

Criteria	Levels	Approaches
- Consistency - Completeness - Conciseness - Expandability - Sensitiveness - Accuracy - Adaptability - Clarity - Organizational fitness - Computational efficiency	- Lexical, vocabulary, or data layer - Hierarchy or taxonomy - Other semantic relations - Context or application level - Syntactic level - Structure, architecture, design	- The gold standard evaluation - Data-driven evaluation - Evaluation by humans - Application-based evaluation

2.4 Bottom-up Ontology Development

This section regards the bottom-up development of ontologies, one of the core topics of this thesis. The development of an ontology following a bottom-up approach means that the terms of the ontology, and possibly the relationships between them, are obtained based on empirical data. It differs from the previously identified bottom-up since the former aims at building the class hierarchy, regardless of how the terms are acquired. While this bottom-up links to their acquisition, disregarding the class hierarchy.

The process of acquiring the domain terms and relations for the development of an ontology, falls on the developer. He must decide the approach through which he will perform it. This process can be performed under three approaches [25]:

- **Manual:** The ontology is built by hand. It takes more time. However, the involvement of domain experts ensures that the ontology itself and the level of detail is as correct as possible;
- **Semi-automatic:** Computer system that recommends the addition of new ontological components depending on analysis on the domain data. These recommendations may or may not be accepted by the ontology engineer;
- **Automatic:** The ontology is developed by the system, with no human intervention required. The system extracts the ontology components from the data corpus related to the domain and builds the ontology. This approach, despite consuming less time can lead to loss of quality, as it is not guaranteed the guarantee that all the domain concepts obtained are the most correct in terms of relevance.

The state-of-the-art has identified the main techniques and sources to facilitate the ontology learning process, to help in the development of bottom-up ontologies [39]. These sources and techniques are as follows:

1. Reuse of other knowledge-based representations: conceptual data models of DB and application software, such as UML diagrams, ER diagrams, and ORM models;
2. Extraction of types from a DB physical schema and data in DB (i.e., DB reverse engineering) and object-oriented software applications, and least common subsumer and clustering to infer new concepts;
3. Abstractions from or formalisation's of models in textbooks and diagram-based software;
4. Thesauri and other structured vocabularies;
5. Other (semi-)structured data, such as spreadsheets and company product catalogs;
6. Text mining of documents to find candidate terms for concepts and relations;
7. Terminologies, lexicons, and glossaries;
8. Wisdom of the crowds tagging, tagging games, and folksonomies.

The first two only require mapping or transformation to a suitable ontology language; the third is an approach considered as relatively under-explored; fourth and fifth tend to end up as bottom-up development instead of reused due to their core relations; the sixth is related to Natural Language Processing

(NLP) techniques, the search of candidate terms and relations in text documents. The ambiguity of natural language is one of the main problems for not finding immediate matches; seventh relates to documents that don't present much semantics to allow the extraction of terms and relations.

We identified some studies that address and report the development of ontologies, following a bottom-up approach.

Xu, Boyi et al. proposed an ontology developed through the extraction of data from a DB into an XML file [40]. There was the need to define rules for tables and table columns to create them as concepts and properties of concepts were defined. Then experts adjusted the ontology in a top-down approach. The software used for the ontology development was Protégé.

Murphy, et al. followed a bottom-up development from terms and structures within legal documents [41]. It identifies how to perform it, advantages comparing with a top-down approach, and examples. From the reports it is extracted a lot of information from titles, dates, legislative and case references. *"A significant part of legal semantic research has been devoted to top-down ontology development to produce re-usable shared ontologies"*.

Lee, Sunjae, et al. proposed a different use for the bottom-up approach [42]. Instead of merely develop ontologies from scratch, it consists of ontology evolution. Constants evaluations of the ontology and updates performed on it, based on the user's feedback, making the resultant ontology more appropriate to the goal it is supposed to achieve. The addition of concepts or terms divides into five well-defined situations and according to which one it fits, can be or not implemented on the ontology.

There was the proposition to use an ontology as a tool to convert spreadsheets files into XML ones [43]. It follows a semi-automatically approach to develop the ontology from a spreadsheet. There are predefined classes and properties design, leaving the user with the correspondence between the data contained in the spreadsheet to the components present in the ontology. This process relies on an algorithm.

Using a hybrid algorithm approach to identify clusters in XML documents, a bottom-up fuzzy ontology construction was developed [44]. Applies the K-means algorithm as the first step, which then provides initial categories to the k-nn algorithm. Class constraints are defined after the clustering process.

Through the analysis of Security documents, the goal is to develop a bottom-up ontology [45]. This ontology expectation was to aid in the solution to problems presented by a particular organization. The generated ontology bases entirely on the information provided in the security reports. Therefore it was not influenced by standard security ontologies (similar to what we intended to develop) The ontology was then evaluated by comparing it with standard ontologies, with the same intention of usage and scope.

An ontology following both top-down and bottom-up approaches [46] was developed for local research communities. Bottom-up is used to extract terms and relations from papers with NLP. Although, a bottom-up approach is proven to be very helpful in the construction of knowledge management. The

matching of all the keywords, specially unclassified elements or small clusters, turned out to be difficult, so it was considered the top-down approach to produce a reflection on top of the bottom-up classification.

After a conceptual structure defined in a top-down approach, a bottom-up approach by text-mining [47], was used to populate and create an incremental refinement of the conceptual structure. Francesconi, Enrico, et al. concluded that *"bottom-up approaches can support the refining and expanding of existing ontologies by incorporating new knowledge emerging from texts."*. But some cons of the bottom-up approach were also highlighted, like the fact that it *"results in a very high level of detail, difficult the spotting of commonality between related concepts and increases the risk of inconsistencies."*.

Like the work developed in [46], it followed a combination of top-down and bottom-up ontology approaches. It resulted in a three-phase ontology construction: 1) top-down for core domain ontology; 2) NLP with machine learning techniques to extract domain terminology from legal documents; 3) refinement and linking of ontological and lexical layers.

2.5 Skills Ontologies

In this section, we will portray some of the existing developments of ontologies for skills, some of which regarding IT skills.

An ontology-based algorithm, in the form of a graph, was proposed in [48]. The principles of the ontology are that each sub-ontology represents a domain. The nodes represent skills, and directional edges represent the relationships and their relative weight to the existing relationship between the nodes. The process of matchmaking itself is conducted by calculating similarities between the job requirements and the candidates' skills.

To guide students based on their training (acquired skills) and their interests related to the IT context, Nguyen, et al. proposed an ontology of IT skills [49]. It consists of 214 terms selected from the curriculum and job requirements, grouped into specific IT categories. Four of these categories are more generic, encompassing management, problem-solving, soft skills, and natural science.

When associated with a student, these terms are also subject to the assignment of a knowledge level, on a scale of 0 to 9. This means that a student needs to indicate both the skills he possesses and his knowledge level.

Nguyen et al. then proceed to apply the created ontology for the matching between students and job positions. The system had four matchmaking functions, which facilitated the induction of a category of IT for which the student would have a greater predisposition.

These matchmaking functions consider the existence of skills similar to those requested in a job advertisement. That is skills are at the same level of depth, under the same category. An example is the

case of objected oriented programming languages, if the student has the skill of ”programming in C++” then he may have skills such as Java or C.

Michel et al. proposed a human resources ontology [12]. The ontology intention was to be used as a job portal to allow job and candidate posting and also provide support as matchmaking in job seeking.

The Reference Ontology, which is a composition of several ontologies, was developed to help the process of the job-seeking CV and job posting [11]. The whole ontology is composed of 13 sub-ontologies, one of which corresponds to the Skill Ontology. This ontology has two concepts, the Skill concept and the ICT Skill concept. The last is a subclass of the first. For example, the Hardware skill has ICT Skill Hardware programming. It is based on the European Dynamics Skill classification, which has 291 skills.

The research, conducted to obtain the required theoretical knowledge to complete this master's thesis, allowed us to verify that although information of IT Skills Ontologies exist, there is none regarding IT Skills Ontologies, developed following a transparent empirical bottom-up approach.

When our work was already in the ontology development phase, more specifically in the first phase of the evaluation, we became aware of a new proposal for an IT Skills Ontology [50] developed in the same domain as our own. It is an ontology developed for the IT recruitment area, which eventually was adopted by the company that motivated this work. It is more of a literature-focused approach, in which the terms that constitute the ontology were acquired through an extensive research of skills for the various IT categories. We took advantage of this ontology to evaluate ours and thus improve the final quality in terms of structure and knowledge generated.

3

Research Problem

This section is the concretization of the first step of the DSRM ”Problem identification and Motivation”, where the objective is to identify the research problem.

Many companies choose to delegate the recruitment process to specialized companies for this purpose [51]. These recruiting companies usually have a large pool of candidates, according to their field(s) of expertise, to facilitate the process. Meaning that, according to job requirements provided by the client company, they do the filtering and preliminary interviews from the candidates within their candidate pool. So the company itself only enters the process in a more advanced stage, with fewer and better candidates for screening.

Recruitment companies are paid a fee for this service, which often corresponds to a percentage of the job's annual salary [51]. But this only happens if the candidate is selected for that job. Meaning that even if the recruitment company provides 100 candidates, if neither of these candidates is the right one for the job position, the recruitment company will not receive anything.

The more precise the candidate filtering process is, for the first interviews and later sending the information to the company, the better may be the outcome. Another factor to take into account is the time. The less time it takes to complete the candidate selection process, ensuring accuracy in the selection, the faster they can move on to the next advertisement job. Therefore, creating a higher probability of generating profit for the company.

To achieve this goal, these recruitment companies need means, which has been the interest of several researchers and organizations [52]. More automatized and accurate methods or tools than what they possess, easy to use from someone inside the company instead of paying curators to do it (as in the company that motivated this work) are the goal. Otherwise, they may miss ideal candidates and keep taking too long, leading to money loss.

Ontology-based techniques are one of the matchmaking approaches [52], between candidates and job positions, identified in the literature and having already been considered, and in some cases applied. These ontologies foundations are skills, for example, the match between a candidate and job position through the skills that each one has.

For the specific case of the recruitment company that motivated this work, there is a need for an ontology directed to the IT field. An ontology as those described in Section 2.5.

However, those ontologies are generic, their terms, the skills that they identified are based on literature and information accessible on the internet and from various other sources. Thus, not contemplating the specific domain in which the company performs its work.

The categories and skills used to define a job advertisement or a candidate are stored in specific tables in the recruitment company's DB. Therefore, a future candidate to make a new registration or a company to publish a new job advertisement, count on the help of lists of categories and skills provided by the company. In the same way it facilitates this process, since there is no need to fill this information

with text, it can also help in the process of ontology development.

With this information (containing around 700 terms) stored in the companies' DB, it is possible to develop an ontology in a bottom-up approach, differentiating it from other ontologies. Furthermore, with this approach, we will be capable to represent a more accurate domain in which IT recruitment companies operate. As far as it was possible to ascertain, in previous research, there were no developments of bottom-up ontologies for IT skills.

Therefore, the problem identified in this research is the **lack of ontologies for IT skills, developed taking advantage of existing empirical information in the recruitment companies themselves to better represent the domain in which they work.**

4

Proposal

Contents

4.1 Objective 25

4.2 Ontology Development Method 25

4.3 Proposal Implementation and Development Tools 28

This Chapter corresponds to the second step of the DSRM, ”Define the objectives for a solution”. We explain our proposal to the problem identified in Chapter 3, and provide a detailed description of how we implement it.

4.1 Objective

The main objective of this thesis work's development is the creation of an ontology for IT skills following an empirical bottom-up approach. An ontology which goal is to facilitate the matching process of a candidate to a category or a specific job position.

To answer the problem described in Chapter 3, we propose the development of an artifact to meet the needs of recruitment companies. The artifact consisting of an ontology, built following a bottom-up approach to acquire the terms regarding IT skills used for the ontological learning. The terms used for the ontology construction came from the DB of a recruitment company, especially terms related to IT skills and job positions.

4.2 Ontology Development Method

As a guide to the ontology development, we followed the Ontology Development 101 methodology [2,3]. The choice of this methodology was due in the first instance to the description of its application is made using the same software chosen for the development of our ontology. Also, the simplicity and clarity in applying the various steps compared to other methodologies was an important fact since it was the first contact with the development of ontologies. The ontology in the field of Islamic knowledge, focusing on the Solat [53], is an example of another development, aside from the one provided by Noy et al. [3], that followed this methodology, which helped us as a real example. The methodology process can be better visualized in the Figure 4.1.

This methodology, as previously mentioned, consists of seven steps [3], better described next:

1) Determine the domain and scope

The first step consists of defining the domain and scope, answering some predefined questions of the methodology: a) What is the domain that the ontology will cover?, b) For what we are going to use the ontology?, c) For what types of questions the information in the ontology should provide answers? and d) Who will use and maintain the ontology?. The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model.

In our work, it is portrayed the domain of IT skills and IT job categories. Recruitment companies can use this ontology to facilitate the selection and recruitment process of candidates for job positions,

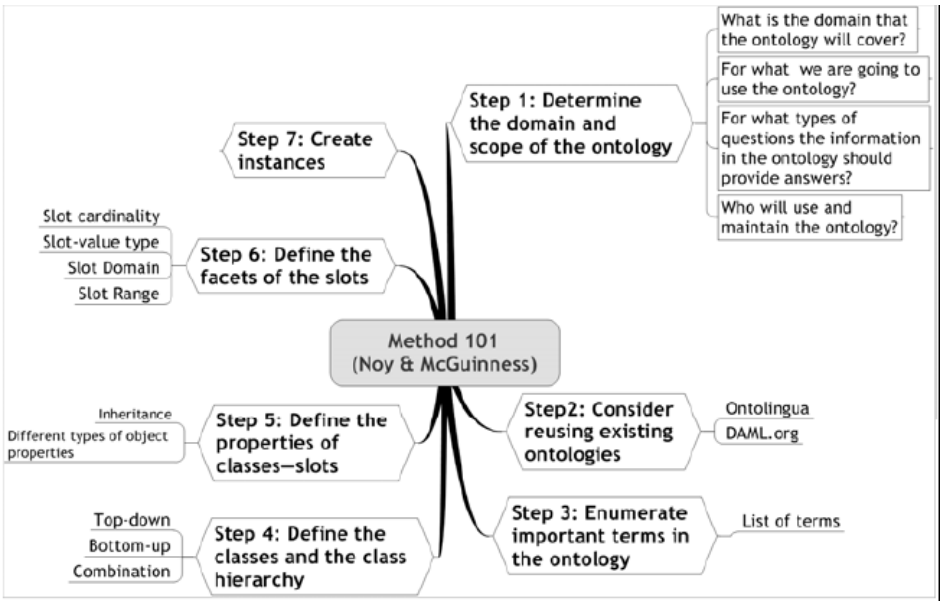

Figure 4.1: Ontology Development 101 Method [2]

based on the candidate's skills and the requirements for a given job. The maintenance process will be the responsibility of the company using the ontology. Competence questions were defined to help determine the ontology scope:

1. What are the best suited category for a candidate with the skills PHP, HTML and Javascript?
2. Which skills one should to have, to be a good Front-end developer?
3. Java is a good to have skill for which categories?
4. Is Bootstrap a good complementary skill for HTML?
5. What are the skills shared by a Front-end developer and a Data Scientist?
6. What are the programming frameworks associated with Python?
7. Can the ontology provide match's between candidates and jobs advertisements?
8. Can the ontology provide match's between jobs advertisements and categories?
9. Can the ontology provide match's between candidates and categories?
10. Can the defined properties provide inferred knowledge ?

2) Consider reusing existing ontologies

In this step, we should consider using ontologies developed by other persons, to the same end of ours or through the refinement and extension for our particular domain.

Since our problem focuses on the lack of developed ontologies, using the existing data in recruitment companies as an advantage, we did not reuse an existing ontology that followed the same development process as our own, with a methodology specifically for ontologies. Nevertheless, the data used for the terms that later became our classes came from a database since that in its nature is already an ontology, in which the data there portrayed follows a defined structure and provides some understanding regarding the domain of the contained information, we could also say that in fact, we do reuse an ontology.

Also, an already existing ontology was used as a form of evaluation to compare to our ontology.

3) Enumerate important terms

It is at this stage that the acquisition of terms to be used in the ontology takes place. We followed a manual approach to gather the terms. Initially, they were grouped in an untreated list, coming from the company's DB, through SQL queries. At this phase, we did not pay attention to the existing relationships between them or even if there is an overlap of concepts.

The following two steps are closely intertwined and are of the most importance for the ontology-design process.

4)Define classes and the class hierarchy

From the list obtained in step 3, we started by selecting the terms that describe skills that fit the PES concept of skill, assessing their utility to the ontology, and making them classes in our ontology. These classes were then be organized into a hierarchical taxonomy, which is the definition of superclasses, classes, and subclasses.

As we saw earlier in the research, there are three approaches to develop this hierarchy: top-down, bottom-up, and the combination of both (middle-out). For the *IT Skills hierarchy*, we followed a bottom-up approach, taking into consideration the type of skills that we are dealing with to better organize them. As for the *job positions category hierarchy*, we only define the existing categories, not further detailing them.

Classes for the representation of candidates and job advertisements, will also be created.

5)Define class properties

Classes alone are not enough to clearly define the ontology nor to answer the competency questions written in step 1. Therefore, after the class definition, we must describe the internal structure of the concepts. At this point, our focus was primarily on the defining relationships between classes representing skills and categories but also among skills. Like in the hierarchy development, when we consider the skill types to create the best set of properties, to define the existing relations.

For example, the category *Front-end developer* is expected to have something similar to *"isProgrammerOf Javascript"*, which states that for someone to be a front-end developer having the Javascript programming language skill is a good thing.

6)Define the facets of the properties

Properties can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features the property can take. For our ontology, we used numeric values to define a minimum number of associations with a specific property, for example, to be a *Back-end Developer* one must possess at least two skills of the *"Programming_Languages_Type"*.

Allowed classes for slots of type Instance are often called a range of a property. For example, a possible class named *Programming Languages* is the range of the *"isProgrammerOf"* property.

The classes to which a property is attached is called the domain of the property. So the *Programming Languages* class has as domain the entire set of job categories such as the *Frontend developer*.

7)Create instances

The last step is creating individual instances of classes in the hierarchy and defining rules to be applied to them when reasoning the ontology. The process of defining instances is divided into three parts: (1) choosing a class, (2) creating an individual instance of that class, and (3) filling in the property assertion values. For example, we defined the instance for a candidate and specify the skills he possesses, finding out the categories he matches to. Furthermore, the same can happen between candidates and job advertisements.

4.3 Proposal Implementation and Development Tools

The development of an ontology is iterative. So, at the end of each phase, it is likely that it will be necessary to go back some steps, to improve the quality of the final ontology. Figure 4.2 gives perspective to the kind of iterations that we will probably face during the entire ontology development.

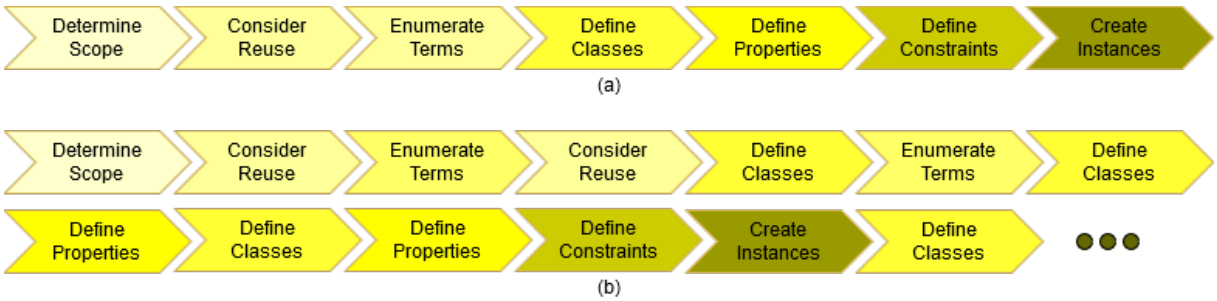

Figure 4.2: Ontology development 101 Method (adapted from [3])

The data was obtained from the company's DB, exporting the results from SQL queries to CSV files. Since the DB is a PostgreSQL one, using the pgAdmin 4 software platform to access it and obtain the data.

The ontology was developed using the OWL language, that was generated by the tool used for the development, the Protégé. It is the most used application for the development of ontologies, being also the one that has more supporting documentation, which was be fundamental for the success of our ontology.

Another fact that leads us to use this software is one of its plugins. Cellfie allowed us to import of data from excel files. By using simple mapping rules, it is possible to direct each of the records present in the files to the desired class.

5

First Iteration

Contents

5.1 Description 33
5.2 Design and Development 33
5.3 Demonstration 39
5.4 Evaluation 42

5.1 Description

As stated before, the DSRM goal is to guide the development and evaluation of an IT artifact to see if it solves the identified research problem.

At one point of our artifact development, the ontology for IT Skills, we figured that the final ontology would benefit from the evaluations assessment. Therefore, to obtain the better result possible, two iterations of the DSRM process were performed as described in section 1.1.

This Chapter portrays the first iteration of the Design and Development, Demonstration, and Evaluation steps of the DSRM. It starts by presenting the initial development and demonstration of the obtained ontology, followed by the process evaluation, which led to the conclusion of the necessity for a second iteration.

5.2 Design and Development

This Section describes the DSRM's Design and Development step and intends to detail the Ontology Development 101 methodology steps taken to implement the proposal defined in Chapter 4. Initially, we design the entire process, filtered all the terms, and specified the properties set. Then we went to the development itself, where everything designed and obtained in the initial phase was represented in the ontology.

5.2.1 Design

1) Determine the domain and scope

For the first step, our domain was defined by the research topic of IT Skills and the scope by the competency questions we want to answer, already identified in Chapter 4.

2) Consider the reuse of existing ontologies

This thesis aims the lacking problem of bottom-up ontologies created from empirical data. Therefore, it would not make sense to reuse existing IT Skills Ontologies, at least those identified in the reasearch, that followed the same development process as our own. But, since a database is in its nature an ontology, in which the data there portrayed follows a defined structure and provides some understanding regarding the domain of the contained information, we could say that in fact, we do reuse an ontology.

Nevertheless, We still used an existing IT Skills Ontology as a mean of comparison with our own, as described in section 5.4.

3) Enumerate important terms

This was when the ontology development work really started, so to speak. We started by installing software that would allow the manipulation and analysis of the data present in the company's DB. Since it is encoded in the PostgreSQL language, it was necessary to use the pgAdmin software development platform, which is used by the company itself.

After becoming familiar with the structure of the DB and with PostgreSQL, we moved on to the identification of which tables would be useful for the development of ontology.

We carried out, together with company employees, a survey of which tables contained information related to skills and categories. We concluded that only four tables would be of value to our research:

- One of the data referring to the categories;
- One of the skills (also relates the skills with the candidates, but, at this moment, we are only interested in knowing which skills exist);
- One that relates skills with categories (it does not contain all possible relationships, only a small portion);
- One of descriptions for the codes used to reference skills and categories in all the previous ones.

From here, we research, within the various ways to develop bottom-up ontologies described in Sections 2.2 and 2.3, tools or methods that would allow the conversion of data directly from the format of a DB to an ontology.

In this field, researches previously carried out, with a view to the development of such tools or methods, do not provide the obtained solution. Others that make it available are old software, which needs to run on top of old operating systems or previous complementary software versions, like java, which created conflicts when installed.

Due to the difficulty of software compatibility and the fact that some of this software makes a complete transformation of the DB, whereas in our case we are only interested in a small portion (four) of the 150+ tables that make up the entire system, we decided to take another approach.

We came across a plug-in, called **Cellfie**, present in the Protégé software. This plug-in allows the mapping of data from data sheets (the traditional Excel sheets) in the different ontology classes. Since the pgAdmin platform allows for the extraction of query results in CSV format, we decided to follow this path.

To learn how to work with Cellfie, we followed a tutorial [1], which proved to be very accessible and able to be applied to our work. The procedure consists of three steps: 1) Obtaining an excel sheet (or similar) with the domain data; 2) creation of the ontology classes in Protégé; 3) Definition of the rules for the population of the ontology. The way we conducted this process is explained in Section 5.2.2.

[1] https://github.com/protegeproject/cellfie-plugin/wiki/Grocery-Tutorial

Having the plan outlined, we moved on to the terms acquisition, using a semi-automatic approach to gather them from the DB. Initially were grouped into three files, coming directly from the company's DB through SQL queries. We treated the terms individually, so we did not look to the existing relationships.

- One with all the skills - 761 terms, analyzed individually following the *PES concept of skill* to assess if they were useful or not. At this point, we eliminated duplicates (i.e. "JIRA" and "jira"); non-valid IT terms(i.e. "veterinary"); those that were a category (i.e. "Backend Development"); some were modified to create a more uniform collection (i.e. "Sharepoint" became "Microsoft Sharepoint"); and cases were the terms corresponded to organizations/companies (i.e. "SAP Solutions"). Throughout this process, we created little descriptions of each valid one to help in the process of relating them with categories;
- One with all the categories - 16 terms, among them we found three that are very borderline. They are not entirely related to IT, but since they portrayed the vision of the company at a given point, we decided to keep them;
- One with all the existing relations between skills and categories - 126 relations that would prove useful in the next steps.

4)Define classes and the class hierarchy

From the lists of terms obtained in the previous step, we created a main excel sheet, where lines corresponded to skills and columns to categories and the position in the hierarchy that a given skill had. We started by designing a skill class hierarchy. In addition to the help it provided allowing for a better organization of skills, it also helped to easily identify relationships between the classes in the next step of the methodology. The resulting hierarchy can be seen in Figure 5.1 and Table 5.1 contains the definition of some classes of skills and categories.

Table 5.1: Glossary of the class hierarchy and categories used in the ontology

Term	Description
Programming_Languages_Type	Programming languages
Software_Development_Type	Concepts related with the development of software
Video_Image_ProcessingT_Type	Tools used for the processing of video and images
Back-end Developer	Job of developing the back end of a website or applications
QA_Testing	Quality Assurance and Testing
UX_UXDesigner	Job of developing the user interface with focus on the user experience and interactions

Figure 5.1: Class hierarchy for skills

As said in the description of this step in Chapter 4, the final result of the class hierarchy was obtained by taking into consideration the skills from the database that fit with our domain and with the PES skill concept. Many of them represent programming languages, hence the Programming_Language_type. Other regard concepts relate with knowledge that one may possess, as is the case of Machine Learning, that fall into the Knowledge_type, and more specifically, the DataK_type.

5)Define class properties

Regarding the relations between skills and categories, we used the description previously made for each skill and conducted an online search to identify with which category or categories that skill associates. We also added a new column to the excel sheet, to portrait the relation between skills and categories. Table 5.2 shows the properties set that was defined. Some of those properties were introduced to the ontology in this first iteration, others in the second one.

Table 5.2: Properties

Relation Name	Domain	Ranges	Introduced to the ontology
uses	Categories	Skills	First Iteration
hasKnowledgeOf	Categories	Knowledge_type	First Iteration
isProgrammerOf	Categories	Programming_ Language_type	First Iteration
isFrameworkOf	Frameworks_type	Programming_ Language_type	Second Iteration

Their usage focus on relating categories and skills, and in further steps candidates and job advertisements, and are as follows:

- Uses - relates categories with skills that represent development tools and frameworks, for example,

a *Front-end Developer uses Bootstrap*;

- hasKnowledgeOf - relates categories with skills that represent knowledge in the IT area, for example, *Data Scientist hasKnowledgeOf Data Mining*;

- isProgrammerOf - relates categories with skills that represent programming languages, for example, *Back-end Developer isProgrammerOf Java*;

6)Define the facets of the properties

In this step, we started by defining the properties domain and range, as shown in Table 5.2. The domain regards the classes described by the property, meaning that the any category class is the domain of the *hasKnowledgeOf* property. As for range, it represents the allowed classes for a property, which for *hasKnowledgeOf*, it is the entire set of skills from the "Knowledge_type.

Another characteristic that we associated with the ontology properties was the cardinality. When defining the skills related to one category, minimum values for those properties were defined, which help to better match candidates with categories (i.e. to be a *Front-end Developer* one must possess a minimum of 2 Programming Languages).

5.2.2 Development

With the terms corresponding to skills and categories, the existing relations among them, and the hierarchy of skills already identified, we went on and started to map it in the ontology. We started by creating a new ontology in Protégé, with two top classes: Categories and Skills. Cellfie was used here to perform the mapping of concepts from the excel sheet to the ontology in the Protégé. The following steps were performed:

- Obtaining an excel sheet with the domain data: We divided the identified terms as skills, categories, and the ones for the hierarchy of classes in three sheets, one to populate the *Categories* class with categories, a second one for the population of the *Skills* class with the identified hierarchy, and the third to populate the resulting *Skills* class hierarchy with skills in their respective position in the hierarchy. A sample of the excel sheet can be seen in Figure A.1.

- Definition of the transformation rules: The purpose of the transformation rules is to map the information of the excel sheets to the ontology. Therefore, we come up with three rules one to map each excel sheet into the ontology. The one used to map terms identified as skills into the ontology, and more specifically, into the identified positions in the hierarchy, is portrait in Figure 5.2.

  We defined that each term of column A (the entire set of skills) was to be created as a class and to be a subclass of the term in column B (hierarchy position of the column A term), which required

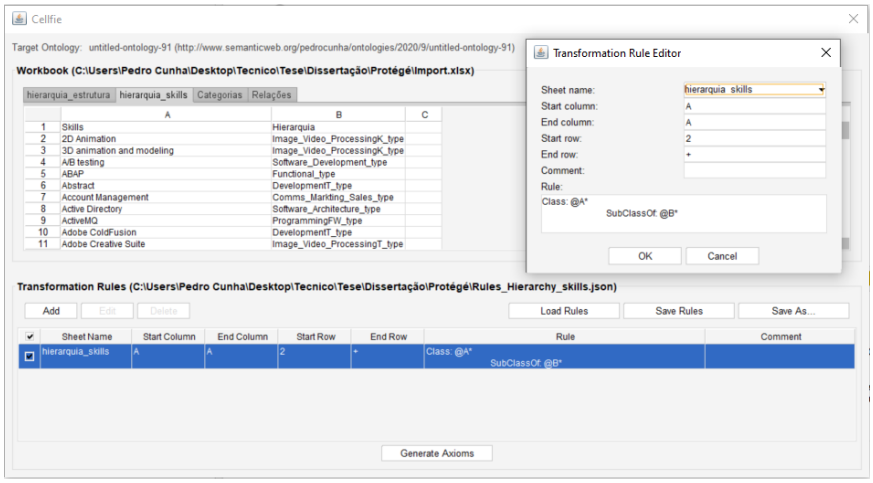

Figure 5.2: Rule developed to map the skills spreadsheet to the ontology

the prior mapping of the class hierarchy. The result of applying these three mapping rules to the excel sheets is showed in Figure 5.3.

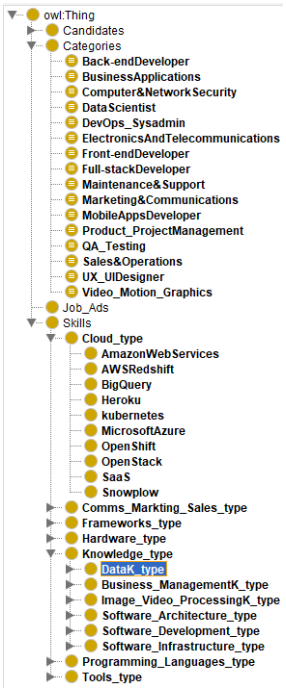

Figure 5.3: Ontology obtained from applying the Cellfie plug-in rules to the excel sheets

At this point of the development process, we found limitations of the Cellfie plug-in since it wouldn't allow the mapping of properties that we had identified nor the identified relationships between classes (skills and categories). We could only map the classes. Therefore, there was a need to perform a more manual process, more time-consuming, in which we manually inserted each property and relation.

To start the process, and similar to what was made for the classes, we created two top properties: *hasSkill* and *isSkillOf*, in which one is the inverse of the other one. Under *hasSkill* we created the properties present in Table 5.2 related with the first iteration, as Figure 5.4 shows, and in *isSkillOf* the corresponding inverse properties.

Although it was a very manual process, the fact that the relations already were defined on the excel sheet allowed to simplify it and even to perform minor corrections, from modifying how a given skill relates with a category to identify a couple of relations that were not described yet or even exclude some. Figure 5.5 portrays an example of how relations of a category were defined in Protégé.

5.3 Demonstration

This section covers the Demonstration step of the DSRM, where for the exact purpose of demonstrating the feasibility of the developed solution, data from candidates and job advertisements sourced from the company's DB will be used.

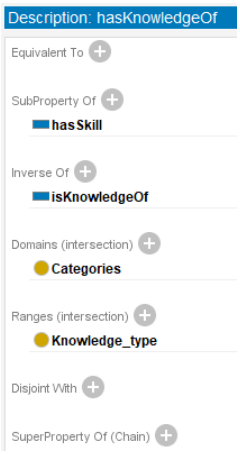

Figure 5.4: Definition of the *hasKnowledgeOf* property

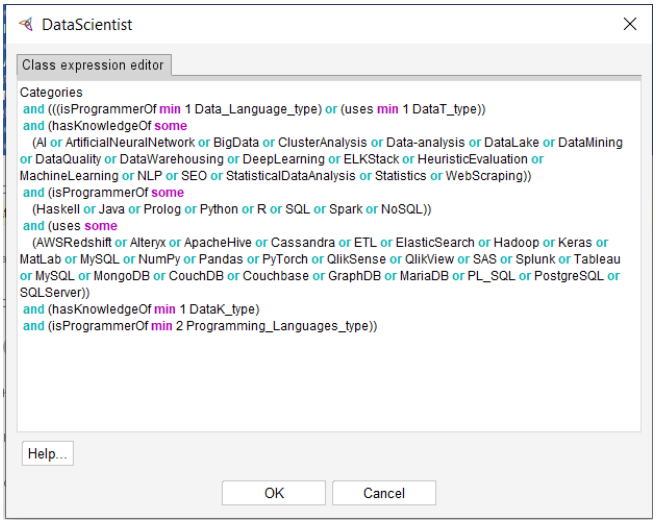

Figure 5.5: Example of the *DataScientist* category

Data from the candidates and job advertisements was obtained by performing a new search to the company DB. The information that we retrieved for the candidates was ids and skills, and for the job advertisements, we retrieved ids and the skills posted as requirements.

Again, similar to what was the analysis made to find out which tables were useful for the acquisition of terms for skills and categories, we study the DB and concluded which were the tables useful to our needs:

- One with all the candidates;
- One that related the candidates and skills that they possess;
- One with all the required job advertisement information;
- One table of descriptions for the codes used to reference skills in all the previous ones;

To extract candidates and job advertisements, we initially limited the search to a minimum of four skills and no more than 10, but this upper limit was slightly increased due to the small number of matches between applicants and jobs. This condition filtered those with few skills and those with many. The ones with few skills most likely would not match any category, while those with too many skills could be a match to more than two or three categories, posing more demanding to identify the best one.

We extracted this information to two excel sheets to work offline, not needing to search every time to add one candidate or job to the ontology. Similar to what happened in the case of Skills and Categories, in this case, two top classes were added to the ontology, *Candidates* and *Job_Ads*, under which we manually cr eated classes to represent each candidate and job advertisement.

The obtained results, after the ontology had been classified and inferred by the reasoner, can be seen in Figure 5.6, which portrays the match between a candidate and a job advertisement, based on the skills that the candidate possesses and the ones required for that specific job.

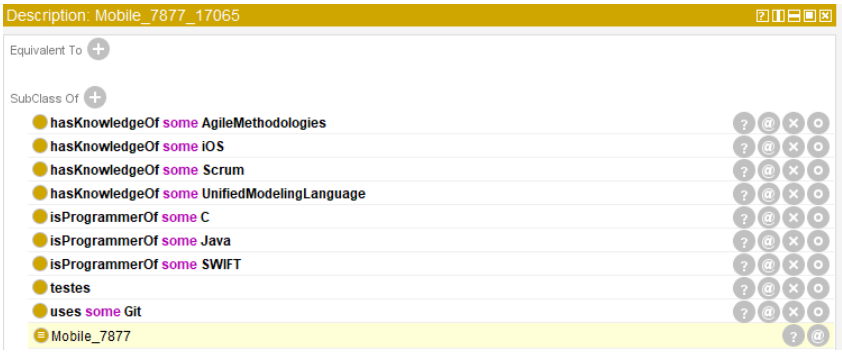

Figure 5.6: Example of the match between a candidate and a job advertisement

As for Figure 5.7, it portrays the match between a job advertisement with the *MobileAppsDeveloper* category, based on the skills required for it.

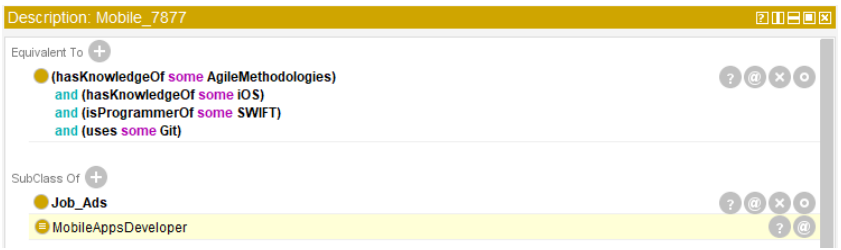

Figure 5.7: Example of the match between a job advertisement and a category

Figure 5.8 shows the case where a candidate with the possessed skills is a match to two categories, *Front-end Developer* and *MobileAppsDeveloper*. It is a possible scenario since several skills are common in both these two categories, which is also the case with others.

There were also cases when no match occurs, either for candidates and for job advertisements, as can be seen in Figures 5.9, 5.10 and 5.11.

Figure 5.8: Example of a candidate matching with two categories

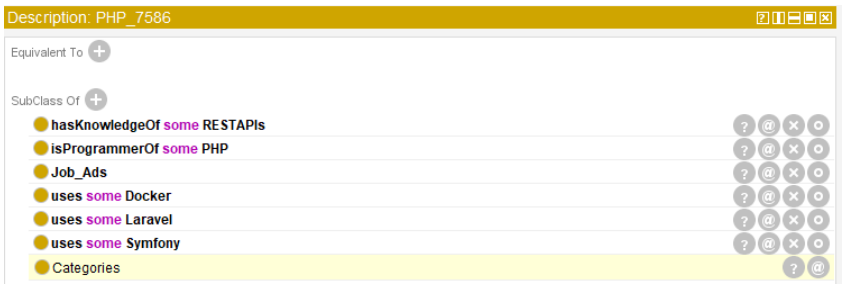

Figure 5.9: Example of a job advertisement matching no category (1)

5.4 Evaluation

This section corresponds to the Evaluation step of the DSRM and will describe how we conducted it on the development of the initial artifact, our ontology. The goal was to find inconsistencies and aspects to improve in the development process, thus avoiding a wrong development process that could lead to a high effort re-work process. Although not being an extensive evaluation phase, it helped to improve the quality of the ontology by adding new contents and modifying the existed ones to better answer the identified problem.

Through this first iteration of the ontology development, we conducted little evaluations to ensure that we were going on the right path, leading to a useful, both above all, a correct ontology for IT Skills. For that matter, we performed three kinds of evaluation in this initial phase:

- With reasoners available in the Protégé, to ensure that nothing was being wrongly made. We used mostly the Pellet reasoner. The usage of reasoners helped us identify problems regarding the definition of classes and properties in the Protégé. Running the reasoner classify and infer the ontology, given what was defined, and highlights problems structural problems of the development;

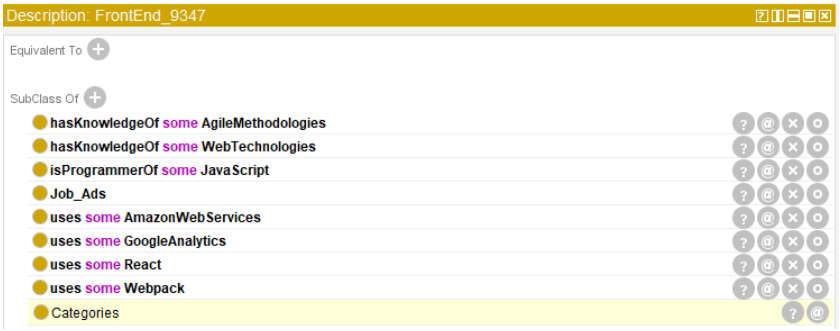

Figure 5.10: Example of a job advertisement matching no category (2)

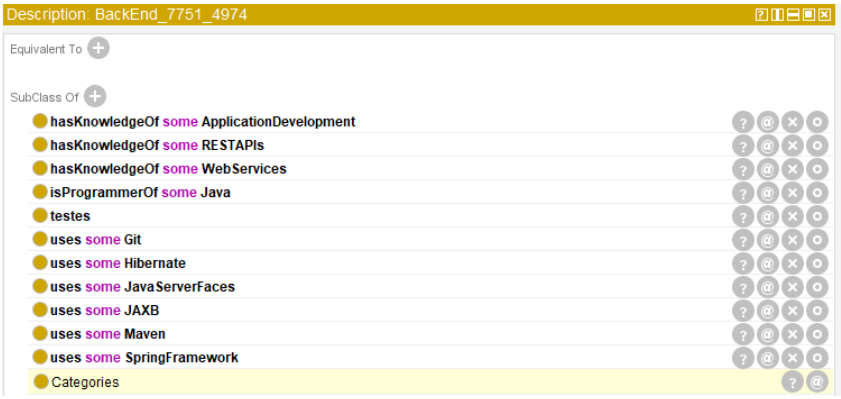

Figure 5.11: Example of a candidate matching no category

- While using real data of candidates and job advertisements from the company DB in the Demonstration step, we put the ontology to the test. It helped to understand its capability to correctly identify to which category a candidate or job advertisement match, and also, if they match between them. Leading to small modifications, performed while conducting the demonstrations, to the defined relations among skills and categories, which improved the number of matches between candidates or job advertisements and job categories;

- Comparing it to another IT Skills ontology [50], developed for the same goal of our own: the recruitment process. When analyzing the ontology, it was possible to identify a couple of properties that we didn't have expressed in our ontology. The property *isFrameworkOf* was one of them and is used to represent relations between skills that are *Programming Languages* and skills that are *Frameworks* of those languages.

  We understood that if added to our ontology, this property would allow us to improve both the overall quality and the knowledge that can be extracted from it. The most notorious improvement is, of course, the fact that it creates a higher understanding of the existing relations between skills, that until this point, were not portrayed in the ontology. From a practical perspective, it allows whomever uses the ontology to know with which Programming Language skill does another one

Framework skill relates.

These improvements were performed to the ontology, and their implementation is described in detail in Chapter 7.

6

Second Iteration

Contents

6.1 Description 47
6.2 Design and Development 47
6.3 Demonstration 50
6.4 Evaluation 52

6.1 Description

This Chapter portrays the second iteration performed to develop our IT Skills Ontology, encompassing the Design and Development, Demonstration and Evaluation steps of the DSRM.

Following the evaluation performed on the ontology resulting from the development process described in Chapter 5, the need for a second iteration to improve the quality of the ontology was clear. Plus, both the research methodology chosen and the ontology development methodology contemplate the possibility to perform iterations over the life-cycle of the development of the artifact/ontology.

6.2 Design and Development

After the evaluations carried out at the end phase of the first iteration, there was a consensus that the representation of a new property to describe the relations among skills was an improvement to be made.

For that, we went back a few steps on the Ontology Development 101 methodology to perform all the required ones to add a new property. This way allowed us to ensure that it was well represented in the ontology. Therefore, we returned to the 5th and 6th steps of the methodology:

5)Define class properties

As mentioned before, this new property has the function of representing the existing relations between skills, and more specifically *Programming Languages* and their *Frameworks*, so this time the relationships between skills and categories were not analyzed.

For example, with this property, the ontology can represent that Django is a framework of the Python programming language. In the case of a candidate represented with the capability to use the framework Django the ontology will infer that he may also have the know-how to program in Python.

6)Define the facets of the properties

As for the domain and range of the property, the type of skills it relates are the ones that defined them. So the skills that are *Programming Languages* represent the domain while the *Frameworks* ones represent the range. This can be easily seen in Table 5.2 and its representation on the ontology in Figure 6.1.

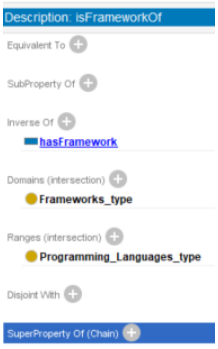

Figure 6.1: Representation of the *isFrameworkOf* property

7)Create instances

In the first iteration, this step of the methodology was not performed. This because the evaluations made were at a point of the development where instances were not yet required. Nevertheless, after adding the new property, we faced the necessity of adding them to the ontology since with only classes the ontology would not correctly represent the property. They represent not only the entire set of skills but also candidates and job advertisements.

Once again, the Cellfie plug-in proved useful since it allowed for an automatic creation of instances and the portrayal of the *isFrameworkOf* property. We created a new excel sheet with the entire set of terms used to represent the ontology skills on one column and then duplicated that column adding a suffix to the terms. This because Protégé would not allow to create instances with the same name as the class (we could not define the Java instance since it already existed as a class).

An analysis of the skills, previously identified as frameworks, was also performed to know with which programming languages they were associated. These relations also were added to the excel sheet. Table 6.1 shows a sample of the relations among the class and the instance name, and the identified frameworks and programming languages relations. The class Python, which represents the programming language, has as an instance Python_I and is not a framework of any other programming language (or skill). On the other hand, the class Espresso, which represents the framework, has as an instance Espresso_I and is a framework of Java_I and Kotlin_I (that represent the classes Java and Kotlin).

Table 6.1: Instances sample

Class	Instance	Framework of
Python	Python_I	—
Django	Django_I	Python_I
Espresso	Espresso_I	Java_I, Kotlin_I

The transformation rule defined in Cellfie to map the instances and relations between frameworks and skills was portrayed in Figure 6.2. The ”Facts” definition on the transformation rule defines that if columns C to L have values, they will be an object property assertion of the instance that is being mapped. These values correspond to other instances as shown in Figure 6.3.

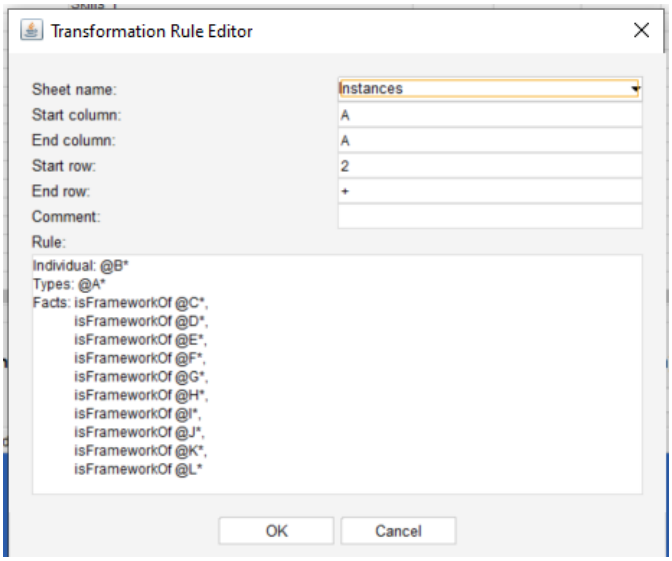

Figure 6.2: Cellfie rules to map instances to the ontology

207	EmberJS	**EmberJS_I**	JavaScript_I	
208	Enterprise Software	**Enterprise Software_I**		
209	Entity Framework	**Entity Framework_I**	.NET_I	
210	Enzyme	**Enzyme_I**	JavaScript_I	
211	Erlang	**Erlang_I**		
212	Espresso	**Espresso_I**	Java_I	Kotlin_I
213	Ethereum	**Ethereum_I**		
214	ETL	**ETL_I**		
215	Event Marketing	**Event Marketing_I**		
216	Express.js	**Express.js_I**	Node.js_I	
217	Ext JS	**Ext JS_I**	JavaScript_I	

Figure 6.3: Sample of the excel sheet used to map the instances

Figure 6.4 is an example of the obtained result after mapping the instances and the *isFrameworkOf* property. It shows the created *Django* instance, named *Django_I*, having as an object property assertion the ”fact” that *isFrameworkOf Python_I* (which is the *Python* class instance).

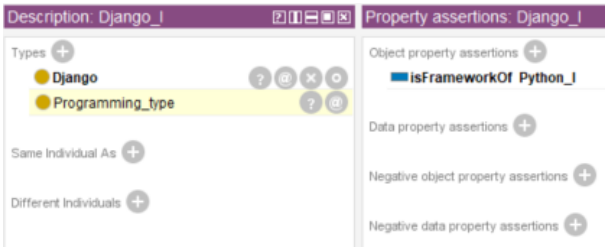

Figure 6.4: Example of the usage of the *isFrameworkOf* property

In the process of adding instances to the ontology, we also defined rules to implement our new property. To do so, we used the Rules view of Protégé to defined the needed ones. They work ”together” with the reasoners. This means that, when running the reasoner ”against” the ontology, both the reasoner and the rules will classify and infer, given what was defined.

The rules portrayed in Figure 6.5 represent the two rules defined to work on the *isFrameworkOf* property. So that when running the reasoner, the compilation also classify and infer that if some candidate or job advertisement uses a given framework, and that framework is related to a given programming language by the *isFrameworkOf* property, then the candidate or job advertisement will also program in that programming language.

Figure 6.5: Rules for the *isFrameworkOf* property

6.3 Demonstration

This section covers the Demonstration step of the DSRM, where for the exact purpose of demonstrating the feasibility of the developed solution, data from candidates and job advertisements sourced from the company’s DB will be used. This time, we focused more on the created individuals in opposition to the first iteration, where the focus was on the classes.

The instance ”Candidate_9164” represents one candidate of the company’s DB, with the skills he possesses, defined as object property assertions. With his set skill, after running the reasoner, the profile match’s the ”Back-end developer” and the ”ElectronicsAndTelecommunications” categories, as Figure 6.6 shows.

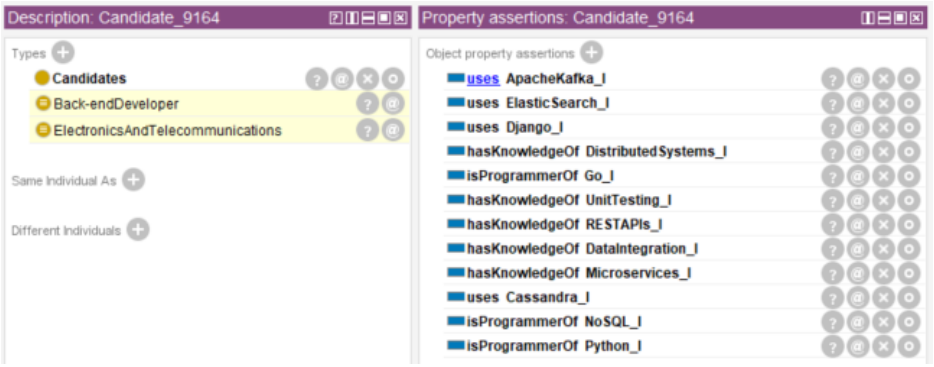

Figure 6.6: Instance of the candidate 9164

Then we removed the object property assertion of *"isProgrammerOf Python_I"* thus, eliminating the direct relation between the candidate and the *Python* skill. Due to the inference of the rule previously defined, when running the reasoner, the ontology classifies the candidate as being able to program in *Python* since he uses the *Django* framework. As shown in Figure 6.7.

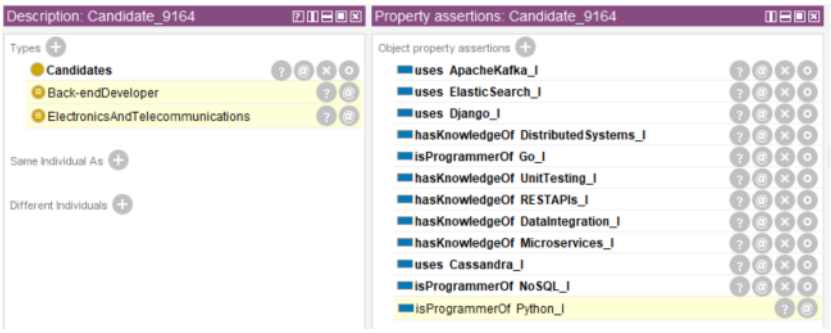

Figure 6.7: Instance of the candidate 9164 without the Python skill

Figure 6.8 shows another example of a match, this time the "Candidate_7565", with his skill set, matches with the"MobileAppsDeveloper" category.

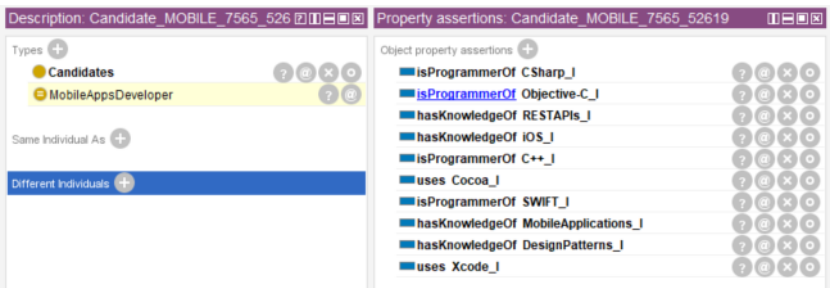

Figure 6.8: Instance of the candidate 7565

The last example describes the case when the candidate matches with a given job advertisement and category, but on the other hand, there is no match between job advertisement and category. It is the case of the ”Candidate_9127”, in which the skills required for the job are not enough to match a category, but the candidate does match both the job advertisement and the ”Full-stack Developer” category. Figure 6.9 portrays this case.

Figure 6.9: Job example that does not match a category but a candidate match’s the job and a category

6.4 Evaluation

In this Evaluation section, which corresponds to the fifth step of the DSRM, we aim to verify how well the developed artifact supports a solution to the problem described in Chapter 3.

Similar to the evaluation process conducted in the first iteration, we conducted this evaluation with different kinds of approaches. But this time, we conducted a much more extended and deepened evaluation to the ontology:

- Through the Competency Questions defined in the first step of the development methodology, allowing us to verify to what extent the developed ontology can answer them also identifying some limitations;
- With Reasoners available in the Protégé, to ensure that nothing was being wrongly made. We used mostly the Pellet reasoner. It also helped with the defined rule;
- Using real data of candidates and job advertisements from the company DB, we put the ontology to the test. To understand its capability to correctly identify to which category a candidate or job advertisement match, if they match between them, and if the added property increased the generated knowledge;
- An overall assessment, following the identified criteria and levels in Table 2.2.

6.4.1 Competency Questions

The defined competency questions, also served as a means of evaluation for our ontology, and their result interpretation can be seen in Table 6.2.

Table 6.2: Competency Questions result interpretation

	Competence Question	Result Interpretation
Q1	What are the best suited category for a candidate with the skills PHP, HTML and Javascript?	Front-end Developer and Full-stack Developer.
Q2	Which skills one should have to be a good ”Front-end developer”?	One should program in at least two programmig languages and use one programmig framework. Due to a very high level of description there are numerous skills associated with Front-end Development.
Q3	Java is a good to have skill for which categories?	Back-end Developer, DevOps and SysAdmin, Fullstack Developer and MobileApps Developer.
Q4	Is Bootstrap a good complementary skill for HTML?	Yes, every category that has HTML as a skill also has CSS. Bootstrap is a framework of CSS, and therefore it is a good skill combination.
Q5	What are the skills shared by a Front--end developer and a Data Scientist?	There are no skills shared by this two categories in the ontology.
Q6	What are the programming frameworks associated with Python?	Django, Falcon, Flask, Keras, NumPy, Pandas, Pyramid, Qlik Sense, SQLAlchemy, Starlette, TensorFlow and ZeroMQ
Q7	Can the ontology provide match's between candidates and jobs ads?	Yes, but only if the candidate has all the job ad skills.
Q8	Can the ontology provide match's between jobs ads and categories?	Yes, but many job ads have few skills, which having such high-level categories makes hard to have match's.
Q9	Can the ontology provide match's between candidates and categories?	Yes, although Q7 problem is also verified adding to the fact that to many candidates have more than 10 skills, which facilitates the match.
Q10	Can the defined properties provide inferred knowledge ?	Yes. The isFrameworkOf provides inferred knowledge, since just after inferred, will it compile and be represented in the ontology due to the defined rules.

6.4.2 Reasoners

We used reasoners in the entire development process, as said before, to ensure that in terms of construction, relations between classes, usage of properties, nothing was being wrongly made. Plus, in this second iteration, with the new property and rules, running the reasoner also inferred and classified

rules. We were able to verify, if after compiling, the defined rules were working or not.

Figure 6.10 shows as an example of the practical use of rules and relations that we defined in the second iteration. In particular, two relations: between a) Django and Python and b) Java Server Faces and Java. It is possible to observe that being Django defined as a framework of Python (the same with Java Server Faces and Java), after compiling, it is inferred that the candidate ”Framework_Test” is capable of programming in Python (and Java).

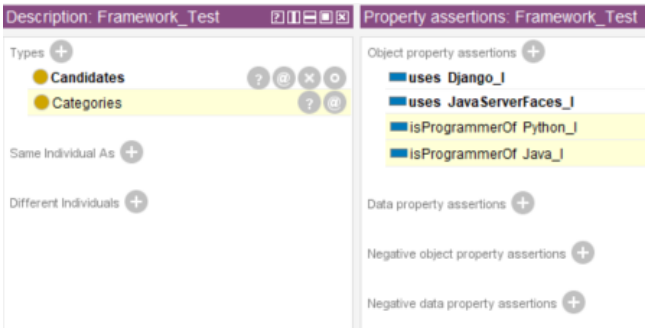

Figure 6.10: Rules inferred after compilation

6.4.3 Data-Driven with Candidates and Job Advertisements

The Demonstration step performed in both iteration provided a Data-driven evaluation of our solution. Where we took advantage of the information regarding candidates and job advertisements present in the company's DB. With it, we were able to better adjust the needed minimum skills that one must possess to match to a given category and also prove that the ontology can match candidates with job advertisements.

Performing this evaluation, we notice that some candidates and job advertisements had few skills and would not match any category. To a certain point, it was reasonable to reduce the category number of minimum skills required to match but reducing more would lead to one candidate or job advertisement match several categories based on a handful of skills.

While in the first iteration, these small modifications were applied to the defined relations, increasing the generated knowledge, in this second iteration, we did not change them since they were at a point that no increase of knowledge was to be gained with modifications.

In the first iteration, the evaluation of candidates (and job advertisements) focused more on the classes, while in the second one, it focused more on the individuals rather than as classes. This because it allowed to test the rules, and therefore, the added new property, to understand if the improvement helped in the ontological classification process.

Associated with job advertisements were names and descriptions that usually contained a specific position that could identify as a subclass of a category already present in the ontology. If represented in the ontology, that position would allow to match jobs and even candidates, with a higher level of correctness (i.e. a further specification of the *MobileAppsDeveloper* could be made, to define specifically *iOS* or *Android* developers).

6.4.4 Evaluation Assessment

This section presents the result of the evaluation assessment on the ontology, conducted through the first and second iteration, and follows the criteria and the chosen approach levels identified in Table 2.2.

Regarding the criteria parameters, the results were the following:

- *Accuracy* - Although no user evaluation was conducted, the ontology captures and represents the view of a company in the recruitment for the IT field, and to that extent, it can be considered as accurate;
- *Completeness* - If we consider the whole IT area as a domain, then the ontology does not cover it entirely. However, considering as domain the universe in which the company was developing its work, at the time in which the data was acquired, and the capacity to answer the formulated competency questions, then the ontology covers the domain to a certain point;
- *Computational efficiency* - As we added more candidates and job advertisements to the ontology, it would take more time for reasoners to process it. We can expect that the ontology, at least if used directly in Protégé, although being able to grow, it will take more time for reasoners to process it. A solution may pass by removing all the unnecessary classes or instances (remove non-used candidates or job-advertisements) or use the ontology definitions as a knowledge base for a tool with a higher processing capacity than Protégé;
- *Conciseness* - In fact, it does include some irrelevant axioms. There were three categories in the data that we opted to include in the ontology;
- *Consistency* - The usage of reasoners helped to ensure in fact that the developed ontology is consistent;
- *Expandability* - The ontology is quite tolerant to the addition of new definitions and concepts. A practical example is the addition of the most recent property and the relationships that were defined from it, in which there was no need to make changes to the existing structure;
- *Organizational fitness* - We were not able to evaluate this parameter since our ontology was not deployed in an organizational environment;

- *Sensitiveness* - The process of identifying the minimum skills needed to match a candidate or job advertisement with a category (or more) has shown that there is considerable sensitivity on the part of ontology. Sometimes reducing the minimum number of a given relation by one was enough to infer three or four new matches (in a universe of 50 candidates and job advertisements).

As previously mentioned, we conducted a data-driven evaluation of the ontology, which acts on three of the six levels identified in Table 2.2. Our analyses of them is as follows:

- *Lexical, vocabulary, or data layer* - All the vocabulary used for the construction of the ontology, from the terms of skills, categories to properties and the hierarchy class, are terms whose source is the company's DB or other such as existing IT ontologies;

- *Hierarchy or taxonomy* - This ontology's main objective is to serve as a tool to facilitate the recruitment process, and in particular, to help in the match process between candidates and categories or job advertisements. Therefore, this was our main goal during the development, so the created concept hierarchy, and here we included not only the relationship between skills and categories but also the class hierarchy, was undoubtedly our focus;

- *Other semantic relations* - The calculation of the accuracy performed in the first iteration set, with 20 candidates and 32 job advertisements, resulted in a 57% precision, meaning that in 57% of the cases, there was a match between candidates or job advertisements and at least one category. This value increased slightly to 60% after the improvements of the second iteration. This values does not include matches between candidates and job advertisements since this match only depends on the skill set possessed and required.

  Based on the set of candidates and job advertisements mentioned above, it was also possible to verify that adding the new property and rule returned eight inferences. Meaning that in eight cases, it generated knowledge of programming in a particular language from a possessed or required framework;

- As for the remaining three, context or application, syntactic level and structure, architecture and design, since their evaluation results of following other approaches besides the Data-Driven, we do not have any outcome for them.

7

Conclusion

Contents

7.1 Contributions 59
7.2 Limitations 60
7.3 Future Work 60

This thesis work followed the DSRM, which comprises six steps, where after a literature review in the context of this work, we defined the problem in the first step as **the lack of ontologies for IT skills, developed taking advantage of empirical information existing in the recruitment companies themselves to better portray the domain in which they work**. From the identified problem, a proposal for the development of an IT Skills Ontology was outlined. It followed the Ontology Development 101 methodology, which consists of seven steps for an ontology development, having been the terms of the ontology gathered from an IT recruitment company's DB.

Both methodologies that guided this work contemplate iterations. Therefore, the development was performed in two iterations of the DSRM, targeting the Demonstration and Evaluation steps, being the Ontology Development 101 also a target of these iterations.

Regarding the first iteration, in the Demonstration step of the DSRM, the first six steps of the development methodology were performed, followed by the Evaluation step of the DSRM. Based on the results of the evaluation, a second iteration was performed to improve the ontology quality and inferred knowledge. Regarding the second iteration, the Demonstration step of the DSRM started at the fifth step of the development methodology, being all the remaining steps performed, after which a final Evaluation phase was conducted to evaluate the resulting ontology.

Five approaches were used to evaluate the ontology. In the first iteration, a comparative evaluation, conducted with another IT Skills ontology, which resulted in improvements to the ontology under development. Common to both iterations was the use of Reasoners to validate the correct construction of the ontology, as well as a Data-Driven approach, where candidates and job advertisements data sourced from the DB demonstrated the usefulness of the ontology in the match between candidates and job advertisements or categories. Only in the second iteration did we evaluate the ontology's ability to answer the defined Competency Questions and an overall assessment, which returned encouraging results of the ontology's validity as a solution for the problem identified.

7.1 Contributions

With this thesis work, we hope to contribute to answer the identified problem with the solution that was developed, but also that it can be used in future research. The development of this work allowed to make the following contributions: (1) the creation of an understanding of the fundamental of ontologies, how to develop them with a high focus on bottom-up ontologies, (2) a more detailed account of how to perform the development of an ontology following the Ontology Development 101 methodology and (3) an ontology of IT Skills, developed following a bottom-up approach, that contributes to an easier matching between candidates and IT categories.

The resulting evaluation of applying the ontology to a set of data representing its field of action allows

us to verify its usefulness as an IT Skills Ontology in the recruitment context.

7.2 Limitations

During the development of this thesis, we faced some limitations and challenges:

- In terms of methodologies for ontology development, most of the ones we studied did not contain practical examples of how they were applied;
- Somewhat related to the previous point, the ontologies we found, from IT and other domains, made little reference to the applied development methodology;
- Lack of information on IT Skills Ontologies, especially regarding the lack of detail on how their development;
- The process of identifying each skill to know its functionality and consequent identification of the category to which it belongs;
- The limitation of the Cellfie plug-in at the mapping level, which led to more manual work in defining the relationships identified for the ontology;
- The defined categories were of such a high level that it was complex to define the necessary minimums for the match;
- The biggest one was the change of context for which the ontology was developed. The initial idea was to be an ontology to serve as a comparison with a second ontology, developed following a non-bottom-up approach, in which the goal was to arrive at a final ontology that was composed of the best aspects of the two. The ontology therefore lacks on-field usage, thus only performing a data-driven and structural evaluation.

7.3 Future Work

The results of the evaluation of the ontology allowed us to verify that there are certain aspects of the ontology which can improve as a way to increase the quality and inferred knowledge.

By the evaluation conducted, it is clear that the ontology lacks detail as to the categories that it represents. In terms of high-level representation, the ones defined are adequate for the IT field. However, being so high level, the specification of each one is reduced. It is one aspect that could be further detailed, even with the usage of the company's data, by representing a specific job position. For example, there are several job advertisements for Java and PHP developers, among others alike, that could add to

the ontology as subclasses of the Back-end Developer category, and thus creating a greater specificity of the category.

Another feature regarding the ontology properties that would increase the knowledge generated by the resulting ontology is a property that relates similar skills, for example, skills as PostgreSQL and MySQL.

New instance properties, such as the version of a given skill, the experience a candidate has with a given skill, or the experience needed with a given skill for a given job.

To reduce the required time for the classification and inference process, thus boosting the amount of knowledge generated at once, it may be considered the development of an application whose core is the ontology and around it a system capable of taking a higher advantage of it. A system that also facilitates the introduction of new elements such as candidates and job advertisements, with a more user-friendly design.

Bibliography

[1] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.

[2] K. K. Breitman, M. A. Casanova, and W. Truszkowski, "Methods for ontology development," *Semantic Web: Concepts, Technologies and Applications*, pp. 155–173, 2007.

[3] N. F. Noy, D. L. McGuinness *et al.*, "Ontology development 101: A guide to creating your first ontology," 2001.

[4] C.-V. Priporas, N. Stylos, and A. K. Fotiadis, "Generation z consumers' expectations of interactions in smart retailing: A future agenda," *Computers in Human Behavior*, vol. 77, pp. 374–381, 2017.

[5] C. Piotrowski and T. Armstrong, "Current recruitment and selection practices: A national survey of fortune 1000 firms," *North American Journal of Psychology*, vol. 8, no. 3, pp. 489–496, 2006.

[6] P. Van Esch, J. S. Black, and J. Ferolie, "Marketing ai recruitment: The next phase in job application and selection," *Computers in Human Behavior*, vol. 90, pp. 215–222, 2019.

[7] M. H. Randall and C. J. Zirkle, "Information technology student-based certification in formal education settings: Who benefits and what is needed," *Journal of Information Technology Education: Research*, vol. 4, no. 1, pp. 287–306, 2005.

[8] J. Pinkelman, "Computing changes: an industry perspective," *ACM Inroads*, vol. 4, no. 4, pp. 39–42, 2013.

[9] U. Shahbaz, A. Beheshti, S. Nobari, Q. Qu, H.-Y. Paik, and M. Mahdavi, "irecruit: Towards automating the recruitment process," in *Service Research and Innovation*. Springer, 2018, pp. 139–152.

[10] C. Calero, F. Ruiz, and M. Piattini, *Ontologies for software engineering and software technology*. Springer Science & Business Media, 2006.

[11] A. Gómez-Pérez, J. Ramírez, and B. Villazón-Terrazas, "Reusing human resources management standards for employment services." in *FIRST*, 2007, pp. 28–41.

[12] M. Mochol, H. Wache, and L. Nixon, "Improving the accuracy of job search with semantic techniques," in *International Conference on Business Information Systems*. Springer, 2007, pp. 301–313.

[13] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, pp. 75–105, 2004.

[14] P. Attewell, "What is skill?" *Work and occupations*, vol. 17, no. 4, pp. 422–448, 1990.

[15] F. E. Weinert, "Concept of competence: A conceptual clarification." 2001.

[16] F. D. Le Deist and J. Winterton, "What is competence?" *Human resource development international*, vol. 8, no. 1, pp. 27–46, 2005.

[17] F. Green, *Skills and skilled work: An economic and social analysis*. Oxford University Press, 2013.

[18] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review*, vol. 11, no. 2, pp. 93–136, 1996.

[19] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International journal of human-computer studies*, vol. 43, no. 5-6, pp. 907–928, 1995.

[20] O. Corcho, M. Fernández-López, and A. Gómez-Pérez, "Methodologies, tools and languages for building ontologies. where is their meeting point?" *Data & knowledge engineering*, vol. 46, no. 1, pp. 41–64, 2003.

[21] S. Staab and R. Studer, *Handbook on ontologies*. Springer Science & Business Media, 2010.

[22] P. E. Van Der Vet and N. J. Mars, "Bottom-up construction of ontologies," *IEEE Transactions on Knowledge and data Engineering*, vol. 10, no. 4, pp. 513–526, 1998.

[23] R. Ramakrishnan and J. Gehrke, *Database management systems*. McGraw Hill, 2000.

[24] T. Gruber, *Ontology*. Boston, MA: Springer US, 2009, pp. 1963–1965.

[25] B. Yildiz, *Ontology-driven information extraction*. na, 2007.

[26] M. Fernández-López, "Overview of methodologies for building ontologies," in *IJCAI99 workshop on ontologies and problem-solving methods: Lessons learned and future trends*, vol. 430, 1999.

[27] M. Fernández-López and A. Gómez-Pérez, "Overview and analysis of methodologies for building ontologies," *The knowledge engineering review*, vol. 17, no. 2, pp. 129–156, 2002.

[28] H. S. Pinto and J. P. Martins, “Ontologies: How can they be built?” *Knowledge and information systems*, vol. 6, no. 4, pp. 441–464, 2004.

[29] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, “Methontology: from ontological art towards ontological engineering,” 1997.

[30] S. Staab, R. Studer, H.-P. Schnurr, and Y. Sure, “Knowledge processes and ontologies,” *IEEE Intelligent systems*, vol. 16, no. 1, pp. 26–34, 2001.

[31] M. Grüninger and M. S. Fox, “Methodology for the design and evaluation of ontologies,” 1995.

[32] A. Bernaras, I. Laresgoiti, and J. M. Corera, “Building and reusing ontologies for electrical network applications,” in *ECAI*, 1996.

[33] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, “The neon methodology for ontology engineering,” in *Ontology engineering in a networked world*. Springer, 2012, pp. 9–34.

[34] O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez, “Ontological engineering: what are ontologies and how can we build them?” in *Semantic web services: Theory, tools and applications*. IGI Global, 2007, pp. 44–70.

[35] A. Gómez-Pérez, “Ontology evaluation,” in *Handbook on ontologies*. Springer, 2004, pp. 251–273.

[36] D. Vrandečić, “Ontology evaluation,” in *Handbook on ontologies*. Springer, 2009, pp. 293–313.

[37] J. Brank, M. Grobelnik, and D. Mladenic, “A survey of ontology evaluation techniques,” in *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*. Citeseer Ljubljana, Slovenia, 2005, pp. 166–170.

[38] M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, “Introduction: Ontology engineering in a networked world,” in *Ontology engineering in a networked world*. Springer, 2012, pp. 1–6.

[39] C. M. Keet, “Dependencies between ontology design parameters,” *International Journal of Metadata, Semantics and Ontologies*, vol. 5, no. 4, pp. 265–284, 2010.

[40] B. Xu, H. Cai, and L. Jiang, “A clinic ontology construction method in distributed hospital information systems,” in *2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2013, pp. 685–689.

[41] J. E. Murphy and R. Steele, “A framework for semantically rich legal documents and applications.” in *WEBIST (2)*, 2008, pp. 193–200.

[42] S. Lee, W. Seo, D. Kang, K. Kim, and J. Y. Lee, “A framework for supporting bottom-up ontology evolution for discovery and description of grid services,” *Expert systems with Applications*, vol. 32, no. 2, pp. 376–385, 2007.

[43] Y. Shu, D. Ratcliffe, M. Compton, G. Squire, and K. Taylor, “A semantic approach to data translation: A case study of environmental observations data,” *Knowledge-Based Systems*, vol. 75, pp. 104–123, 2015.

[44] P. Ceravolo, A. Corallo, E. Damiani, G. Elia, M. Viviani, and A. Zilli, “Bottom-up extraction and maintenance of ontology-based metadata,” *Fuzzy Logic and the Semantic Web*, pp. 265–282, 2006.

[45] J. Borrego-Díaz, A. M. Chávez-González, J. L. Pro-Martín, and V. Matos-Arana, “Specifying and verifying meta-security by means of semantic web methods,” in *International Joint Conference SOCO’14-CISIS’14-ICEUTE’14*. Springer, 2014, pp. 355–365.

[46] H. Ren, J. Tian, A. P. Wierzbicki, Y. Nakamori, and E. Klimasara, “Ontology construction and its applications in local research communities,” in *Modeling for Decision Support in Network-Based Services*. Springer, 2012, pp. 279–317.

[47] E. Francesconi, S. Montemagni, W. Peters, and D. Tiscornia, “Integrating a bottom–up and top–down methodology for building semantic resources for the multilingual legal domain,” in *Semantic Processing of Legal Texts*. Springer, 2010, pp. 95–121.

[48] H. Lv and B. Zhu, “Skill ontology-based semantic model and its matching algorithm,” in *2006 7th International Conference on Computer-Aided Industrial Design and Conceptual Design*. IEEE, 2006, pp. 1–4.

[49] C. D. Nguyen, K. D. Vo, D. B. Bui, and D. T. Nguyen, “An ontology-based it student model in an educational social network,” in *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, 2011, pp. 379–382.

[50] B. Carrilho Toscano, M. Mira da Silva, and R. Sousa Pereira, “It skills ontology,” https://fenix.tecnico.ulisboa.pt/cursos/meic-a/dissertacao/846778572213024, (Accessed on 02/07/2021).

[51] M. L. Gusdorf, “Recruitment and selection: Hiring the right person,” *USA: Society for Human Resource Management*, 2008.

[52] M. Fazel-Zarandi and M. S. Fox, “Semantic matchmaking for job recruitment: an ontology-based hybrid approach,” in *Proceedings of the 8th International Semantic Web Conference*, vol. 525, 2009.

[53] S. Saad, N. Salim, H. Zainal, and Z. Muda, “A process for building domain ontology: An experience in developing solat ontology,” in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE, 2011, pp. 1–5.

Appendix A

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	**Skills**	**Obs**	**Description**	**Hierarchy**	**Property**	**Video /**	**Mobile**	**QA /**	**Business**	**Electronic**	**Product**	**Maintenance**	**Compute**	**DevOps**	**Data**	**Front-end**	**Full-stack**	**Back-end**	**UX / UI**	**Sales**	**Marketing**
2	2D Animation			Image_Video_ProcessingK type	KnowledgeOf	x															
3	3D animation and			Image_Video_ProcessingK type	KnowledgeOf	x															
4	A/B testing			Software_Development type	KnowledgeOf			x													
5	ABAP		linguagem de	Functional type	ProgrammerOf											x	x	x			
6	Abstract		version control	DevelopmentT type	UsedBy																
7	Account Management			Comms_Marking_Sales type	KnowledgeOf															x	
8	Active Directory			Software_Architecture type	KnowledgeOf					x		x		x							
9	ActiveMQ		apache for	ProgrammingFW type	KnowledgeOf					x								x			
10	Adobe ColdFusion		plataform for web/mobile	DevelopmentT type	UsedBy		x														
11	Adobe Creative Suite		aplicativos para design	Image_Video_ProcessingT type	UsedBy	x															
12	Adobe Flex		framework mobile, web,	Image_Video_ProcessingT type	UsedBy	x										x	x	x			
13	Adobe Lightroom		edição de imagens,	Image_Video_ProcessingT type	UsedBy	x															
14	Adobe Premiere		edição de video	Image_Video_ProcessingT type	UsedBy	x															
15	Adobe XD		ferramenta de design	DevelopmentT type	UsedBy		x									x	x	x			
16	Advertising			Comms_Marking_Sales type	KnowledgeOf															x	
17	AEM		Adobe Experience	Business_ManagementT type	UsedBy		x									x	x	x			
18	After Effects		pós-producão de video,	Image_Video_ProcessingT type	UsedBy	x															
19	Agent Based Modelling		actions and interection of	Software_Development type	KnowledgeOf	x															
20	Agile-Waterfall Hybrid		comportamentos,	Software_Development type	KnowledgeOf						x										
21	Agile Methodologies		comportamentos,	Software_Development type	KnowledgeOf			x			x										
22	Agile Transformation			Software_Development type	KnowledgeOf						x										
23	AI		inteligência artificial	Software_Development type	KnowledgeOf										x						
24	AJAX		javascript, json, XML	ProgrammingFW type	UsedBy											x	x	x			
25	Akka		ferramentas apps	ProgrammingFW type	UsedBy													x			
26	Alfresco		Sist. Gestão de conteudo	Business_ManagementT type	UsedBy				x												
27	Altera		dispositivos logicos	Hardware type	UsedBy					x											
28	Alteryx		ciencia e analise de dados	DataScience type	UsedBy										x						
29	Amazon Web Services		plat. Computação em	Cloud type	UsedBy									x							
30	Analytical Skills			DataScience type	KnowledgeOf																
31	Android		Mobile OS	Software_Architecture type	ProgrammerOf		x														
32	Android Jetpack		libraries, tools, for mobile	ProgrammingFW type	ProgrammerOf		x														
33	Angular		typescript - superconjunto	ProgrammingFW type	ProgrammerOf											x	x	x			
34	Animation		manipulação de imagens	Image_Video_ProcessingK type	KnowledgeOf	x															
35	Ansible		tool provisionamento,	Business_ManagementT type	UsedBy				x												
36	Apache		free web server	DevelopmentT type	UsedBy													x			
37	Apache Airflow		plataforma para gerir fluxo	Business_ManagementT type	UsedBy				x												
38	Apache Camel		open source orientado a	DevelopmentT type	UsedBy					x											
39	APC		Alternative PHP Cache	ProgrammingFW type	UsedBy													x			
40	Apex		Application Express – a	DevelopmentT type	UsedBy											x		x	x		

Figure A.1: Excel sample