Double Degree Final Project

DEVELOPMENT OF AN AUTOMATED SYSTEM FOR MANUFACTURING COST CALCULATION IN SHEET METAL PARTS FOR THE AERONAUTICAL INDUSTRY

LEONARDO CAVALCANTI HERNANDES

Mechatronics Engineering

Tutor/a: Jordi Surinyac Albareda

Co-tutor: Anderson Szejka

June of 2023

Acknowledgments

Initially I thank God for being present at all times of my life and for giving me health and strength every day. I also thank my parents, Alceu e Edlainy, who always believed in the education I was given and trusted in me. I thank my godmother Cláudia who, along with my late godfather Paulo Lemes, encourages me in absolutely everything I set out to do, making my life so much happier. I also thank my sister Luysa, for being so companionable, kind, and special, and who, many times, showed me the happiness of life without even realizing it.

My special thanks to my “nona” Lúcia Mazzorana, who even though she lives far away from me is always with me in all moments of my day-to-day life. I also thank my aunt Ednéia and my cousin Miguel, who strengthen me and are part of my family.

I would like to thank my supervisor Anderson Luis Szejka, who spared no effort to help me in all the development tasks, always guiding me to the best solution. I am extremely grateful to my supervisor from the UVic, Jordi Surinyac Albareda, who was always very helpful with all the difficulties overcome. I am also grateful to Murillo Skrzek for his contribution in defining concepts that I had no knowledge of and in proposing solutions to simplify and solve the problem.

Finally, I thank all my friends who supported me, gave me strength in times I needed it, and who make my life better every day; after all, life is made of friends.

Resum

Títol: Desenvolupament d'un sistema automatitzat de càlcul de costos de fabricació de peces de xapa per a la indústria aeronàutica.

Autor: Leonardo Cavalcanti Hernandes

Co-Tutores: Dr. Anderson Szejka (PUCPR) i Dr. Jordi Surinyac Albareda (UVic)

Data: Juny de 2023

Paraules clau: *Indústria aeronàutica. Sistema automatitzat. Preu de les peces.*

La indústria aeronàutica té una gran complexitat en tots els seus projectes, amb variables que s'han d'observar en de cada procés i en les quals qualsevol error comporta un cost addicional en el desenvolupament d'un projecte. La funció principal d'aquest treball és mostrar el procés de disseny i desenvolupament d'un sistema automatitzat capaç de calcular els costos de fabricació de peces de xapa en la indústria aeronàutica a partir de múltiples paràmetres heterogenis. Per a aquest desenvolupament s'estudiaran conceptes sobre els processos de conformació pels quals han de passar les peces, així com la comprensió del procediment de fixació de preus de peces complexes i la definició de regles de negoci per, així, dur a terme les regles d'estructuració i el desenvolupament d'una ontologia que comprengui totes les variables necessàries. La implementació de la solució es basarà en la construcció d'una ontologia en format OWL juntament amb el desenvolupament d'un programa en el llenguatge de programació Python, que usarà funcions i biblioteques per a la inserció i integració d'ontologies. En finalitzar la implantació i desenvolupament del sistema automatitzat, el resultat serà l'aplicació d'aquesta solució en una peça real per a obtenir-ne el valor de cost estimat, tenint en compte les seves particularitats i altres variables i paràmetres de procés.

Summary

Title: Development of an Automated System for Manufacturing Cost Calculation in Sheet Metal Parts for the Aeronautical Industry

Author: Leonardo Cavalcanti Hernandes

Supervisor: Dr. Anderson Szejka (PUCPR) and Dr. Jordi Surinyac Albareda (UVic)

Date: June of 2023

Keywords: *Aeronautics industry. Automated system. Parts pricing.*

The aeronautical industry has a high complexity in all its projects, which have variables to be observed within each process, in which any error that occurs brings an additional cost in the development of a project. The main role of this work is to show the process of conception and development of an automated system capable of performing the calculation of the involved manufacturing costs of sheet metal parts in the aeronautical industry based on multiple heterogeneous parameters. For this development, concepts about the forming processes that the parts must go through will be studied, along with the understanding of the pricing procedure for complex parts and the definition of business rules in order to perform the structuring rules and the development of an ontology that involves all the necessary variables. The implementation of the solution will be based on the implementation of an ontology in OWL format together with the development in Python programming language, using functions and libraries for insertion and integration of ontologies. At the end of the implementation and development of the automated system, the result will be the application of this solution on a real part to obtain the estimated cost value, taking into account its particularities and other variables and parameters of the process.

SUMMARY

1. Introduction i
- 1.1. Context i
- 1.2. Problem iii
- 1.3. Objectives iv
  - 1.3.1. General objectives iv
  - 1.3.2. Specific objectives iv
- 1.4. Justificative v
- 1.5. Research methodology vi

2. Bibliographic review vii
- 2.1. Aeronautical parts forming viii
- 2.2. Ontologies x
- 2.3. Pricing of production costs xii

3. Development xiv
- 3.1. 3D Features extraction xv
- 3.2. Business rules xvii
  - 3.2.1. General definition of all involved variables xvii
  - 3.2.2. Definition of the rules xix
- 3.3. Ontology development xx
  - 3.3.1. Class definition xx
  - 3.3.2. Object and Data properties definition xxiv
  - 3.3.3. Ontology Individuals xxvi
- 3.4. Obtaining the cost for the part xxvii
  - 3.4.1. Detailing the created functions xxviii
  - 3.4.2. Declaration of variables and parameters and functions executions xxxii

4. Test-case xxxiv

5. Results and discussion xxxvi

6. Conclusion xxxviii

7. References xxxix

Table List

Table 1 – Description of the stipulated price by weight of steel types………………………. 18
Table 2 – Variables and parameters needed for the definition of business rules……………. 18

Figure List

Figure 1 Industrial division of the A400M aircraft (F.MAS et al., 2013) ii
Figure 2 Press forming process (LEMINEN et al., 2015) viii
Figure 3 Illustration of the bending processes (MARCINIAK, 2002) ix
Figure 4 Illustration of the pressing and bending processes (MARCINIAK, 2002) ix
Figure 5 Example of ontology in the field of ecology (LAMY JEAN-BAPTISTE, 2021) xi
Figure 6 MASON proposed ontology (S. LEMAIGNAN et. al., 2006) xii
Figure 7 Proposed cost quotation for manufacturing thin sheet metal (GEIGER, 1999) xiii
Figure 8 Conceptual architecture of the implementation (the author, 2023) xiv
Figure 9 The taxonomy of feature types proposed (GHAFFARISHAHRI, 2023) xv
Figure 10 Example of the software output file (the author, 2023) xvi
Figure 11 Step model named “Val15” (the author, 2023) xvi
Figure 12 OntoGraf of the main classes of the ontology (the author, 2023) xxi
Figure 13 Product_Data subclass demonstration (the author, 2023) xxi
Figure 14 Product_Data and Product_Features subclasses detailed (the author 2023) xxii
Figure 15 Machining_Reference OntoGraf representation (the author, 2023) xxii
Figure 16 Material_Reference class demonstration (the author, 2023) xxiii
Figure 17 Product_Cost_Composition OntoGraf subclasses (the author, 2023) xxiii
Figure 18 Product_Cost_Composition subclasses detailed (the author, 2023) xxiv
Figure 19 Data Properties implemented (the author, 2023) xxv
Figure 20 Separation of Data Properties in main classes design (the author, 2023) xxv
Figure 21 Example of application of properties (the author, 2023) xxvi
Figure 22 Elements of the “Individual” type (the author, 2023) xxvi
Figure 23 Output information associated with Individual definitions (the author, 2023) xxvii
Figure 24 Loading the ontology into the program (the author, 2023) xxviii
Figure 25 Functions created in the algorithm (the author, 2023) xxviii
Figure 26 Definition of the Product_Cost function (the author, 2023) xxix
Figure 27 Function Get_ID_Information definition. (the author, 2023) xxix
Figure 28 Get_Machines_Into_Individuals function description (the author, 2023) xxix
Figure 29 Code snippet of the Get_Individuals_Information function (the author, 2023) xxx
Figure 30 Detail of the function Get_Setup_Costs in the code (the author, 2023) xxxi
Figure 31 Get_Manufacturing_Cost function description (the author, 2023) xxxi
Figure 32 Get_Overhead_Costs code snippet (the author, 2023) xxxii
Figure 33 Get_Product_Finishing definition (the author, 2023) xxxii
Figure 34 Declaration of parameters and useful variables (the author, 2023) xxxiii
Figure 35 Code snippet that performs the verification (the author, 2023) xxxiv
Figure 36 Output message with condition not fulfilled (the author, 2023) xxxiv
Figure 37 Program output without the Individual “ID8” (the author, 2023) xxxv
Figure 38 Program output with the new Individual “ID8” (the author, 2023) xxxv
Figure 39 General ontology developed (the author, 2023). xxxvi
Figure 40 Example of program output response (the author, 2023). xxxvii

1. Introduction

1.1. Context

Aviation is one of the branches of aeronautics, which can be defined as "a high-tech industry that is characterized by complexity and sophistication and that, in this way, supports a developed nation in order to ensure its technological independence" (ETTAIBI, 2021). Therefore, as aeronautics is an extremely complex science with an exceptionally large number of variables to be observed, the process of designing and manufacturing an airplane also has a high degree of difficulty. The process of developing, designing and manufacturing an aircraft involves people from different areas and, for this reason, the chance of error in any of the processes is very high, which in turn generates costs for the company that produces the aircraft. In addition to what has already been said, most of the components that make up the construction of the aircraft go through some procedure of forming material that, through a manufacturing process, becomes a useful component for some area. The application of the forming process is interesting when the components have a geometry that is moderately complex and there is a high production volume, so that the tooling cost per unit produced remains low. This resource also becomes attractive in cases where the properties and metallurgical integrity of the components are of extreme importance, as is the case of cargo aircraft, jet engines and turbine components (T. ALTAN et al., 1983).

One of the objects of study for this research is the Airbus aircraft, the A400M, which will be the instrument of information extraction so that the data and parameters of the parts inside this aircraft are able to be applied to the research field of this work or to future works in the same study area. To get a sense of how complex the manufacturing process of this object of study can become, the Airbus A400M is composed of almost 700,000 parts where all these parts are designed and manufactured in different countries around the world, as shown in Figure 1 (F.MAS et al., 2013):

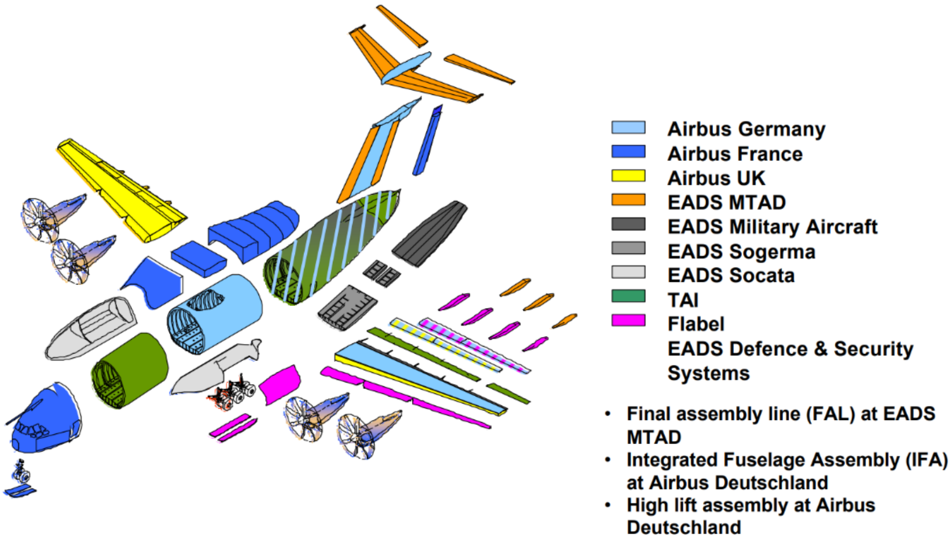


Figure 1 Industrial division of the A400M aircraft (F.MAS et al., 2013)

In the case of application and construction of this research proposal, the concept of ontology will be applied with the purpose of understanding the cost of manufacturing parts, and it can be defined as "a formal description of all entities of a domain and the existing relationships between these entities" (LAMY, 2021). The construction of the ontology will also be part of this research project, making use of the advances made by the research work group in this area, along with the background on the subject and the understanding of how the process of forming parts is applied within the aeronautical industry. The cited elements will be key pieces for the development of the final product of the work.

Furthermore, since there is a difficulty in the process of generating the necessary expenses for the manufacture of a metallic part, the result that the research will generate will be an expert software capable of calculating this manufacturing cost of metallic components, being made possible in the *Python* programming language, together with any library and function necessary for the integration of ontologies with the concepts of conformation of aeronautical parts. Besides an extremely valuable tool for the industry as a whole, there is a contribution also with the concept of *Product Design and Manufacturing Knowledge-Based System* (PDMKBs), indicated and defined in Szejka (2022), so that this concept makes the integration between manufacturing models and the "semantic Web". The software to be developed will be considered one of the examples of application of this system conception, enabling other applications within the same study axis.

1.2. Problem

The theme of Industry 4.0 brought about several changes within the general industrial market, forcing the use of classical production methods and establishing new patterns of adaptive and automatic systems and production networks, causing improvements in product quality, reduction in the amount of production cycles and cost reduction (HOZDIC, 2015). With the ability to make processes more unique and particular, the concepts of smart factories and smart manufacturing have become more common within the industrial environment for containing the application of other technologies such as the Internet of Things, artificial intelligence, cloud computing, cyber-physical systems, among others (ZHENG, 2018).

The research addresses more specifically the whole process of pricing a sheet metal part within the aeronautical industry, a procedure that sometimes has no automation or has only one of the parts automated, which makes the calculation of manufacturing cost of these components something slow. Time is a significant variable when it comes to service sales, so that agility and price are decisive in hiring or not this service. Thus, the objective of this work is to conceive and develop a solution through the creation of an automated system that enables the calculation, in an automated and fast way, of the manufacturing costs of complex parts for aviation. For this, besides the theoretical foundation in the area of part conformation, the ontology to be created must be structured containing characteristics of the part to be manufactured, the machines to be used for each process and also the other pricing variables involved in the process. With the structuring well defined, all the concepts will be applied and integrated into the programming along with the business rules created and the definition of the multiple cost parameters, the final goal of this implementation is to provide the correct result of this calculation. The validation step of the solution should be applied to sheet metal parts with different complexities and shapes, so that for each one the pricing is obtained with a plausible value.

1.3. Objectives

1.3.1. General objectives

Therefore, the general objective is initiate and design the implementation of an automated system with the function of performing the automated calculation of the manufacturing costs of sheet metal parts in the aeronautical industry based on multiple heterogeneous variables.

1.3.2. Specific objectives

For the software implementation to be well structured and for the algorithm creation and development process to be successful, some goals must be stipulated so that there is a correct delimitation of what will be worked on and developed in this project. Thus, the specific goals are:

a) Explore the literature study for understanding which variables and parameters will be relevant to the project.
b) Propose a conceptual architecture for the solution in order to validate it in real parts.
c) Create the ontology and the business rules involved.
d) Correctly apply the concepts studied within the algorithm in Python to obtain the cost.
e) Ensure that all information from the implemented ontology is organized correctly in the program.
f) Validate the system using an example of sheet metal part.

1.4. Justificative

Since the subject of design and manufacturing of an aircraft requires several process steps and that each of these steps includes the involvement of different sectors and people from different regions of the planet (F. MAS et al., 2013), the chance of errors in any of these steps occurring are great. As previously shown, the object of study of this research is the A400M aircraft, which is manufactured by several divisions of Airbus located in different countries and, because of this, in each error that happens there is the possibility that other mistakes of subsequent processes also need correction. The result of all this is more unnecessary costs for the company, since they involve, for example, the proposal of new design solutions and also all the logistics that this correction covers. According to (GEIGER, 1999), "the manufacturing of sheet metal parts in large-scale production is mainly done with progressive dies. The design and production of these tools are responsible for an essential part of the manufacturing costs", thus demonstrating that a design or project error causes more expenses, and this amount reflects in the final value of the product or reduces the company's revenue.

Given this, one of the strands of this great theme of study is based on the manufacturing procedure of aeronautical parts, more specifically those produced using sheet metal, and how the cost calculation of these parts is performed. This calculation is a complex procedure, involving different components in the composition of the total cost of the part, and it is necessary to use several tools both for the extraction and organization of information about the product to be produced as well as tools for analysis and accomplishment of this operation. According to KALPAKJIAN (2014), the manufacturing cost represents about 40% of its selling price and in the quest to reduce this rate, there is the mass use of automation and other optimization sectors so that efficiency is increased. Moreover, the end-customer satisfaction also considers that, nowadays, the product life cycles have shortened, and the trend of product varieties and complexities is increasing, thus making the operation of obtaining the cost something decisive in the product’s marketability. An example of the means to be applied in this calculation are ontologies, used to explore the characteristics that need to be extracted from the part and to gather certain key parameters for some later analysis or applicability. Moreover, its visualization serves as a simplification of ontology management, especially in the case of large and complex networks (REBSTOCK, 2008), since even with a simple structure, the correlations performed by an ontological method are extremely powerful. The correct structuring together with the constant insertion of information and new definitions would make the overall working system an essential daily use tool for pricing and with different application possibilities, since the application of ontologies in visual representation and knowledge elicitation is a way to understand the relationships created (HÉON, 2020).

Therefore, the study to be carried out has a very clear reason to be executed: the execution of the estimate of production expense of a specimen within the aeronautical industry and automation of this existing process, together with the reduction of errors. Besides what has already been mentioned, the benefit that will be reaped by all parties interested in this work is a tool that will facilitate and optimize the cost calculation process of complex parts. Furthermore, a direct advantage of the software is the reduction of errors in this process, also reducing the expenses that occur in this process.

1.5. Research methodology

In order to describe in greater detail, the development of the work, the methodology of this research is based on the description of the processes that involve the progress of the project. Thus, in order for the final product to be made, the student must first delve into the basic and reference subjects, which will be better described in the theoretical reference section. Together with the theoretical knowledge there is also the search for similar works in books, articles or magazines that can add or facilitate the research such as a function already implemented and applied to an ontology example or a cost calculation project in the area of conforming materials that has not yet been developed.

After the acquisition of all this knowledge and with the most important objectives of the research well stipulated, the next step would be starting the development of the ontology that will contain the description of manufacturing processes to make the sheet metal parts, along with a set of characteristics and the key parameters previously defined of the parts. The last step of the implementation would be for the system to return a manufacturing cost for a part with the set of variables described, applying the Python programming language to integrate all the steps and provide the description of this cost correctly.

It is important to make clear that the validation will obtain real data from the industry for the application of the implemented solution ensuring that the result of the research will be plausible, whether they are of simple or complex manufacturing, making clear the applicability of the solution in everyday life.

2. Bibliographic review

A With the intention of theoretically grounding the monograph, this section provides an explanation of the theoretical references that were used to delve into the issues. The research is inserted in the general scope of Industry 4.0, which is characterized by the insertion of the *internet of things*, cyber-physical systems and other definitions in the manufacturing context, allowing the concept of *smart factories* to be applied together with production systems (THOBEN et al., 2017). Given the various other important conceptualizations within the topic of I4.0, the digitalization that the industry sector is constantly undergoing applies one of the concepts that give context to this paper entitled smart manufacturing. It can be defined as "an emerging form of production that integrates manufacturing assets of today and tomorrow with sensors, computing platforms, communication technology, control, simulation, data-intensive modeling, and predictive engineering," in a way that still makes use of concepts such as cyber-physical systems, IoT, cloud computing, artificial intelligence, and data science (KUSIAK, 2018). What all this digital transformation aims at is an attempt to modify current systems in search of making them more adaptable and flexible, possessing the ability to learn as the course of and make autonomous decisions (F. PIRES et al., 2019).

Given the magnitude in which the research theme is inserted, explanations based on the theoretical references of the aspects that were identified as the main ones for this work will be described below. Among these aspects are the description of the automated systems concept, processes for the conformation of parts in the aeronautical industry, explanation of the general definition of ontologies, and also a description of the pricing of production cost.

2.1. Aeronautical parts forming

The process of forming metal parts in the aircraft industry is one of the initial procedures of manufacturing an aircraft, since the component to be created can vary from raw material due to the need and functionality of the part. For Stern (1962), forming can be defined as follows:

> *"Forming is one of the most important methods of metal fabrication. The principles of forming for all metals are basically the same, but for each metal there is a difference in the amount of deformation, the details, the tool design, and the annealing procedures."*

The adjustment of a given material, which normally has no shape or a simple geometry, through a manufacturing process, transforms it into a useful component that, most of the time, has a complex geometry containing size, shape, tolerances, precision, appearance and mainly its properties well defined (T. ALTAN, 1983). Given the different possibilities of forming materials, the main focus of the research is on thin metal sheets (also found as *sheet metal parts*) and, therefore, some common processes for forming these sheets can be applied such as pressing, bending and hydroforming. Regarding the first method, it is used to create three-dimensional shapes and the procedure is initiated by fixing the sheet metal on the template, which represents the final shape that the part should contain, and then the sheet is inserted in the press suffering pressure for a certain time. After that the formed part can be removed, making room for a new metal plate (MAROTO, 2005). This process can be better visualized by the illustration below:

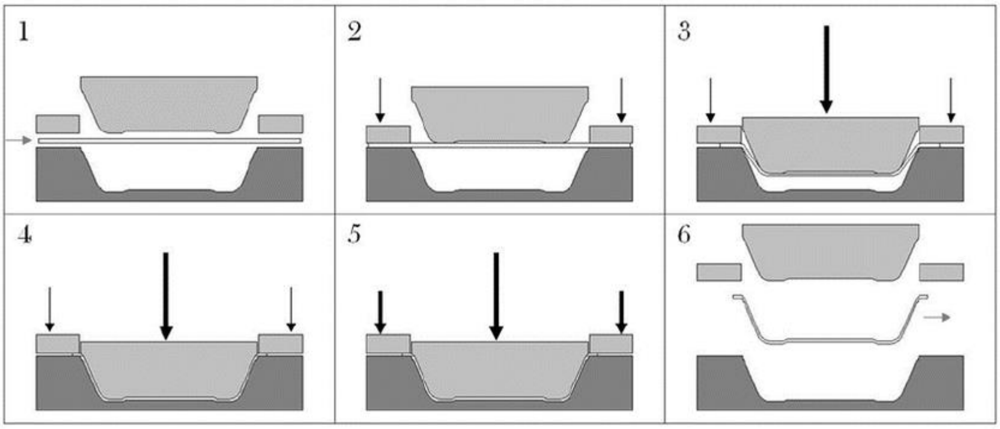


Figure 2 Press forming process (LEMINEN et al., 2015)

In forming by bending the sheet metal suffers a deformation so that this methodology is not only applied for the production of edges, curves or seams, but also aiming to increase the stiffness of the parts (MARCONDES, 2016). Below you can see two examples of the bending process, one performing a simple bend in a straight line (a) and the other performing a bend with a "V" shaped matrix (b).

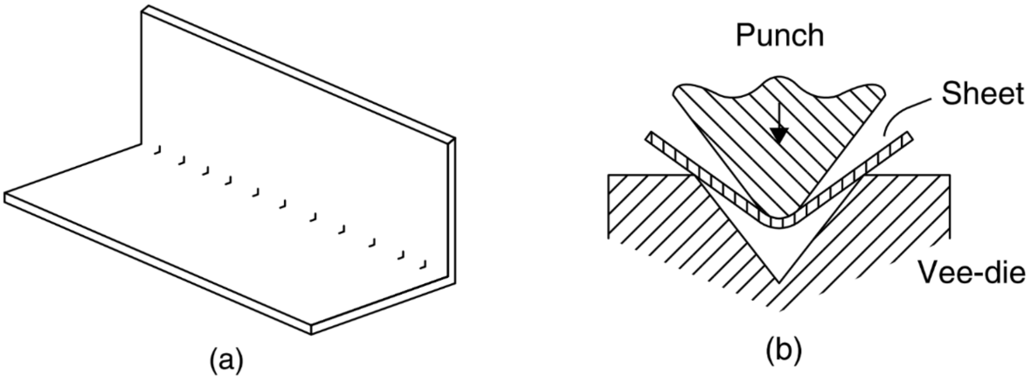


Figure 3 Illustration of the bending processes (MARCINIAK, 2002)

The third forming process to be described is called *hydroforming*, a process in which the sheet metal is formed against a matrix of the desired shape by fluid pressure (MARCINIAK et al., 2002). Compared to the conventional stamping process, hydroforming has considerable advantages such as better surface quality, less sheet metal blowback effect, better dimensional freezing, and the ability to form more complex shaped parts (ZHANG et al., 2004). Furthermore, the production of a jig is simple, and the process can be economical if there is a small number of parts. A typical metal part produced from the hydroforming procedure can be seen in the figure below (a) and a demonstration of the die and the process in operation (b):

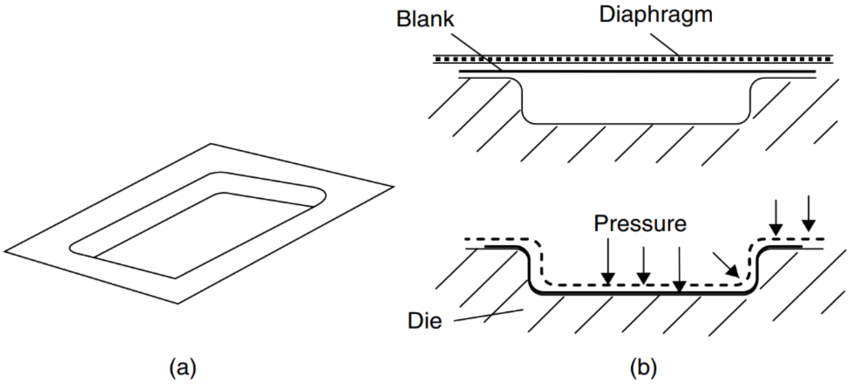


Figure 4 Illustration of the pressing and bending processes (MARCINIAK, 2002)

Hydroforming is also used to form tubular parts. With specially designed forming machines, a large number of parts can be formed by this process at low cost. Another application is the formation of tubular parts where the round tube is bent and then placed into a mold and then pressurized internally and formed into a square section (MARCINIAK et al., 2002). With the geometric shape diversities of the tubes that are hydroformed, there is a marked research and development effort being made to increase the economic efficiency of the process to address the forming mechanism and tooling system of this process (P. HEIN, 1999).

2.2. Ontologies

Besides the metal parts forming processes, one of the other theoretical pillars that the present work will use for the development of the solution are the ontologies, which can be characterized as a way of knowledge representation, making a definition between relationships and classifications of concepts within a specific domain of knowledge (LAMY JEAN-BAPTISTE, 2021). In other words, an ontology formally describes any existing relationship between the entities that are contained in a scope of knowledge and, it is this ability that makes it a powerful tool for application in conjunction with software implementation (JEPSEN, 2009).

Ontologies act in the integration of data and knowledge, because they are a means of documenting the structure of a particular domain that, by performing a shared scheme of information, they end up developing a common understanding of its already inserted concepts. However, there is no general consensus on what requirements this formal representation must meet in order to be correctly named an ontology, since they can be simple dictionaries and taxonomies, or high-level formalizations. Taking into account this naming difficulty, the construction of an ontology depends on specialized resources and that, because it is a complex task due to its information dependencies, are means that sometimes do not support each other. This formal creation will only achieve the goal of producing expected results in the future if specialized engineers familiar with the theory and practice work together with those competent in the area to be researched, because once the ontology is built, all the evolution of the parameters will require continuous efforts so that the maintenance of this information is correct (LEHMANN et al., 2014).

Throughout the process of learning and evolving an ontology, it is very common that the learnings have some uncertain and even contradictory knowledge, caused most of the time by lack of maintenance of information together with imprecision. Thus, these ontologies where there are uncertainties and inconsistencies in the data end up becoming doubtful knowledge bases due to the fact that they have to face the difficulty of solving the reasoning with a lot of non-concrete information, and thus, generating erroneous results in the ontologies generated later (HAASE, 2008).

In order to visually represent an ontology, Figure 7 shows an example of an ontology in the field of ecology. By way of explanation, the terms "*Pike*" and "*Roach*" refer to two species of fish.

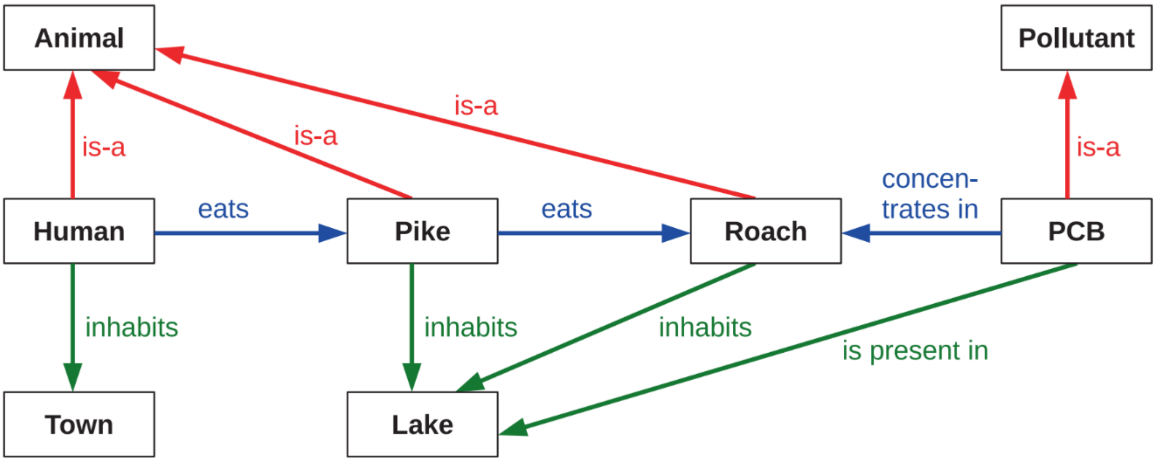


Figure 5 Example of ontology in the field of ecology (LAMY JEAN-BAPTISTE, 2021)

It is possible to see eight entities, represented by rectangles, along with the relationships that exist between these entities. First there are the hierarchical relationships, which connect one entity to another more general one, for example the human being is an animal. Another type of relationship is called geographical, indicating the location of the entity, connecting the entity to a place. For example, the fish *pike* is located in the lake. Finally, one can observe several cross-cutting relationships, such as "*eats*" or "*concentrates in*".

Observing in detail the simple ontology proposed, it is noted that *PCB* (a pollutant) can intoxicate a human. The role of the ontological method is to perform the structuring of knowledge in order to make it accessible to machines that, with the help of software, will be able to deduce the risk of a human being intoxicated for example (LAMY JEAN-BAPTISTE, 2021).

Given the general conceptualization and exemplification of ontologies, another approach that applies to the present work refers to the ontology representation language named Web Ontology Language (OWL). This representation language has some essential elements for their use in the creation of rules, which are: "classes, relations, datatype properties, object properties, individuals and restrictions". To explain a little more about the power and expressiveness of these languages, another example to be mentioned is OWL2, which consists of a more recent version of OWL together with the addition of functions for object and data properties. There are also the Resource Description Framework (RDF) and RDFS languages, which are less expressive when compared to OWL. Thus, when comparing OWL with OWL2, there is an expressive difference constructs, as the newer version has property strings and keys, specific constructs for expressing constraints and adding new property features (HEIYANTHUDUWAGE, 2016).

In order to demonstrate an implementation of an ontology in the scope of manufacturing and associate all the explanations given, the ontology "MASON" is equivalent to

"Manufacturing's Semantics Ontology" proposed by S. Lemaignan et. al. (2006) has three main classes: Entities, containing product specification concepts, the class Operations, associated with the description of the manufacturing process, and the class Resources, associated with the resources of manufacturing as a whole. This ontology can be observed in the Figure 6 below:

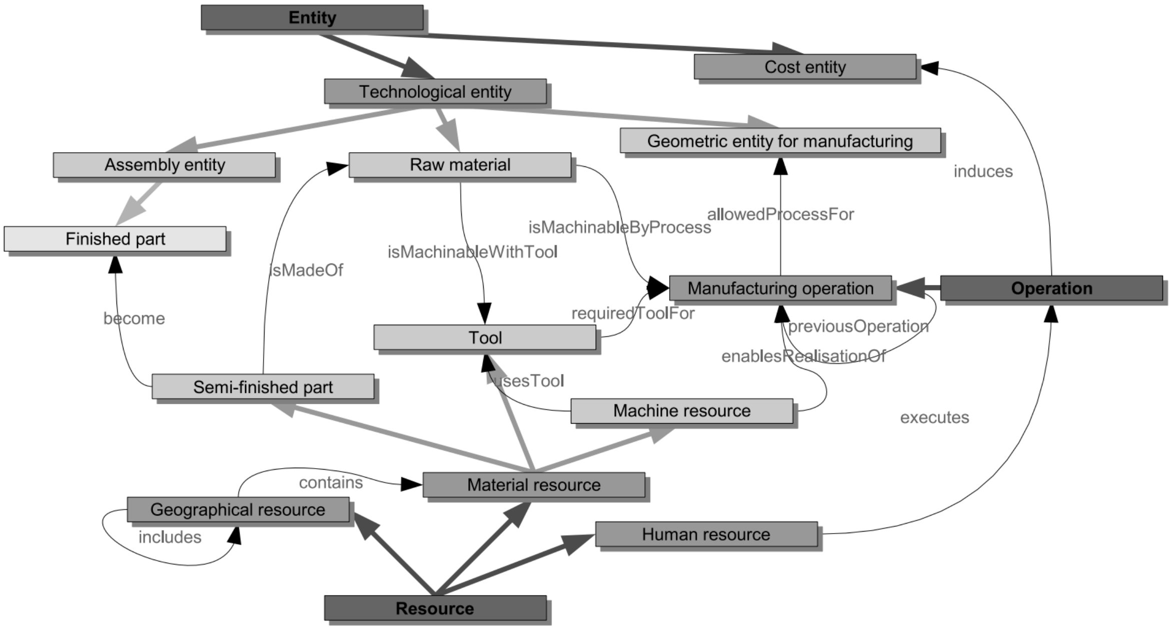


Figure 6 MASON proposed ontology (S. LEMAIGNAN et. al., 2006)

2.3. Pricing of production costs

The pricing process within the scope of manufacturing has a short timeframe for decisions, where the service provider must present an offer so that the price is attractive, because this cost is decisive (VERLINDEN et al., 2008). In an attempt to increase accuracy and facilitate management, several ways to quote for parts have already been created. One of these proposed ways of cost estimation takes into account the characteristics (called *features*) obtained from the available 3D model of the part to be manufactured. According to Geiger et al. (1999), the manufacturing cost determination for the proposed model is divided into 3 steps, being the first related to the collection of the technical and geometric characteristics of the part, which may contain the geometry, quality, material, among other variables. The return of this analysis is crucial for the basis of the quotations to be implemented.

The next step is to assign the acquired characteristics to the different manufacturing process techniques, where here the different information is transformed into manufacturing specific data. In the third step, this data is transformed into a cost estimate, and in addition to all the information obtained, at this point internal cost rates (of the company itself), cost of refinement, tooling, transportation and distribution are entered.

A model of the system proposed above can be seen in the figure below:

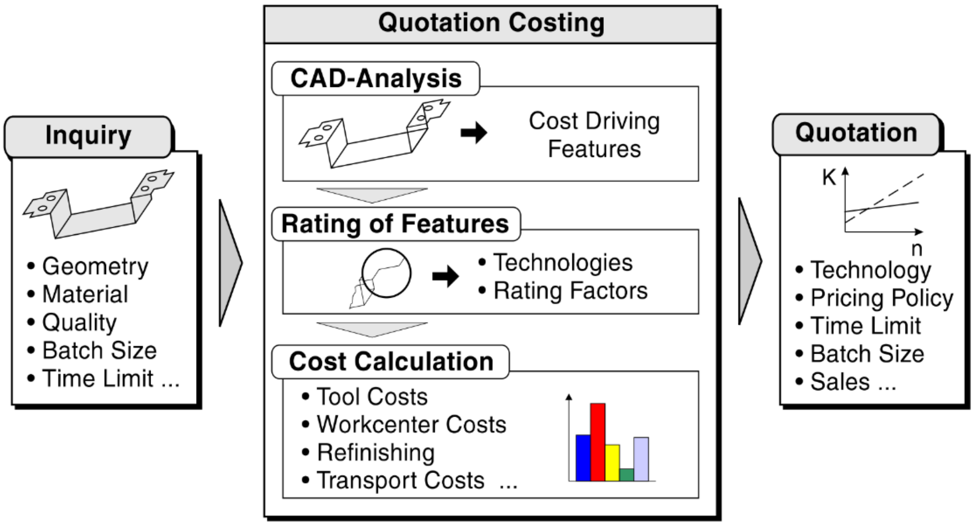


Figure 7 Proposed cost quotation for manufacturing thin sheet metal (GEIGER, 1999)

In order to recognize the information analyzed in the mentioned study for a sheet metal part, Geiger (1999) considers for the pricing proposal the following items as geometric and technological characteristics:

> *"Edge contour shapes (such as notches and grooves), internal contour shapes (standard openings such as round, square and rectangular holes and particular profiles that require special drilling dies), bends, joining techniques (screwing, riveting, welding (spot), soldering and gluing), edge shapes (bending, double bending, rolled edges, with an insert if needed), gills and beads (open and closed)."*

After the description of an automated process for estimating the cost of manufactured parts, in most cases this described process does not occur in its entirety, often not being applied any automation. In these cases, to perform this cost estimation in sheet metal, all the information has to come from the manufacturing process planning, which comprises more details about the part, what are the production and process steps, tolerances, among other data (VERLINDEN et al., 2008). But often what happens is that the whole procedure of conversion and organization of this data takes an amount of time not compatible with the available for decision making at the time, because the customer who is requesting the service is waiting almost at the same time the quotation of the cost of manufacturing the part.

3. Development

Given all the theoretical structuring of the research approach together with the explanation of the main important points and objectives, it is necessary to formalize a proposal that comprises the concepts defined previously together with the solution of the problem presented. Thus, the proposed solution to computerize the procedure for obtaining the manufacturing cost of metallic parts for the aeronautical industry would be an automated system that, through specific characteristics and parameters collected from the part, can calculate the manufacturing cost for these components to be manufactured. For this to be possible, several steps should be accomplished before the final validation in real parts, steps involving the correct structuring of the ontology in conjunction with programming for information synthesis, and the correct description and definition of the key parameters to be inserted.

In order to better describe this planning, a conceptual organizational structure of the present project was created, contemplating, at first, a simplistic structure of what should be done, containing objectives and goals of what is intended to be done, considering implementation time and the whole system structure design. This section describes in detail the entire process of designing and creating the proposed system.

The conceptual architecture can be visualized in Figure 8 embedded below.

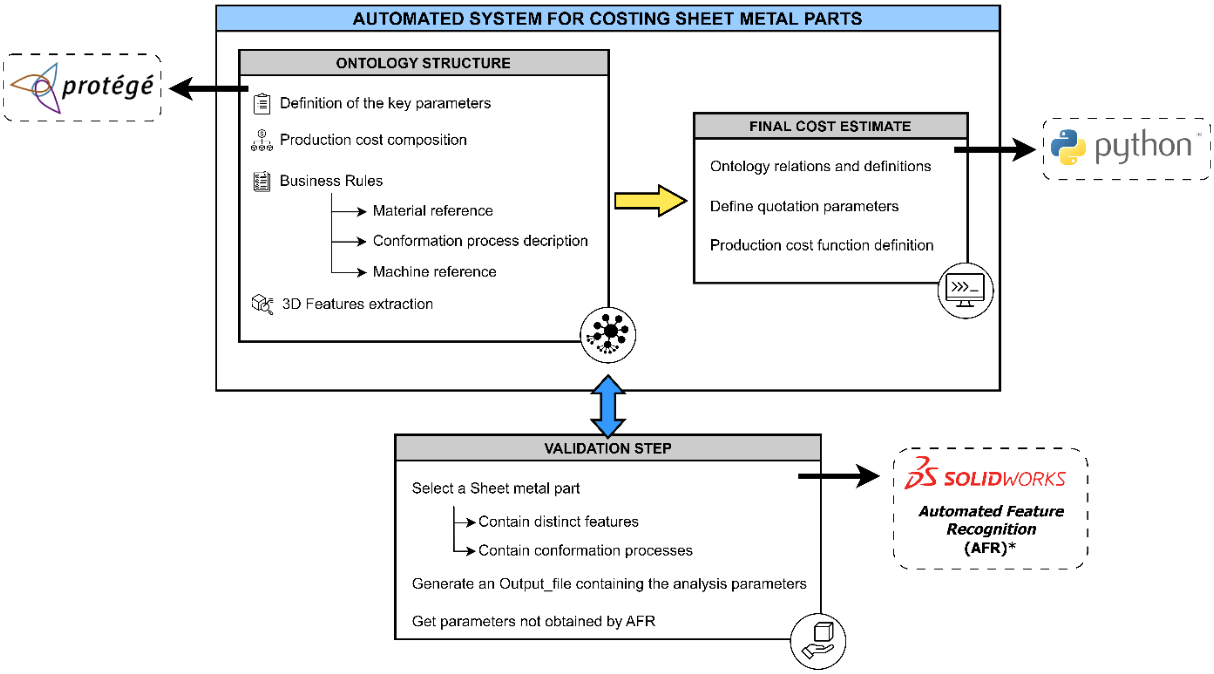


Figure 8 Conceptual architecture of the implementation (the author, 2023)

3.1. 3D Features extraction

Considering that this research addresses the beginning of the automated system design for the pricing of metal parts in the aeronautical industry, the central theme is inserted in a much larger scope and, thus, it was possible to obtain tools that assist and perform the extraction process of all the information from a 3D part in the .step format. The tool used is called Automated Feature Recognition (AFR) and is applied to identify features of parts from the STEP 3D model of sheet metal applied to aeronautical structures, performing the pre-processing of the model to classify the elements and thus create the sets of faces, subtypes, geometry, recognize the relevant parameters and the typical features of aerospace sheet metal (GHAFFARISHAHRI, 2023).

Besides the entire program for the extraction of the features from the 3D model, the taxonomy proposed by Ghaffarishahri (2023, p.89-91) for the characterization of ASM features was also applied to the development of the ontology. This taxonomy divides the features of the part into 3 major features: Base Feature, where all other features are built or modified; the Contact Features, which have functions of attachment and adjustments, and has a relationship with interfacing the parts and providing the contact; and finally, the Refinement Features, which are created for a purpose or a combination of them, such as reinforcement, weight reduction, among other utilities. The Figure 9 demonstrate the division of the information explained before.

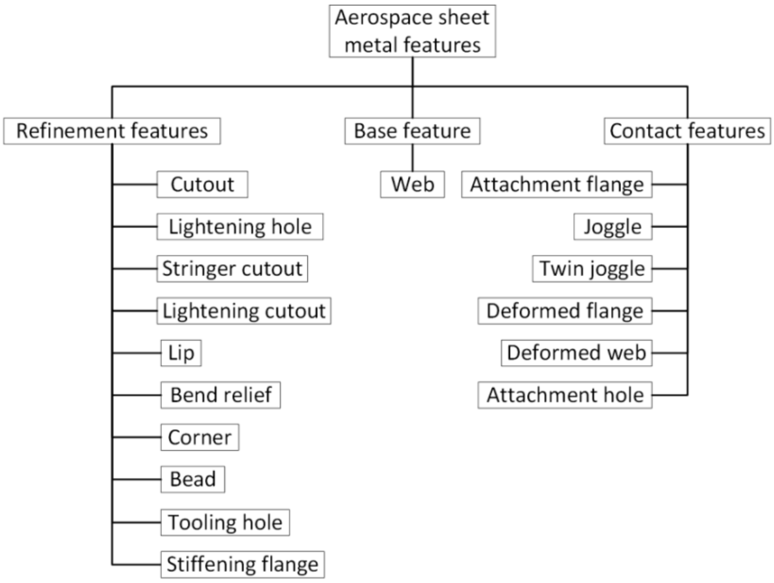
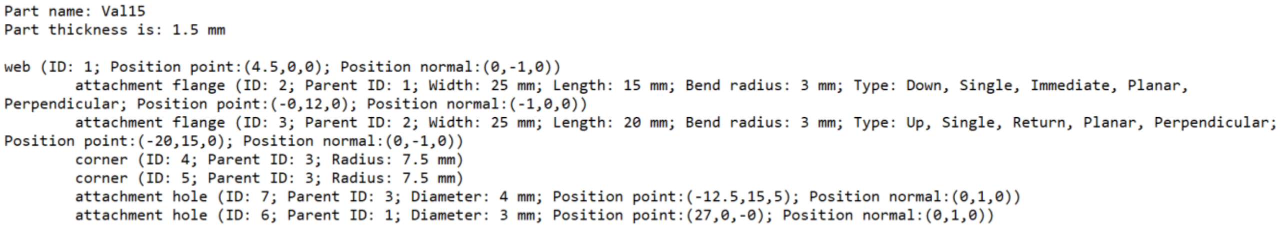


Figure 9 The taxonomy of feature types proposed (GHAFFARISHAHRI, 2023)

Applying the AFR algorithm in a STEP model, the output file of this tool is a .txt format file that has all the characteristics of the part entered for the analysis. One of these output files is represented below:



```
Part name: Val15
Part thickness is: 1.5 mm

web (ID: 1; Position point:(4.5,0,0); Position normal:(0,-1,0))
        attachment flange (ID: 2; Parent ID: 1; Width: 25 mm; Length: 15 mm; Bend radius: 3 mm; Type: Down, Single, Immediate, Planar,
Perpendicular; Position point:(-0,12,0); Position normal:(-1,0,0))
        attachment flange (ID: 3; Parent ID: 2; Width: 25 mm; Length: 20 mm; Bend radius: 3 mm; Type: Up, Single, Return, Planar, Perpendicular;
Position point:(-20,15,0); Position normal:(0,-1,0))
        corner (ID: 4; Parent ID: 3; Radius: 7.5 mm)
        corner (ID: 5; Parent ID: 3; Radius: 7.5 mm)
        attachment hole (ID: 7; Parent ID: 3; Diameter: 4 mm; Position point:(-12.5,15,5); Position normal:(0,1,0))
        attachment hole (ID: 6; Parent ID: 1; Diameter: 3 mm; Position point:(27,0,-0); Position normal:(0,1,0))
```

Figure 10 Example of the software output file (the author, 2023)

As can be observed, each of the features has different parameters and are separated and organized by a level of importance and, also, follows a sequence. For example, the parameter named "Attachment flange" has an identification property "ID" with a value of 2 and a "Parent ID" with a value of 1, representing that this parameter has a relationship with the "ID" parameter equal to 1, i.e., the "Web". Within "Attachment flange" it is still possible to identify other properties with their respective information, which are essential for further use in the automated system.

By obtaining the characteristics of the 3D models, for the subsequent application of the final version of the solution, the part named by “Val15” were selected to demonstrate the applicability of the solution provided. Below is the 3D demonstration of these model:

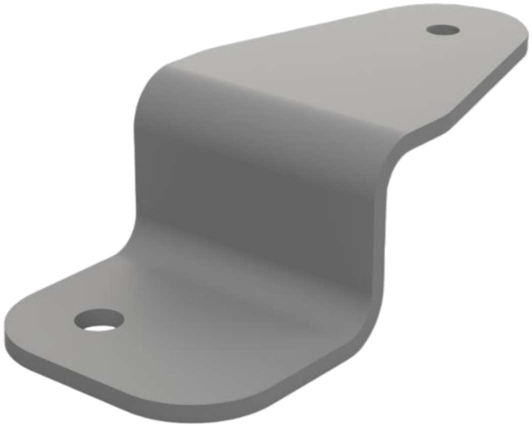


Figure 11 Step model named “Val15” (the author, 2023)

3.2. Business rules

The business rules were implemented together with the ontology structuring seeking a greater correlation of the part chosen as a test model with the cost definitions. According to Chisholm (2003), a business rule can be defined as:

> *"Business rules are statements that define how an organization operates. They are used to control how data is processed and what actions can be taken. Business rules can be expressed in a variety of ways, including natural language, code, or graphical notation."*

3.2.1. General definition of all involved variables

After defining the concept, the first step was to understand the manufacturing process of the part chosen for validation, concluding that two processes would be done: laser cutting and sheet metal bending. The laser cutting process can be quoted taking into account the cutting perimeter, whether this operation is a hole (requiring the calculation of the circumference perimeter) or the cutting of the entire piece for this application, so that this whole procedure has as its main variable the cutting speed, which varies according to the specifications of the machine to be used and the material and, moreover, has as unit of measurement meters per minute (mm/min). To make the sheet metal bends, the process must consider a bending matrix tool with a radius value in millimeters (mm). In view of these definitions, the first procedure can be priced in cost per hour (R$/hour) considering what is the material for performing the operations and, for the second, the cost is also priced in the definition of the value per millimeter to be folded (R$/mm).

Besides the characteristics of each of the manufacturing processes, another determining factor in the price is the specification of the type of material to be used. The price is commonly given in cost per weight of material (R$/kg), with the need to calculate the weight of the part to obtain the final value. For the applicability of the proposed solution to the 3D part, only the metals class was considered since, given the various alloys that exist and are used, it is important to note that each alloy changes the parameters for performing these operations, changing the final value of the product, as well as the number of parts to be produced. For the quotation of the materials, three steels commonly used in general industry were quoted for this example. Below are the price values by weight for these materials:

Table 1 – Description of the stipulated price by weight of steel types.

Steel	Value [R$ per kg]
ASTM A36	14,30
SAE 1010	15,50
SAE 1020	23,00

Source: the autor, 2023

Other defined pricing parameters that must be defined are the fixed, variable and also semi-variable Overhead costs, the price of Product Finishing processes, such as heat treatment and also surface treatment processes for metals. Besides these parameters, another cost with a high relevance for the pricing process is the setup preparation cost, involving machine preparation and programming, number of component reorientations, among other specifications. For the final and definitive version of the solution, the parameters cited above were not implemented at the same level of complexity as the other pricing parameters but received the correct domain definition within the ontology and description of the data type.

For a better visualization of the definitions described above, all parameters were stipulated and are shown in the table below.

Table 2 – Variables and parameters needed for the definition of business rules.

PARAMETER	UNIT
Cost per Hour - General	R$/hr
Cost per milimiter Bended - Bending Machine	R$/mm
Cutting Speed - Laser Cutting Machine	mm/min
Setup - Pre-production and Production costs	R$/Quantity
Overhead Cost	R$
Product Finishing Cost	R$
Cost per Weight of Material	R$/kg
Part Perimeter	mm
Part Weight	mm
Quantity	-

Source: the autor, 2023.

A sheet thickness value of 1.5 mm was defined for the validation step, and consequently the estimated cost per hour for laser cutting was R$715/hr with a cutting speed of 141 m/min. For bending, the estimated cost per hour was R$240/hr and a cost per millimeter of length bent of R$0,05/mm. The estimated setup cost was thought of more broadly, considering a value of R$160 total for both types of setup cost (Pre-production plus Production cost). This cost is spread over the quantity of parts to be manufactured so that, if the quantity is equal to 1, the cost of a part will be the cost of the processes involved added to the R$160 of the setup. Now, if the quantity is 50, the setup cost for each part is diffused, making the addition in the final price of the part R$3,20 for the example given.

As previously explained, the cost effectiveness of the product finishing types and the Overhead costs were not implemented, since they were considered null values. For the first type, the option chosen for the finishing type would be "None", symbolizing that the part should be delivered to the requester raw, which means without any treatment. As for the Overhead costs, it was stipulated as R$0,00 so that it has no influence on the calculation. However, even with the non-interference of these factors in the final price (for this specific case), the final price calculation considers the presence of these variables.

3.2.2. Definition of the rules

With the detailing of the general definitions explained in the previous subsection, the main equation of application in the proposed system is the Production Cost of a unit of the desired product. This result is the sum of all the described cost variables that are put together in large groups, these being: the costs of the Manufacturing processes, the Setup costs, the Overhead costs, and the costs of Finishing the part. Thus, the calculation can be performed following the subsequent.

$$\boldsymbol{Product\ Cost} = Manufacturing\ Processes + Setup + Overhead \\ + Product\ Finishing$$

The pricing of the cost of a cut on the laser cutting machine involves the application of some variables already explained, being the Perimeter, the Cutting Speed of the machine, the Quantity of pieces desired and the Cost per Hour of the machine divided by 60 to transform into Cost per Minute and enable the realization of the final cost of the laser cutting process. The equation below demonstrates the explained calculation.

$$\boldsymbol{Laser\ Cutting\ Cost} = \left[\left(\frac{Product\ Perimeter}{Cutting\ Speed}\right) * Quantity\right] * \frac{Cost\ Per\ Hour}{60}$$

As for the cost of operating the bending machine, it comprises the definition of the Cost per Bend (CB) for the defined radius bending tool and also the Bend Width (BW) of the bend to be made. The final cost result for the bending processes comes from the multiplication of these parameters in relation to the total number of bends (n), as shown as follows.

$$\boldsymbol{Bending\ Cost} = (BW\ 1 * CB\ 1) + \cdots + (BW\ n * CB\ n)$$

All this information was estimated according to a quotation made by the student, so that he could understand more clearly some of the pricing processes. The main rules implemented are directly related to the part to be manufactured and the processes involved, seeking a simple and functional rule definition to demonstrate the approach and general operation of the conceived system.

3.3. Ontology development

For the stage of structuring and designing the base ontology of the solution, the modeling was done in the Protégé tool, which is a software able to create and model ontologies in OWL (Web Ontology Language), language chosen for the development of this stage. With all the definitions made inside the ontology structure, the preliminary validation step was done to verify and certify that the ontology is coherent and has no inconsistencies in the relations created. So, on the "Reasoner" tab inside the software there is the “Pellet” option, which was the reasoner chosen for determining inconsistencies in the ontology. So, to do this validation and certify this process, the reasoner did not return any cohesion errors to the user.

In order to better organize the description of what was created, this subsection was divided by the main parameters of an ontology.

3.3.1. Class definition

To mark the beginning of the development of the proposed automated system it was necessary to create the ontology, which was made possible through the Protégé tool. Based on the structure explained in the bibliography section, the ontology created has four main classes:

- The `Product_Costs_Composition` class, comprising all the costs of the process.
- The `Product_Data` class, coming from the 3D model.
- The `Machining_Reference` class, containing the main information about the machinery to be used.
- The `Material_Reference` class, describing the material types.

All these classes are created from an ontology base class named "owl:Thing", so they become subclasses of it. The overall structure of the ontology is given by the representation named OntoGraf used for the automatic structuring organization given below.

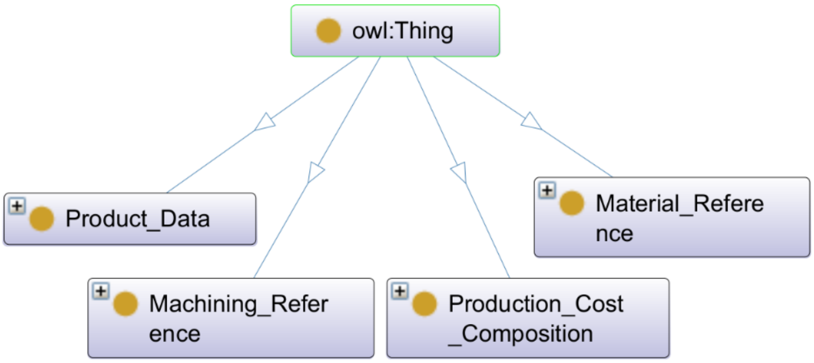


Figure 12 OntoGraf of the main classes of the ontology (the author, 2023)

3.3.1.1. Product_Data class

This class describes the information of the product to be manufactured, since it applies the taxonomy proposed by Ghaffarishahri (2023) aiming at a better organization and description of the data and, moreover, it maintains the 3 major features described in section 3.1: `Base Feature`, `Contact Feature` and `Refinement Features`.

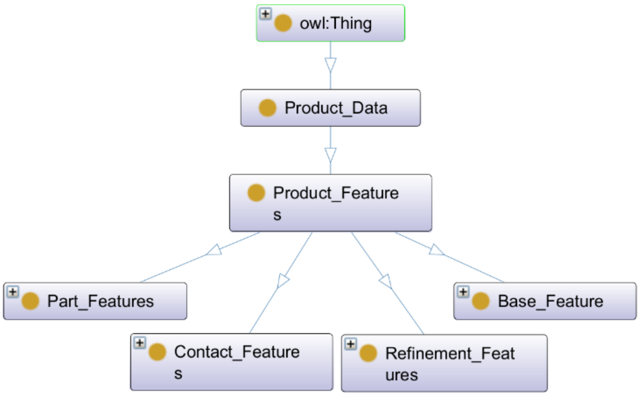


Figure 13 Product_Data subclass demonstration (the author, 2023)

Furthermore, since the program for extracting the characteristics of the 3D models was adopted by obtaining the properties of the part, to have an integration between this data and the ontology, the same structure was maintained with the addition of the `Part_Features` class, which contains other important definitions necessary for the cost calculation, among them the weight of the part (`Part_Weight`) and the total perimeter (`Part_Perimeter`), along with the name of the model that will be manufactured (`Part_Name`) and its thickness (`Part_Thickness`), as shown below.

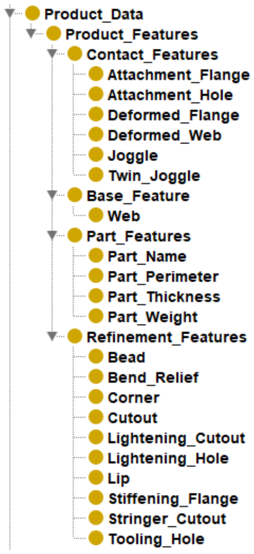


Figure 14 Product_Data and Product_Features subclasses detailed (the author 2023)

3.3.1.2. Machining_Reference class

The second main class to be described is the one that involves the referencing of machines, since all the processes that the parts must go through have the involvement of machines for their manufacture. Starting from the ontology modeled by Szejka (2016), the initial structuring form of this machine reference was maintained in order to separate the types of machines for manufacturing applied to the metal forming process. Thus, this set contains seven subclasses being one defined as "*Other*" and six main machines being: `Laser_Cutting_Machine`, `EDM_Machine`, `Milling_Machine`, `Lathe_Machine`, `Bending_Machine` and `Drilling Machine`. In addition, example machines have been inserted into some of the classes for the demonstration of the correct application of the designed layout.

In view of these descriptions, Figure 15 represents by means of an OntoGraf the definitions described above.

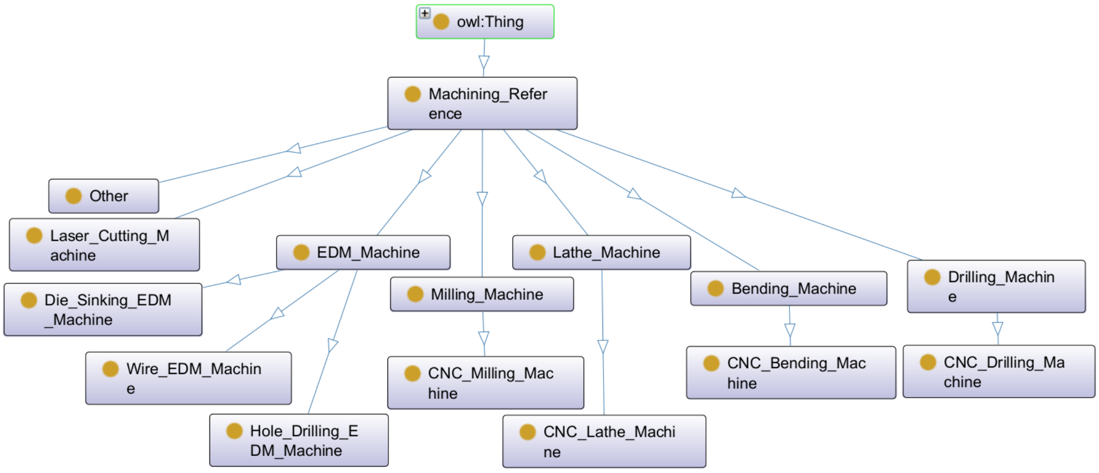


Figure 15 Machining_Reference OntoGraf representation (the author, 2023)

3.3.1.3. *Material_Reference class*

Given the various types of existing materials, for the correct modeling of the materials reference, it was applied one of the structures defined by Ashino (2010) named "Materials Information" and it was also the main materials ontology infrastructure for the development of the "Material Core Ontology", one of the steps cited by Szejka (2016) in his thesis. For sheet metal, where the research is inserted, only the class of metals was defined in greater detail and, to demonstrate in conjunction with the analysis and results section, three types of steels were modeled: ASTM A36 Steel, SAE 1010 and SAE 1020 steel, as shown in Figure 16:

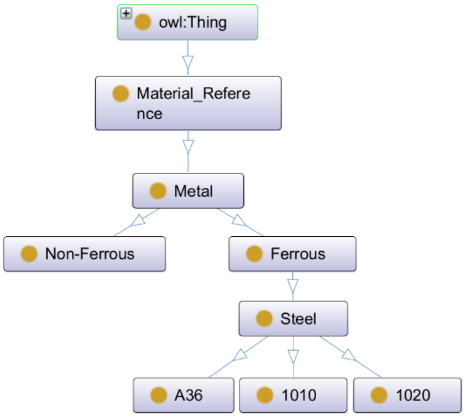


Figure 16 Material_Reference class demonstration (the author, 2023)

3.3.1.4. *Product_Cost_Composition class*

The last class to compose the set of main classes contains the cost definitions to produce the desired product. In order to better describe and separate the costs involved for this implementation, four different subclasses were created named: `Overhead_Cost`, `Pre_Production_Costs`, `Production_Cost` e `Post_Production_Costs`.

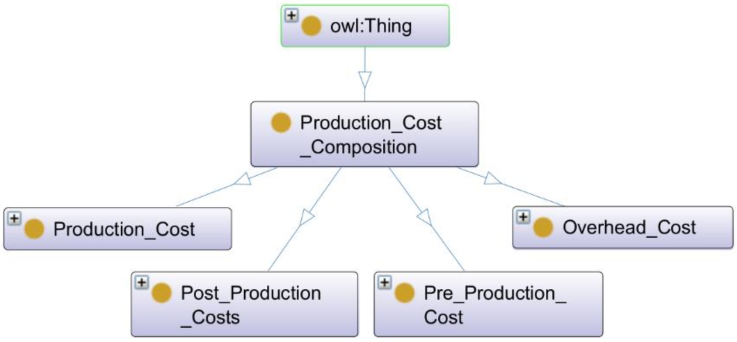
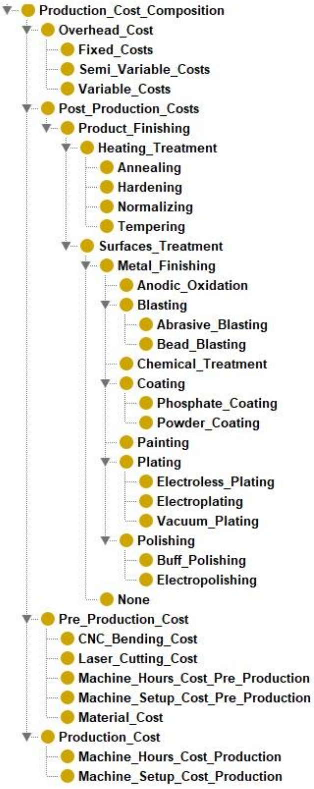


Figure 17 Product_Cost_Composition OntoGraf subclasses (the author, 2023)

Starting with the definition of the subclass `Overhead_Cost`, it covers overheads linked to business activities, being subdivided into fixed, variable and semi-variable costs. The `Pre_Production_Cost` sub-class establishes the cost ratio for the preparation of the material, that is, to make it ready to be shaped. `Production_Costs` involves the costs involved in the actual production of the desired part. Finally, the `Post_Production_Cost` sub-class defines the finishing processes, subdivided into heat treatments or surface treatments, such as metal finishing.

Figure 18 presents in more detail all the subclasses involved in the composition of the production cost of a certain part.



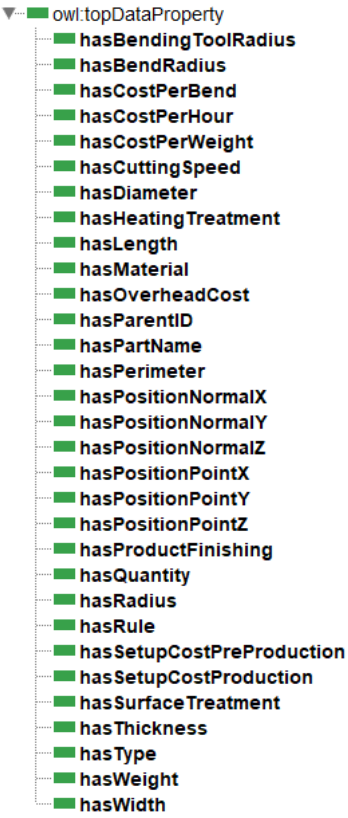
- Production_Cost_Composition
  - Overhead_Cost
    - Fixed_Costs
    - Semi_Variable_Costs
    - Variable_Costs
  - Post_Production_Costs
    - Product_Finishing
      - Heating_Treatment
        - Annealing
        - Hardening
        - Normalizing
        - Tempering
      - Surfaces_Treatment
        - Metal_Finishing
          - Anodic_Oxidation
          - Blasting
            - Abrasive_Blasting
            - Bead_Blasting
          - Chemical_Treatment
          - Coating
            - Phosphate_Coating
            - Powder_Coating
          - Painting
          - Plating
            - Electroless_Plating
            - Electroplating
            - Vacuum_Plating
          - Polishing
            - Buff_Polishing
            - Electropolishing
        - None
  - Pre_Production_Cost
    - CNC_Bending_Cost
    - Laser_Cutting_Cost
    - Machine_Hours_Cost_Pre_Production
    - Machine_Setup_Cost_Pre_Production
    - Material_Cost
  - Production_Cost
    - Machine_Hours_Cost_Production
    - Machine_Setup_Cost_Production

Figure 18 Product_Cost_Composition subclasses detailed (the author, 2023)

3.3.2. Object and Data properties definition

Besides the definitions of classes and subclasses, for the ontology to work correctly, it is necessary to define the rules for the relationship between the classes themselves and between the elements of each class. Thus, inside Protégé the tab "Object Properties" receive all the rules structuring for the class-to-class relationships. In the ontology modelled by the student, no object properties have been created.

Within the Protégé entities there is a "Data Properties" tab that defines the relationships within the classes designed in the structuring. Below is the representation of all the rules designed.



owl:topDataProperty
hasBendingToolRadius
hasBendRadius
hasCostPerBend
hasCostPerHour
hasCostPerWeight
hasCuttingSpeed
hasDiameter
hasHeatingTreatment
hasLength
hasMaterial
hasOverheadCost
hasParentID
hasPartName
hasPerimeter
hasPositionNormalX
hasPositionNormalY
hasPositionNormalZ
hasPositionPointX
hasPositionPointY
hasPositionPointZ
hasProductFinishing
hasQuantity
hasRadius
hasRule
hasSetupCostPreProduction
hasSetupCostProduction
hasSurfaceTreatment
hasThickness
hasType
hasWeight
hasWidth

Figure 19 Data Properties implemented (the author, 2023)

The properties created have a domain of where they can be applied inside the classes and, also, what "range" has to be defined, which would be an expression or more commonly applied, a data type. For a clearer demonstration of where each of these definitions are applied, they have been separated into large groups based on the main classes.

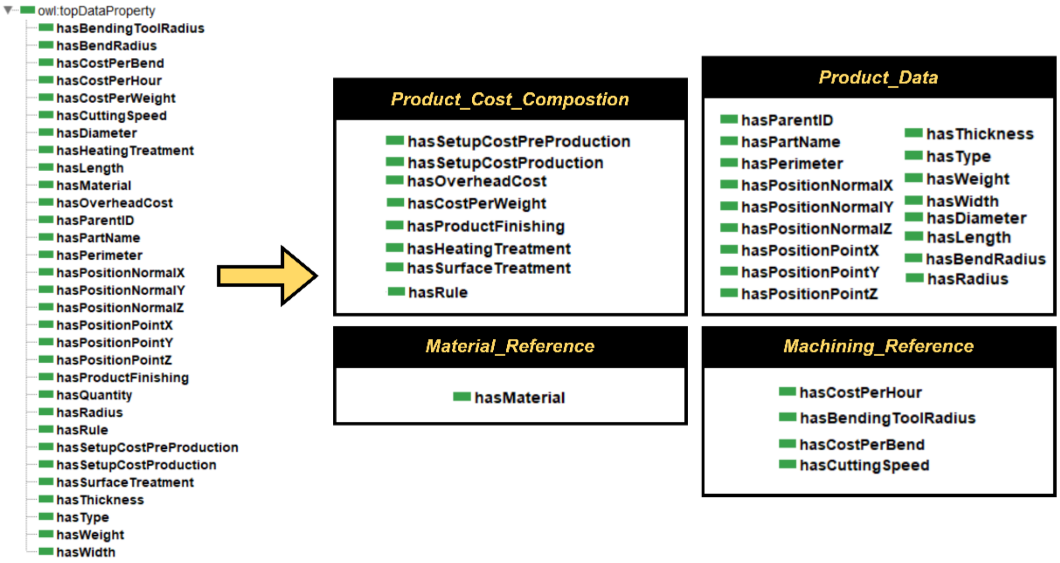


Figure 20 Separation of Data Properties in main classes design (the author, 2023)

Given this better described separation, an example application was created aiming to separate some of the information from the AFR algorithm as a demonstration. Thus, it can be observed that the parameters "Part name" and "Corner" are represented the way they are in the algorithm's output file and, for the integration of this information with the whole ontology structuring to be possible, data properties must be employed.

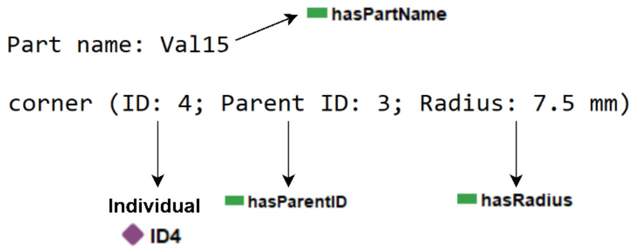


Figure 21 Example of application of properties (the author, 2023)

The data properties needed for pricing were implemented along with the whole data type definition, which also has a high relevance for the program due to the accessibility of the information and integration with the OWL library in Python.

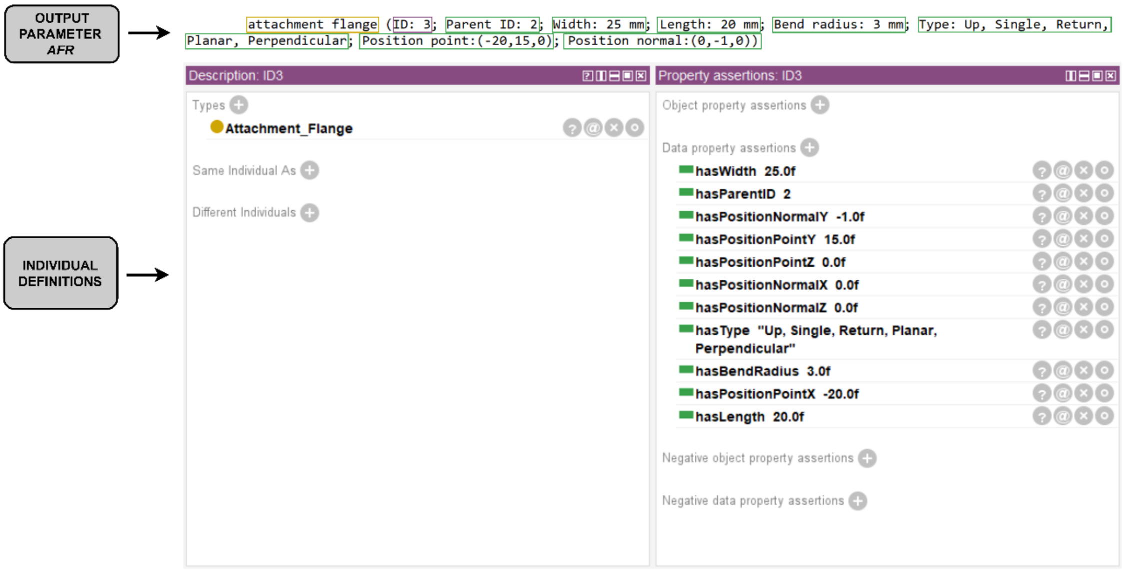
3.3.3. Ontology Individuals

The insertions of the values presented in the application demonstration are described within the individuals, so that this description includes the information from the AFR output file with the part features, the definition and parameters of the machines that are used for each process contained in the product information and definitions. All the data insertion so that the program can collect them and apply them to functions, variables or perform any kind of operation is done through the individuals, since the way this information was modeled (with several different individuals containing different properties) addressed this use of several product definitions for the integration with the programming, better described in future sections. Figure 22 below indicates all the individuals implemented.

◆ BendingMachine
◆ ID1
◆ ID2
◆ ID3
◆ ID4
◆ ID5
◆ ID6
◆ ID7
◆ LaserMachine
◆ Other_Product_Definitions
◆ Product_Finishing
◆ Product_Material
◆ Product_Name
◆ Product_Perimeter
◆ Product_Pre_Production_Costs
◆ Product_Production_Costs
◆ Product_Quantity
◆ Product_Thickness
◆ Product_Weight

Figure 22 Elements of the "Individual" type (the author, 2023)

As a form of demonstration, the individual `ID3` was selected as an example. It is one of the return parameters that the Features file contains and has as defined class an `Attachment_Flange`, which has all the properties described by the "Data Properties" section that have this class as a domain. So, below we have an association of the information from the output file together with the description of the individual, certifying that all the information is contained and available for access.



```
attachment flange (ID: 3; Parent ID: 2; Width: 25 mm; Length: 20 mm; Bend radius: 3 mm; Type: Up, Single, Return,
Planar, Perpendicular; Position point:(-20,15,0); Position normal:(0,-1,0))
```

Attachment_Flange

hasWidth 25.0f
hasParentID 2
hasPositionNormalY -1.0f
hasPositionPointY 15.0f
hasPositionPointZ 0.0f
hasPositionNormalX 0.0f
hasPositionNormalZ 0.0f
hasType "Up, Single, Return, Planar, Perpendicular"
hasBendRadius 3.0f
hasPositionPointX -20.0f
hasLength 20.0f

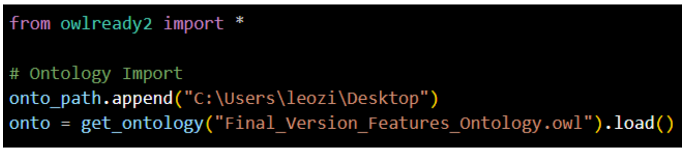
Figure 23 Output information associated with Individual definitions (the author, 2023)

3.4. Obtaining the cost for the part

a The last step of the design for the pricing resolution is to effectively obtain the final price of the 3D model. For this sage, a Python file was implemented including all the processes already demonstrated: the inferences obtained from the ontology created, the business rules containing the general cost and of each operation applied to the parts and the model characteristics. For this, the script must work with the multivariable parameters and, given the insertion in the code, provide the final price in Reais (R$) of the part.

The creation started with the process of defining all the necessary libraries together with the importation of the ontology containing all the information defined by the structuring done in Protege.

For the implementation of the ontology within the Python programming language the *owlready2* library was used resorting to some definitions made in LAMY JEAN-BAPTISTE (2021) containing instructions of the data import, definitions of the relations between the elements of the ontology and other references. Thus, the import of the ontology became possible through the *get_ontology()* command in conjunction with the *load()* parameter, since the file in .owl format was saved locally. The ontology was loaded as shown in the following figure:



```
from owlready2 import *

# Ontology Import
onto_path.append("C:\Users\leozi\Desktop")
onto = get_ontology("Final_Version_Features_Ontology.owl").load()
```

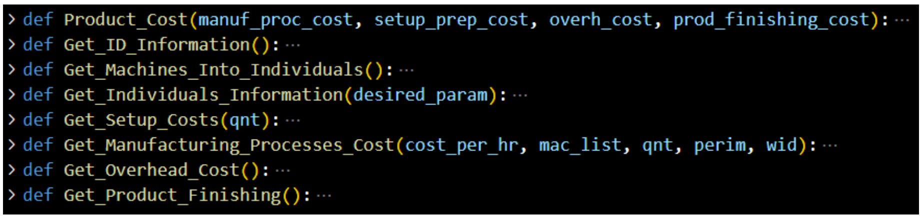
Figure 24 Loading the ontology into the program (the author, 2023)

With this importation done, the next step was the definition of functions to be used in the code that have the objective of synthesizing the obtaining of certain information so that, in any part of the program, these definitions can be called. After the functions are created, the program continues with the definition of parameters and variables necessary for the correct application of the defined functions, and there are also some preliminary verifications to confirm if the set of data imported from the ontology contains all the essential properties for the definition of the product's final cost. Finally, with almost all the parameters already stipulated, the final part of the program contains the call of the necessary functions and the main function, which would be the pricing of the part structured in the ontology.

All these steps will be better described in the following subsections.

3.4.1. Detailing the created functions

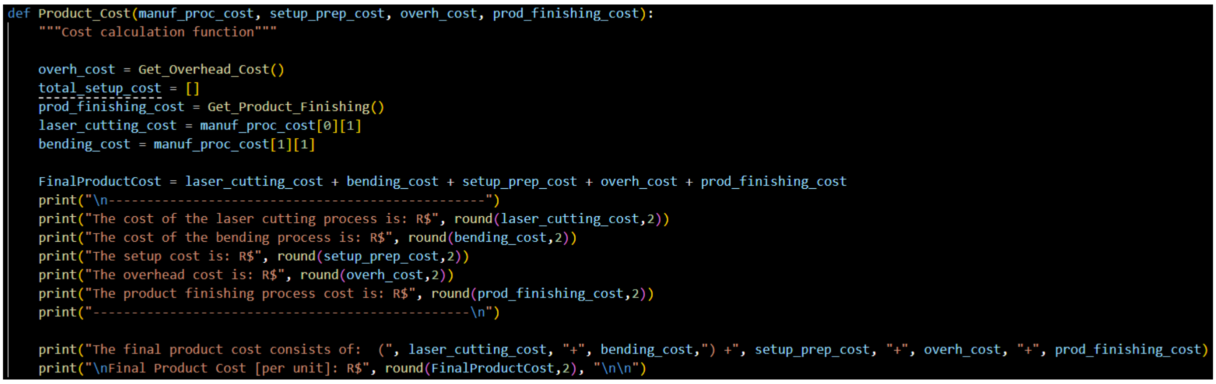
As already described, eight functions were implemented that relate directly to the data coming from the ontology and in this subsection each one of them will be better described and detailed. Below are the names of all these functions created in the program developed.



```
> def Product_Cost(manuf_proc_cost, setup_prep_cost, overh_cost, prod_finishing_cost): ···
> def Get_ID_Information(): ···
> def Get_Machines_Into_Individuals(): ···
> def Get_Individuals_Information(desired_param): ···
> def Get_Setup_Costs(qnt): ···
> def Get_Manufacturing_Processes_Cost(cost_per_hr, mac_list, qnt, perim, wid): ···
> def Get_Overhead_Cost(): ···
> def Get_Product_Finishing(): ···
```

Figure 25 Functions created in the algorithm (the author, 2023)

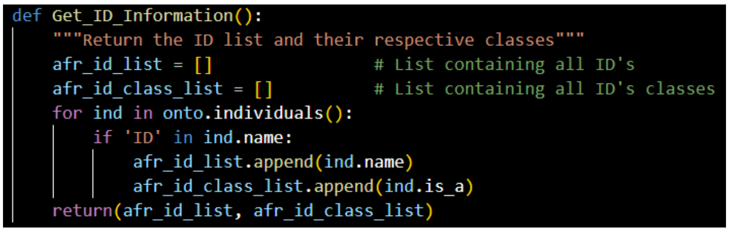
To begin the descriptions of the designed functions, the function named *Product_Cost* has the role of accomplishing the objective of the whole implementation: to show the final production cost for a unit of the part inserted in the program, with the help of all the rest of the created functions. This function receives as parameters the elements for the calculation of the final cost that come from other functions of the program and performs the operation of summing them, presenting to the user in an organized and separate way how the calculation and the final result are made. Below you can see the details of the function's implementation.



```python
def Product_Cost(manuf_proc_cost, setup_prep_cost, overh_cost, prod_finishing_cost):
    """Cost calculation function"""

    overh_cost = Get_Overhead_Cost()
    total_setup_cost = []
    prod_finishing_cost = Get_Product_Finishing()
    laser_cutting_cost = manuf_proc_cost[0][1]
    bending_cost = manuf_proc_cost[1][1]

    FinalProductCost = laser_cutting_cost + bending_cost + setup_prep_cost + overh_cost + prod_finishing_cost
    print("\n-------------------------------------------------")
    print("The cost of the laser cutting process is: R$", round(laser_cutting_cost,2))
    print("The cost of the bending process is: R$", round(bending_cost,2))
    print("The setup cost is: R$", round(setup_prep_cost,2))
    print("The overhead cost is: R$", round(overh_cost,2))
    print("The product finishing process cost is: R$", round(prod_finishing_cost,2))
    print("-------------------------------------------------\n")

    print("The final product cost consists of:  (", laser_cutting_cost, "+", bending_cost,") +", setup_prep_cost, "+", overh_cost, "+", prod_finishing_cost)
    print("\nFinal Product Cost [per unit]: R$", round(FinalProductCost,2), "\n\n")
```

Figure 26 Definition of the Product_Cost function (the author, 2023)

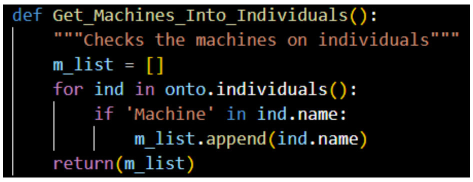
The next function developed was called *Get_ID_Information* in order to get more detailed information about the list of ID's coming from the output file of the Automated Feature Recognition (AFR) tool for feature extraction. What is done when the function is called is a search on the ontology Individuals that have "ID" in their nomenclatures, saving this data in a list named *afr_id_list*. Taking advantage of the fact that the loop already has this information, the function also returns the classes of these individuals through another list called *afr_id_class_list*.



```python
def Get_ID_Information():
    """Return the ID list and their respective classes"""
    afr_id_list = []                # List containing all ID's
    afr_id_class_list = []          # List containing all ID's classes
    for ind in onto.individuals():
        if 'ID' in ind.name:
            afr_id_list.append(ind.name)
            afr_id_class_list.append(ind.is_a)
    return(afr_id_list, afr_id_class_list)
```

Figure 27 Function Get_ID_Information definition. (the author, 2023)

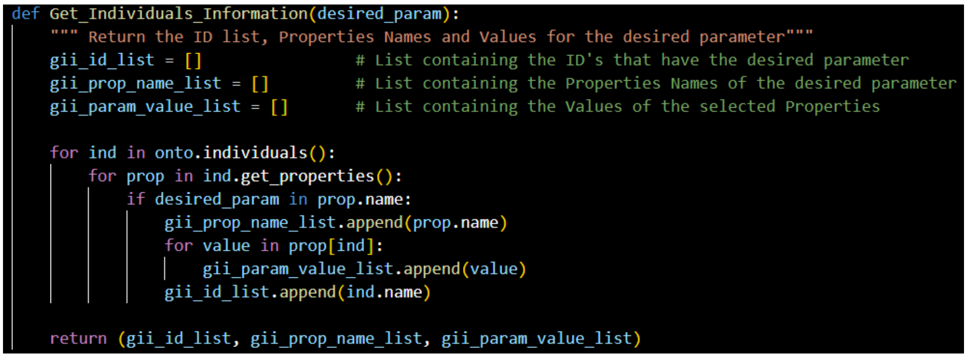
As the name suggests, the function *Get_Machines_Into_Individuals* literally extracts information about whether there is any Individual that has the word "Machine" in its name, indicating to the program that it is a machine that, besides being in the Machines class, will have specific characteristics. This function has no parameters and when it is invoked in the programming it returns a list with only the name of the machine found and not the complete address of location. As can be seen in the code that has the definition of the function in Figure 28, inside the search loop of the ontology individuals only the element Name is effectively inserted into the list.



```python
def Get_Machines_Into_Individuals():
    """Checks the machines on individuals"""
    m_list = []
    for ind in onto.individuals():
        if 'Machine' in ind.name:
            m_list.append(ind.name)
    return(m_list)
```

Figure 28 Get_Machines_Into_Individuals function description (the author, 2023)

The function *Get_Individuals_Information* receives as a parameter a specification of which is the desired individual so that the function can return the main information of this element, allowing several implementations and new designs inside the algorithm. The desired individual must be specified as a string containing its name exactly as the ontology's name, case-sensitive. To better define what is done, the function searches for individuals in the ontology added to the program and gets their properties through the *get_properties()* method, listing all properties that the individual is related to and, furthermore, gets the values of this information through the "Value" method. Thus, the function returns three lists: the first called *gii_id_list* comprises the ID's that the desired individual has (if any), the second called *gii_prop_name_list* has the names of the properties, and the third called *gii_param_value_list* contains the values of these properties.

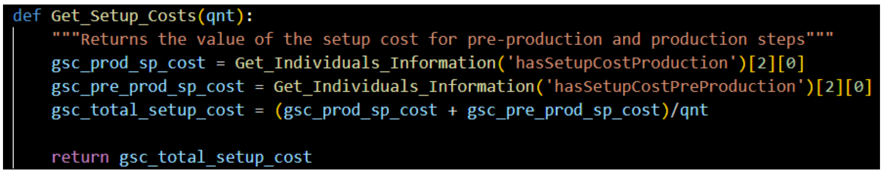


```python
def Get_Individuals_Information(desired_param):
    """ Return the ID list, Properties Names and Values for the desired parameter"""
    gii_id_list = []                # List containing the ID's that have the desired parameter
    gii_prop_name_list = []         # List containing the Properties Names of the desired parameter
    gii_param_value_list = []       # List containing the Values of the selected Properties

    for ind in onto.individuals():
        for prop in ind.get_properties():
            if desired_param in prop.name:
                gii_prop_name_list.append(prop.name)
                for value in prop[ind]:
                    gii_param_value_list.append(value)
                gii_id_list.append(ind.name)

    return (gii_id_list, gii_prop_name_list, gii_param_value_list)
```

Figure 29 Code snippet of the Get_Individuals_Information function (the author, 2023)

The previous functions were implemented to make it easier and clearer how the ontology data should be obtained and also how each type, property, class, individual or any other denomination can be accessed and available to other code snippets. The next functions are related to extracting the costs of each defined cost parameter and, therefore, do not have a degree of development complexity like the functions defined and explained above.

Following the definition of the implemented functions, the function *Get_Setup_Costs* takes advantage of the design of another function already explained to obtain the setup cost information (*Get_Individuals_Information*). This cost involves the Pre-production and Production classes and obtaining the effective value of each of these costs, the function adds them up and divides by the only necessary parameter for this function: the quantity of parts to be manufactured. This division was implemented so that, in the final cost of a desired product unit, the setup value is divided for the quantity so that the part quotation is respected, as described better in section 3.2. Figure 30 represents this implementation in the Code.



```
def Get_Setup_Costs(qnt):
    """Returns the value of the setup cost for pre-production and production steps"""
    gsc_prod_sp_cost = Get_Individuals_Information('hasSetupCostProduction')[2][0]
    gsc_pre_prod_sp_cost = Get_Individuals_Information('hasSetupCostPreProduction')[2][0]
    gsc_total_setup_cost = (gsc_prod_sp_cost + gsc_pre_prod_sp_cost)/qnt

    return gsc_total_setup_cost
```

Figure 30 Detail of the function Get_Setup_Costs in the code (the author, 2023)

The function *Get_Manufacturing_Processes_Cost* is intended to automatically return the costs of the implemented manufacturing processes, based on the parameters entered in the function and necessary for this calculation. This function has as essential parameters the general Cost per Hour, the List of Machines, the Quantity of parts to be manufactured, the total Perimeter of the part and the Widths of the folds. As with other elements of the research, the manufacturing processes entered into the function were only those applicable to the test model of this design, using the Laser Cutting Machine for cutting the part and the holes, and the Bending Machine for all the necessary bending.

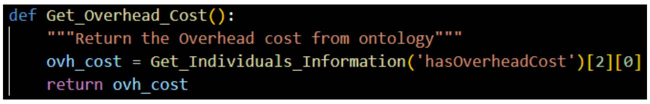
Within the function, the analysis process begins by checking that each machine is contained in the machine list and also has a cost per hour setting. After these checks, each of the types entered in the program has its specific calculation, and if any of the checks are not satisfied, the program issues an error to the user asking for verification of the definitions of individuals. Besides this, the function returns two distinct lists, one for each machine and containing the machine's name and the final value of the process pricing.

All this detailed description is shown in the figure below:

```
def Get_Manufacturing_Processes_Cost(cost_per_hr, mac_list, qnt, perim, wid):
    """Returns the cost values of each implemented process"""
    costperhour_Laser = []
    costperhour_Bending = []
    for i in range(len(cost_per_hr[0])):
        if ('BendingMachine' in cost_per_hr[0]) and ('BendingMachine' in mac_list):
                costperhour_Bending = cost_per_hr[2][cost_per_hr[0].index('BendingMachine')]
                FinalBendCost = 0
                toolradius_Bending = Get_Individuals_Information('hasBendingToolRadius')[2][0]
                costperbend = Get_Individuals_Information('hasCostPerBend')[2][0]
                bendsize_list = []

                for i in range(len(wid)):
                    bendsize_list.append(wid[i])
                    bendcost = bendsize_list[i]*costperbend
                    FinalBendCost+=bendcost
        if ('LaserMachine' in cost_per_hr[0]) and ('LaserMachine' in mac_list):
                costperhour_Laser = cost_per_hr[2][cost_per_hr[0].index('LaserMachine')]
                cuttingspeed_Laser = (Get_Individuals_Information('hasCuttingSpeed')[2][0])*1000  # [m/min] --> [mm / min]
                FinalLaserCuttingCost = round(((perim/cuttingspeed_Laser)*qnt)*(round((costperhour_Laser/60),2)),2)
        else:
            print("\n Verify Individuals definitions")
            exit()
    return(['LaserMachine', FinalLaserCuttingCost], ['BendingMachine', FinalBendCost, bendsize_list])
```

Figure 31 Get_Manufacturing_Cost function description (the author, 2023)

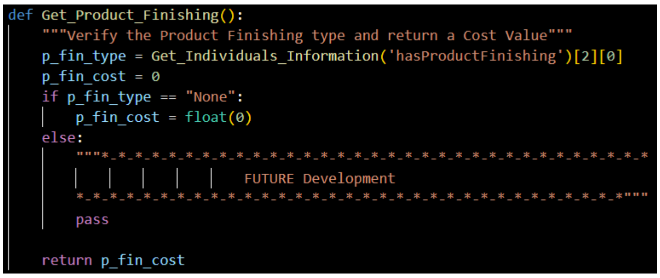
To access the overhead cost information, the function *Get_Overhead_Costs* was implemented by applying the definitions of the other function for obtaining information about the individuals already explained above, accessing one of the data properties that contains the explicit value of this cost.



```python
def Get_Overhead_Cost():
    """Return the Overhead cost from ontology"""
    ovh_cost = Get_Individuals_Information('hasOverheadCost')[2][0]
    return ovh_cost
```

Figure 32 Get_Overhead_Costs code snippet (the author, 2023)

The next and last function defined concerns the types of product finishing since, in order to establish a cost, the type of finishing must be selected and, consequently, within the selection made one must choose which treatment option will be used on the part. Each of these options will have a different way of calculating the cost and, as these definitions are beyond the general scope of this work, this whole process was not performed. However, for the function to be implemented and play a role in the pricing, one of the finishing types that can be chosen is to receive the part with No Finishing, i.e. the "None" option that results in a cost equal to zero. Accordingly, if the selected option is "None", the function returns a final cost of zero. Below is shown the function created together with the field stipulation for further development of this function.



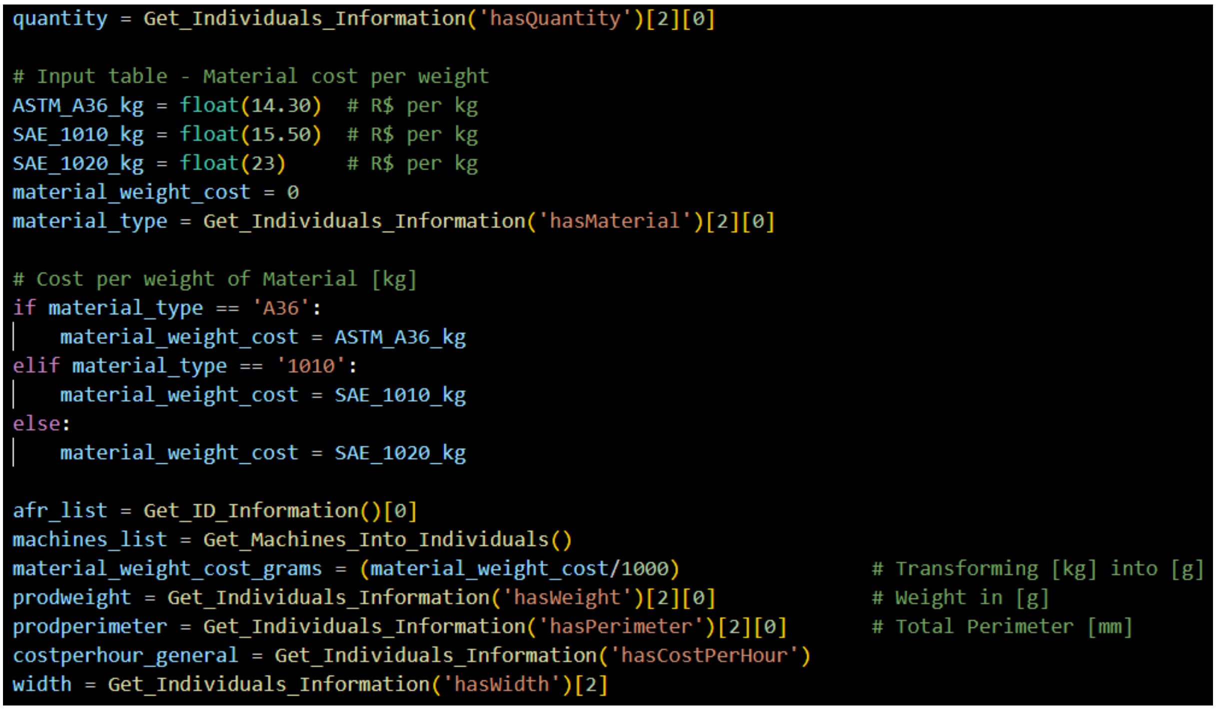
```python
def Get_Product_Finishing():
    """Verify the Product Finishing type and return a Cost Value"""
    p_fin_type = Get_Individuals_Information('hasProductFinishing')[2][0]
    p_fin_cost = 0
    if p_fin_type == "None":
        p_fin_cost = float(0)
    else:
        """*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
                            FUTURE Development
        *_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*"""
        pass

    return p_fin_cost
```

Figure 33 Get_Product_Finishing definition (the author, 2023)

3.4.2. Declaration of variables and parameters and functions executions

After defining the functions, the program continues with the declaration of variables required by the program logic and also others that were defined as demonstration parameters of how the application would be in an implementation that required such parameters. The first element inserted in this parameterization was the Quantity of parts to be manufactured, applying the function *Get_Individuals_Information* and saving it in a variable for future applications. Regarding the materials involved, the table created with the steel price estimate was inserted in the program along with a condition that receives the type of material chosen through the ontology definition and, given this selection, inserts the value of the material price by weight in the *material_weight_cost* variable. Besides all these specifications already explained, applying the functionalities of the function to get information from individuals, the variables to obtain the Perimeter value, the Product Weight, the Width list for the bending process and also the general Cost per Hour were defined, so that the last variable is applied to price the hourly cost of each machine. Furthermore, the function *Get_Machines_Into_Individuals* was applied to return to a created variable that contains the list of machines and, finally, the function *Get_ID_Information* inserts data into another list that contains the set of defined part ID's.

To clarify, all the elements mentioned were declared in the program but are not necessarily being used in any application but are available to be accessed in case of future use. Hence, in sequence there is the demonstration of all these insertions in the program.



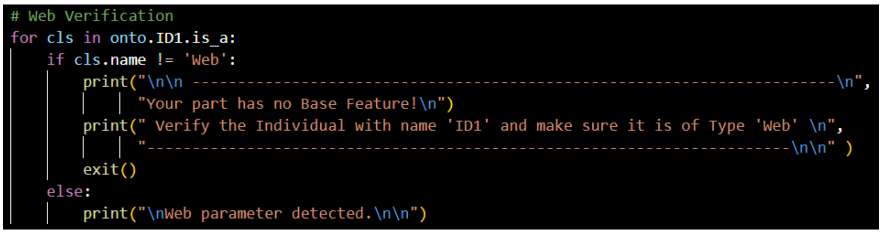
```
quantity = Get_Individuals_Information('hasQuantity')[2][0]

# Input table - Material cost per weight
ASTM_A36_kg = float(14.30)  # R$ per kg
SAE_1010_kg = float(15.50)  # R$ per kg
SAE_1020_kg = float(23)     # R$ per kg
material_weight_cost = 0
material_type = Get_Individuals_Information('hasMaterial')[2][0]

# Cost per weight of Material [kg]
if material_type == 'A36':
    material_weight_cost = ASTM_A36_kg
elif material_type == '1010':
    material_weight_cost = SAE_1010_kg
else:
    material_weight_cost = SAE_1020_kg

afr_list = Get_ID_Information()[0]
machines_list = Get_Machines_Into_Individuals()
material_weight_cost_grams = (material_weight_cost/1000)              # Transforming [kg] into [g]
prodweight = Get_Individuals_Information('hasWeight')[2][0]           # Weight in [g]
prodperimeter = Get_Individuals_Information('hasPerimeter')[2][0]     # Total Perimeter [mm]
costperhour_general = Get_Individuals_Information('hasCostPerHour')
width = Get_Individuals_Information('hasWidth')[2]
```

Figure 34 Declaration of parameters and useful variables (the author, 2023)

4. Test-case

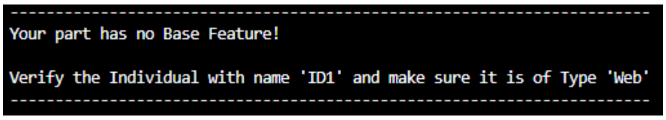
In order to prove the implemented solution, this section describes some tests performed on the algorithm to validate the results that come from the calculations made.

This way, as the scope of the research are the sheet metal parts, by the definition of the taxonomy applied in the definition of the class `Product_Features` (contained in Figure 8) the part must contain a base characteristic named `Web`, where all the other features are inserted. If this is not identified, the pricing calculation cannot be performed, and the program should return to the user an error indicating what the problem is and how to solve it. This is satisfied by the code snippet shown below and the next figure represents the output of the program in case of a failure to comply with the rule defined.



```
# Web Verification
for cls in onto.ID1.is_a:
    if cls.name != 'Web':
        print("\n\n -----------------------------------------------------------------------\n",
              "Your part has no Base Feature!\n")
        print(" Verify the Individual with name 'ID1' and make sure it is of Type 'Web' \n",
              "-----------------------------------------------------------------------\n\n" )
        exit()
    else:
        print("\nWeb parameter detected.\n\n")
```

Figure 35 Code snippet that performs the verification (the author, 2023)



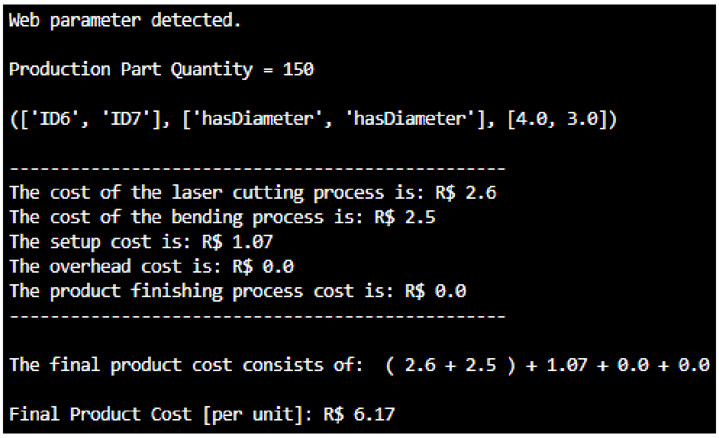
```
-----------------------------------------------------------------------
Your part has no Base Feature!

Verify the Individual with name 'ID1' and make sure it is of Type 'Web'
-----------------------------------------------------------------------
```

Figure 36 Output message with condition not fulfilled (the author, 2023)

What is done by the code snippet shown above is a check of all the individuals and their data properties, aiming to locate the characteristic with the name `Web`. If this statement is not satisfied, the program shows the user the reason for the error and also requests the insertion of this data in the ontology.

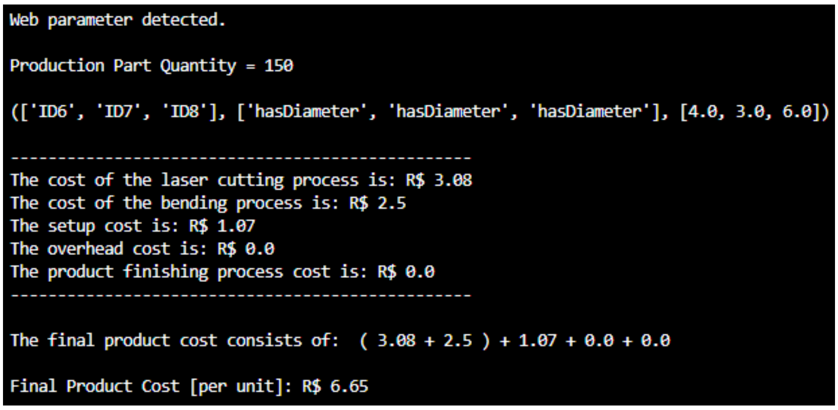
As the proposal of this research has as its main objective to obtain the design of an automated pricing system based on the modeling and structuring of an ontology, it is intended to attest that with different parameters the result of the final calculation should remain cohesive. To this end, a new hole was added to the part with a diameter of 6mm, a value different from the others already contained in the original part. To accomplish this change, the individual `ID8` was created in the ontology having the class `Attachment_Hole` and with the diameter of 6mm and, consequently, the total perimeter of the part was also changed in the ontology. Therefore, with these changes made in the ontology, the program was run twice, the first time without the new hole and the second time with the addition of the hole.

Both output results of the program are shown below by Figures 37 and 38:



```
Web parameter detected.

Production Part Quantity = 150

(['ID6', 'ID7'], ['hasDiameter', 'hasDiameter'], [4.0, 3.0])

------------------------------------------------
The cost of the laser cutting process is: R$ 2.6
The cost of the bending process is: R$ 2.5
The setup cost is: R$ 1.07
The overhead cost is: R$ 0.0
The product finishing process cost is: R$ 0.0
------------------------------------------------

The final product cost consists of:  ( 2.6 + 2.5 ) + 1.07 + 0.0 + 0.0

Final Product Cost [per unit]: R$ 6.17
```

Figure 37 Program output without the Individual “ID8” (the author, 2023)



```
Web parameter detected.

Production Part Quantity = 150

(['ID6', 'ID7', 'ID8'], ['hasDiameter', 'hasDiameter', 'hasDiameter'], [4.0, 3.0, 6.0])

------------------------------------------------
The cost of the laser cutting process is: R$ 3.08
The cost of the bending process is: R$ 2.5
The setup cost is: R$ 1.07
The overhead cost is: R$ 0.0
The product finishing process cost is: R$ 0.0
------------------------------------------------

The final product cost consists of:  ( 3.08 + 2.5 ) + 1.07 + 0.0 + 0.0

Final Product Cost [per unit]: R$ 6.65
```

Figure 38 Program output with the new Individual “ID8” (the author, 2023)

As observed in the results obtained, the program correctly obtained all the parameters, even when adding a new one. To prove this, the command to show the return information of the function `Get_Individuals_Information` with the parameter `hasDiameter` was inserted into de code, resulting in the return of three lists, according to the structure of the function, containing the ID's, names, and values. It can be noted that the 6mm diameter was added to the list of values with the correct addressing of the `ID8` individual created and furthermore, the cost of the laser cutting process increased from R$2,60 to R$3,08 due to the fact that, with the addition of the hole, the total perimeter of the part increases resulting in a price increase. The demonstration also shows that all other parameters were unchanged and kept the same value.

5. Results and discussion

As the main approach of the research is the idealization and construction of an automatic pricing process for sheet metal parts, the main results are related to the structuring of all cost concepts involved in the construction of the system, together with other areas such as the large area of manufacturing and also of materials. Due to the amplitude of the research and the themes involved, it was necessary to select which parts should have a greater focus in the development and which would have to be left aside for this study, however, even with this division, the formalization of the knowledge and concepts should be implemented. Thus, one of the results for the composition of the automated pricing system was the creation of the ontology, with the definition of all classes and properties, represented by Figure 38.

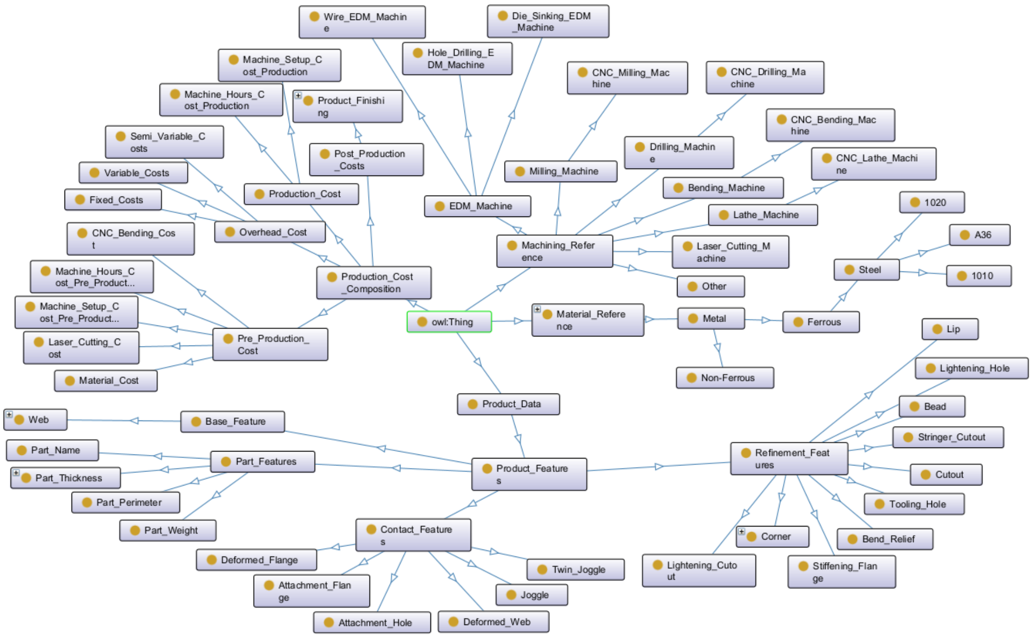
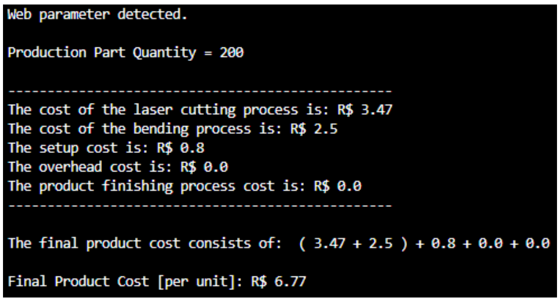


Figure 39 General ontology developed (the author, 2023).

With the ontology being the basis for structuring the data, the implemented algorithm takes advantage of all the concepts and relationships made so that the separation of information within the program is done correctly. As already described and explained in detail, the algorithm receives all this information and applies the functions created in the program to access the necessary data and give as a final answer the cost of manufacturing the part, showing the user how this process is done.



```
Web parameter detected.

Production Part Quantity = 200

-------------------------------------------------
The cost of the laser cutting process is: R$ 3.47
The cost of the bending process is: R$ 2.5
The setup cost is: R$ 0.8
The overhead cost is: R$ 0.0
The product finishing process cost is: R$ 0.0
-------------------------------------------------

The final product cost consists of:  ( 3.47 + 2.5 ) + 0.8 + 0.0 + 0.0

Final Product Cost [per unit]: R$ 6.77
```

Figure 40 Example of program output response (the author, 2023).

As it is possible to observe in Figure 40, the final price (R$) presented is always for one piece, so that the number of pieces to be manufactured is only relevant in the calculation itself. For the quotation of the manufacturing processes, the implemented way made all the calculation automated as long as the individuals inserted into the ontology have the correct structure. This data insertion in the individuals is essential for the created logic to work correctly, even changing some parameters and inserting or removing individuals.

The approach to the discussion of this paper begins by questioning whether the insertion of all separate individuals as was done is a better solution or if there is the possibility of inserting only one individual with the general characteristics of the product to be manufactured. This has a direct impact on the optimization of this system, making the system conceived as a whole even more autonomous and uncomplicated for use in different parts. Furthermore, another pertinent issue to be discussed is the insertion of data about material prices, for example, which can be optimized to import all these values from a table in .csv or .xlsx formats, since this is how this information is often arranged. The functions implemented by the student can and certainly will receive upgrades and improvements, but they allowed the student to understand how the integration of an ontology with Python programming works and to access the parameters and main definitions, and this is relevant for further research.

6. Conclusion

The development process began with the search for more information about the most relevant themes in the literature in order to propose a conceptual architecture for the implementation of the automated system, integrating the ontological approach with programming concepts, applying the solution elaborated in parts made of sheet metal aiming the pricing of the part or, at least, a cohesive estimate of this quotation. The extraction of the information from the 3D model was made possible by the AFR tool, which made it possible to insert this information into the ontology and start to truly conceive the proposed solution.

An automated system that contains the association of the ontology structuring, with the defined rules and knowledge, with the Python algorithm containing the necessary information, the definition of the business rules and also obtaining the characteristics of the part to be manufactured can be extremely profitable and efficient for several industry sectors, because the implemented set has a high ease of insertion of new parameters and definitions. In the industrial scope, since the research is part of the aeronautical industries, the reduction of the parts pricing time associated with the reduction of quotation errors are fundamental advantages to win over the customer and, moreover, the provision of a precise cost estimate quickly speeds up the entire manufacturing process.

Therefore, to conclude, the problem question involving the possibility of automation of the pricing process was answered with the successful implementation of the whole set (information structuring + insertion of business rules to obtain the cost value). Aiming at the future research, since the research study axis is quite broad, the cost ontology can be better described to address several other concepts and also the whole set of pricing manufacturing processes can be filled with more information and optimized. However, as the intent of the project was to design the automated pricing system that had an application on a real part, the end result exceeded the student's expectations, since the automations were successful in describing and receiving all the information in the face of the various tests performed, meaning that the efficiency and reliability of these processes can be improved along with minimizing human error in decision making.

7. References

HOZDIC, Elvis (2015). Smart factory for industry 4.0: A review. Journal of Modern Manufacturing Systems and Technology. 7. 28-35.

ZHENG, P., wang, H., Sang, Z. et al. Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. Front. Mech. Eng. 13, 137-150 (2018). https://doi.org/10.1007/s11465-018-0499-5

ETTAIBI Mohamed, Mokhtari Bouchaib. Integrative Digital Transformation and Organizational Competitiveness (OC): Proposal for a Transition Strategy to the Aeronautics Industry 4.0 (TSAI) based on a Knowledge Management (KM) approach in a Moroccan ecosystem environment. E3S Web Conf. 319 01021 (2021) DOI: 10.1051/e3sconf/202131901021

T. ALTAN, S. Oh and H. Gegel, "Metal Forming Fundamentals and Applications," American Society for Metals, Ohio, 1983.

F. MAS, J.L. Menéndez, M. Oliva, J. Ríos. Collaborative Engineering: An Airbus Case Study. Procedia Engineering. Volume 63. 2013. Pages 336-345. ISSN 1877-7058. https://doi.org/10.1016/j.proeng.2013.08.180.

LAMY JEAN-BAPTISTE. Ontologies with Python: programming OWL 2. 0 ontologies with Python and Owlready2. Berkeley, Ca: Apress, 2021.

SZEJKA, A.L., Mas, F., Junior, O.C. (2022). Towards Knowledge-Based System to Support Smart Manufacturing Processes in Aerospace Industry Based on Models for Manufacturing (MfM). In: Canciglieri Junior, O., Noël, F., Rivest, L., Bouras, A. (eds) Product Lifecycle Management. Green and Blue Technologies to Support Smart and Sustainable Organizations. PLM 2021. IFIP Advances in Information and Communication Technology, vol 640. Springer, Cham. https://doi.org/10.1007/978-3-030-94399-8_31

GEIGER, M., Knoblach, J., & Backes, F. (1999). Cost Estimation for Large Scale Production of Sheet Metal Parts Using Artificial Neural Networks.

KALPAKJIAN, S. and Schmid. S. (2014) Manufacturing Engineering & Technology. 7th Edition, Pearson Publishing Company, Upper Saddle River, New Jersey.

REBSTOCK, Michael & Fengel, Janina & Paulheim, Heiko (2008). Ontologies-Based Business Integration. 10.1007/978-3-540-75230-1.

HÉON, Michel & Paquette, Gilbert. (2020). G-OWL: A Complete Visual Syntax for OWL 2 Ontology Modeling and Communication. Semantic Web.

THOBEN, Klaus-Dieter & Wiesner, Stefan & Wuest, Thorsten (2017). "Industrie 4.0" and Smart Manufacturing - A Review of Research Issues and Application Examples. International Journal of Automation Technology. 11. 4-19. 10.20965/ijat.2017.p0004.

KUSIAK, Andrew (2018) Smart manufacturing, International Journal of Production Research, 56:1-2, 508-517, DOI: 10.1080/00207543.2017.1351644

F. PIRES, A. Cachada, J. Barbosa, A. P. Moreira and P. Leitão, "Digital Twin in Industry 4.0: Technologies, Applications and Challenges". 2019. IEEE. 17th International Conference on Industrial Informatics (INDIN), 2019, pp. 721-726, doi: 10.1109/INDIN41052.2019.8972134.

MAROTO, M. The process of aluminum sheet forming in rubber press for the aeronautical industry. XII SIMPEP, Nov. 7DC.

LEMINEN, Ville & Mäkelä, Petri & Tanninen, Panu & Varis, Juha. (2015). The use of chromatic white light 3D-profilometry in the surface quality analysis of press-formed paperboard trays.

MARCONDES, P. Sheet metal fabrication - bending. Ferramental Magazine, Feb. 2016.

MARCINIAK, Z.; DUNCAN, J. L.; S J. HU. Mechanics of sheet metal forming. Oxford: Butterworth-Heinemann, 2002.

S.H. ZHANG, Z.R. Wang, Y. Xu, Z.T. Wang, L.X. Zhou. Recent developments in sheet hydroforming technology. Journal of Materials Processing Technology, Volume 151, Issues 1-3, 2004, Pages 237-241, ISSN 0924-0136, https://doi.org/10.1016/j.jmatprotec.2004.04.054.

P. HEIN, F. Vollertsen. Hydroforming of sheet metal pairs. Journal of Materials Processing Technology. Volume 87, Issues 1-3. 1999. Pages 154-164. ISSN 0924-0136. https://doi.org/10.1016/S0924-0136(98)00347-1.

JEPSEN, Thomas C. "Just What Is an Ontology, anyway?". IT Professional Magazine. Washington Vol. 11, Ed. 5, (2009): 22-27. DOI:10.1109/MITP.2009.105

LEHMANN, J. & Völker, J. (2014). An introduction to ontology learning. Perspectives on Ontology Learning. 18. ix-xvi.

HAASE, Peter & Völker, Johanna (2008). Ontology Learning and Reasoning - Dealing with Uncertainty and Inconsistency. CEUR Workshop Proceedings. 173. 366-384. 10.1007/978-3-540-89765-1_21.

S. LEMAIGNAN, A. Siadat, J. . -Y. Dantan and A. Semenenko, "MASON: A Proposal for An Ontology Of Manufacturing Domain," IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), Prague, Czech Republic, 2006, pp. 195-200, doi: 10.1109/DIS.2006.48.

HEIYANTHUDUWAGE, S.R., Schwitter, R. & Orgun, M.A. OWL 2 learn profile: an ontology sublanguage for the learning domain. SpringerPlus 5, 291 (2016). https://doi.org/10.1186/s40064-016-1826-0

VERLINDEN, B. & Duflou, Joost & Collin, P. & Cattrysse, Dirk. (2008). Cost estimation for sheet metal parts using multiple regression and artificial neural networks: A case study. International Journal of Production Economics. 111. 484-492. 10.1016/j.ijpe.2007.02.004.

GEIGER, M., & Knoblach, J. (1999). Cost Estimation of Sheet Metal Parts with Neural Networks.

F. TAO and Q. Qi, "New IT Driven Service-Oriented Smart Manufacturing: Framework and Characteristics ". in *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 1, pp. 81-91, Jan. 2019, doi: 10.1109/TSMC.2017.2723764.

GHAFFARISHAHRI, Seyedmorteza (2023). Semantic model difference identification for aerospace sheet metal part CAD models. Thèse de doctorat électronique, Montréal, École de technologie supérieure.

CHISHOLM, M. (2003). How to Build a Business Rules Engine: Extending Application Functionality through Metadata Engineering (The Morgan Kaufmann Series in DataManagement Systems). San Fransisco, CA: MorganKaufman, 12 November 2003.

ASHINO, T. (2010), Materials Ontology: An Infrastructure for Exchanging Materials Information and Knowledge. Data Science Journal, vol. 9, pp. 54-61, 2010.

SZEJKA, Anderson. (2016). Contribution to interoperable products design and manufacturing information: application to plastic injection products manufacturing. Computer Aided Engineering. Université de Lorraine.