INVESTIGACIÓN e INNOVACIÓN en INGENIERÍAS
ISSN2344-8652

An ontology-based application of the ISO/IEC 29110 standard basic profile by using Protégé

Una aplicación basada en ontologías del perfil básico del estándar ISO/IEC 29110 empleando Protégé

Antonio Vera Delgado **Carlos Mario Zapata Jaramillo**

Gloria Lucía Giraldo Gómez
Universidad Nacional de Colombia

Recibido: 21/10/2020
Aceptado: 18/01/2021
Publicado: 19/03/2021

Correspondencia de autores:
avera@unal.edu.co

Resumen

Objetive: provide a scalable environment to represent the concepts of the basic profile of the ISO / IEC 29110 standard for Small and Medium-sized Enterprises (SMEs) and their relationships. **Methodology:** In this paper we propose a novel approach for generating an ontology related to the ISO/IEC 29110 basic profile. We follow some steps: (i) modeling the domain by using executable pre-conceptual schemas; (ii) creating equivalences between pre-conceptual schemas and ontology elements; (iii) translating such equivalences into Protégé, an ontology-based environment; and (iv) creating rules for inferring knowledge from the ontology in order to overcome the aforementioned problems. **Results:** We create an ontology related to concepts of the ISO/IEC 29110 and their relations in its basic profile. We also answer questions in order to ease the implementation process of the ISO/IEC 29110 basic profile. **Conclusions:** The resulting ontology serves as support for VSEs and academics when implementing or teaching the basic profile of the ISO/IEC 29110.

Palabras clave: ISO/IEC 29110 Standard, Knowledge Representation, Ontologies, Pre-conceptual Schemas, Protégé.

Abstract

Objective: Proveer un entorno escalable para representar los conceptos del perfil básico del estándar ISO/IEC 29110 para Pequeñas y Medianas Empresas (PyME) y sus relaciones. **Metodología:** En este artículo se propone un novedoso enfoque para generar una ontología relacionada con el perfil básico de la ISO/IEC 29110. Los pasos a seguir son: (i) modelar el dominio empleando esquemas preconceptuales; (ii) crear equivalencias entre los esquemas preconceptuales y los elementos de la ontología; (iii) traducir esas equivalencias en Protégé, un entorno para la construcción de ontologías; (iv) crear reglas para inferir conocimiento desde la ontología para superar los problemas mencionados. **Resultados:** Se crea una ontología relacionada con los conceptos del estándar ISO/IEC 29110 y sus relaciones en su perfil básico. También, se responden preguntas para facilitar el proceso de implementación del perfil básico de la ISO/IEC 29110. **Conclusiones:** La ontología resultante sirve como apoyo para que PyMEs y académicos puedan implementar o enseñar el perfil básico de la ISO/IEC 29110.

Keywords: Estándar ISO/IEC 29110; Representación del Conocimiento; Ontologías; Esquemas Preconceptuales; Protégé.

Como citar (IEEE): A. Vera-Delgado., C. Zapata-Jaramillo., y G. Giraldo-Gómez. "An ontology-based application of the ISO/IEC 29110 standard basic profile by using Protégé", *Investigación e Innovación en Ingenierías* vol. 9, n°1, 86-99, 2021. DOI: https://doi.org/10.17081/invinno.9.1.4227

Introduction

Organizations promote software process standards for addressing demands for quality in basically three aspects: product, process, and organization. Laporte et al. [1] say a document "established by consensus and approved by a recognized body that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context" is called a standard. ISO/IEC 29110 standard is intended to be implemented by VSEs in some environments. The generic profile group of this standard includes situational factors to be applied to a majority of VSEs [2]. The core of each profile contains processes, activities, tasks, roles, etc., depending on the project complexity. This basic profile—belonging to the generic profile group—is related to quality improvement in software projects and is suitable to any development approach—agile, waterfall, incremental, etc.—based on the VSE needs [2].

Case studies related to the ISO/IEC 29110 standard are usually focused on the basic profile [3, 4, 5, 6, 6,8, 9, 10], since most projects are small enough for using the lightweight process model belonging to the basic profile—i.e., processes useful for Project Management (PM) and Software Implementation (SI). Also, the basic profile can be applied by VSEs to a unique project team with one single application at a time, regardless the specific risks or the situational factors [2].

The benefits of applying the standard are independent of a software development method or a specific lifecycle. Therefore, VSEs face a compliance problem about the steps to be followed in order to implement some standard [10]. Tasks of every activity can be completed by using the ISO/IEC 29110 in an open way. On the other hand, previous studies have been developed to demonstrate the value of ontologies as standard tools for representing knowledge and organizing software projects [11]. The development of ontology-based standards and models is aimed for analyzing, understanding processes, and guiding plans. Besides, such representations can be easily reused for serving multiple purposes. However, some previous ontologies include a set of ISO standards which ends in a high-level process represented, lacking granularity in sub-process, activities, and specific tasks [12, 13]. Likewise, some cases studies include ISO/IEC 29110 ontologies lacking domain representations closer to natural language and essential for understanding such a domain [14]. Also, some ontologies are static like taxonomies [15]. Previous studies include reasoners and the possibility to write rules and axioms for inferring knowledge automatically and answering questions. However, such approaches are constructed for different ISO standards [16]. In addition, some case studies include executable pre-conceptual schemas for representing the ISO/IEC 29110 basic profile but only in the first stages of the standard [17].

In this paper we develop two approaches in order to provide a robust knowledge representation of the ISO/IEC 29110 basic profile: a pre-conceptual schema containing the complete framework of the profile—including structural, dynamic, and conditional relationships—into a graphical form [18, 19] and an ontology developed in Protégé, a graphical ontology framework for building intelligent systems and acquiring knowledge [20]. Protégé is based on the Web Ontology Language (OWL). Therefore, the resulting ontology contains a set of semantic rules structured in the Semantic Web Rule Language (SWRL) embedded in the framework [21]. Besides, the ontology is usable for responding questions by including queries based on the Semantic Query-enhanced Web Rule Language (SQWRL) [22].

The ontology is aimed for including formal naming, definitions, and interrelationships of entities described in the ISO/IEC 29110 basic profile domain, supporting VSEs for implementing and understanding such a profile.-

This paper is structured in the following way: in Section 2 we introduce the method by including a conceptual framework, the required background and related studies, and the development of the method including the executable pre-conceptual schema and the ontology in Protégé; in Section 3 we discuss the results obtained, finally, in Section 4 we provide some conclusions possible lines of future work.

Methodology

Conceptual Framework

ISO/IEC 29110 Basic Profile

The ISO/IEC 29110 standard “Lifecycle profiles for Very Small Entities” is developed and adjusted for VSE projects [2]. Such a standard includes a management and engineering guide with a generic profile group. The profile group includes a set of profiles the scope dependent on the size of the project to be implemented:

- Entry profile. For start-up VSEs
- Basic profile. For the development of a single application
- Intermediate profile. For the development of multiple projects
- Advanced profile. For independent software development businesses [1].

The basic profile can be applied to projects in VSEs with 25 people or less. A single project is developed at a time. Such a profile can be applied to noncritical system development. The project should sign any contract—internal or external. Besides, the basic profile is independent of the development method—e.g., eXtreme Programming XP, Scrum, Rational Unified Process RUP, Custom Development Method CDM, etc. Project management and software implementation are the main processes of the basic profile. Practices belonging to this profile are based on the ISO/IEC 12207 standard. Implementation of the basic profile has the following advantages:

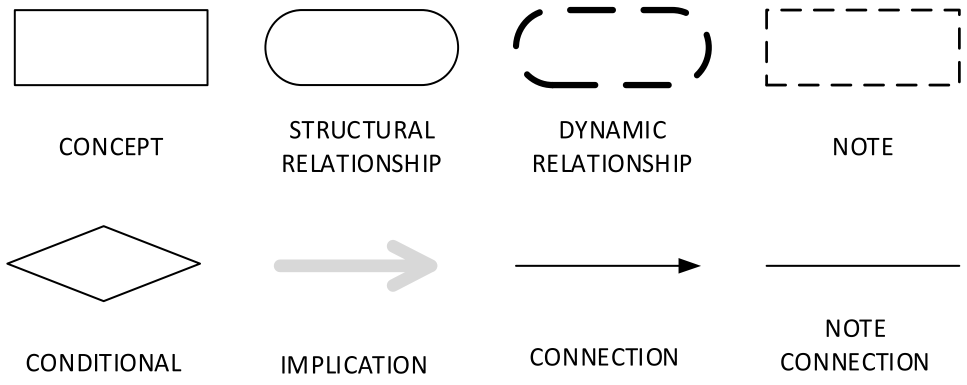
- Establishment of a set of requirements and work products for the project.
- Management processes, which provide disciplined, right, and appropriate activities. The project visibility is also increased.
- Software implementation process for satisfying the acquirer and helping to ensure a quality product.

Pre-conceptual Schemas

Zapata et al. [19] define pre-conceptual schema as "a simple representation framework for ontological knowledge with dynamic and deontic characteristics." Pre-conceptual schemas are used for translating controlled natural language into a knowledge representation. Such schemas can be used for transforming requirements explicit in a stakeholder discourse into a conceptual-graph-like representation as the previous step for developing a conceptual model [19]. Also, such schemas have many useful features, including unambiguous rules, automated translation to several diagrams, structural and dynamic triads, and similarity with natural language [19]. Besides, pre-conceptual schemas can be executable, including cognitive advantages in order to provide a higher understanding to the stakeholder in terms of system instances and operational behavior. Such a functionality allows to analyze a domain as an interactive and agile road map [23].

Main symbols of pre-conceptual schemas are shown in Figure 1. Concepts are labeled with either a noun or noun phrase. Structural relationships are referred to the verbs: be and have. Dynamic relationships are labeled with an action verb. Notes contain possible values of concepts. Conditionals have logical conditions between concepts. Implications are cause-and-effect relationships between dynamic relationships; they can be also used between conditionals and dynamic relationships. Connections have no label and serve to connect concepts with structural and dynamic relationships. Note connections are used to connect concepts with notes [19].

Figure 1. Main symbols of pre-conceptual schemas



Source: [19]

Protégé/OWL

Protégé is a graphical tool for ontology editing and knowledge acquisition released in the Stanford University School of Medicine. Such a tool can be used to represent a conceptual model by using semantic web languages. Protégé is suitable for representing a domain on a conceptual level [24]. Besides, the OWL plugin allows developers for automatically generating code [20]. Protégé provides intelligent and intuitive guidance for creating instances—classes, object properties, data properties, individuals, etc.—and helping to find mistakes [20]. Semantic restrictions help users to easily prototype an environment and evaluate the inferences deduced by the reasoner in each individual. An ontology created by using such a tool can be packaged as a Protégé plugin and can be used in different satellite applications.

Inference rules can be created by using the full-featured interactive open-source rule editor for SWRL included in Protégé, storing SWRL rules in an OWL knowledge base. O'Connor et al. [21] note SWRL is based on "a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language the Unary/Binary Datalog sublanguages of the Rule Markup Language." The Protégé SWRL editor is an extension to Protégé-OWL allowing for creating, editing, reading, and writing SWRL rules. The editor helps developers to directly refer to OWL classes, object properties, data properties, and individuals previously established in the OWL knowledge base [21]. A graphical interface has been developed in the SWRLTab plugin editor in order to have the ability to extract information from OWL ontologies. Such an interface includes the use of SQWRL, a query language based on SWRL semantic providing an array of operators—select, negation, counting, aggregation, etc.—allowing users for creating queries in OWL ontologies [22]. Users of Protégé can automatically answer questions about the domain and obtain a dynamic and natural interaction with the ontology.

Background and related studies

Ontologies have been used for representing software engineering standards and models in order to provide enhanced knowledge of both domain comprehension and task understanding. Ontology-based standards are aimed to support the tailoring of complex process models in the software industry [14]. Ontologies are useful in domain-oriented software development environments supporting software developers in their activities along the software development process [11].

Henderson et al. [12] and Gonzales et al. [13], propose an approach for developing an ontology for harmonizing ISO software engineering standards. Henderson et al. [12] relate a proposal sent to the sub-committee responsible for software engineering standards (SC7), in which they propose five ontologies. Together, such ontologies would homogenize the definitions and process types of different ISO standards. The proposed ontologies are:

- Definitional Elements Ontology (DEO), a taxonomy with a set of attributes and high-level relationships.
- Configured Definitional Ontology (CDO), either a standard or an intermediate template for creating a standard domain ontology for refining and configuring a DEO.
- Standard Domain Ontology (SDO), with some elements of the CDO to specifically represent a particular situation.
- Standards Relationship Diagram (SRD), an ontology of ontologies for defining relationships between standards and the linkage of current and future standards.
- Advanced Foundational Ontology for Standards (AFOS), a lower-level representation than DEO with some of its sub-classes.

Henderson et al. [13] show fractions of diagrams for creating the ontology. A sample of the harmonization of concepts and a high-level DEO diagram are used for obtaining two CDOs. Finally, a list of concepts of a DEO based on the ISO/IEC 29110 standard is developed as an example. However, the creation of such an ontology lacks granularity in sub-processes, activities, and specific tasks. The authors only define the highest level objects for the harmonization, ignoring detailed representation of each standard.

Eito [14] develops a solution focused on companies with complex engineering models. The basis of such a solution is an ontology in OWL with the methodology proposed in the ISO/IEC 29110 standard. Eito [14] intended to adapt the ISO/IEC 29110 guidelines to current practices of a company by creating a structured project development method, including a semantic wiki and a scalable way for including sub-classes. However, the resulting ontology lacks a domain representation for implementing the standard.

Faiano et al. [15] attempt to harmonize the NTP-ISO/IEC 12207, ISO/IEC 29110, and ISO/IEC/IEEE 29119 standards by using structural models. The authors identify common points and create two knowledge representations based on ontologies by using the Software Engineering Ontology Network (SEON) and the Reference Ontology on Software Testing (ROoST). SEON is a graphical software for creating ontologies, and ROoST is an ontology created to establish a common domain of the software testing conceptualization. The authors map the three standards with an emphasis on conceptual harmonization among objects defined in each standard. However, detailed ontologies are static, lacking a reasoner for semantic rule deduction, which makes such ontologies unable to automatically answer domain questions.

Batres et al. [16] detail the development of a high-level ontology in OWL based on the ISO 15926 standard related to the integration of data between computer systems. The authors defined mereology and topology relationships, physical objects, and activities, allowing scalability for the future creation of sub-classes and new relationships. Domain problem layout is done in a causality editor internally developed. The OWL is coded into Java Theorem Prover, which is an object-oriented modular reasoner. This process is intended to ontologically represent an ISO standard at a high level by using diagrams and introduce the ontology for rule deduction.

Our method: From pre-conceptual schemas to Protégé/OWL

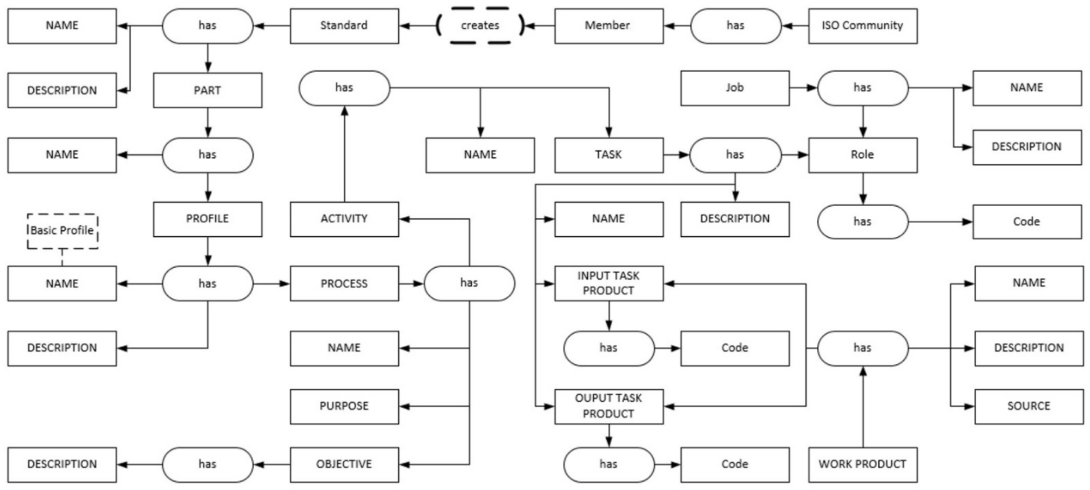
We develop an executable pre-conceptual schema for representing the standard domain in the first step. We propose the resulting executable pre-conceptual schema in Figure 2. The main concept is labeled as ISO Community. We add the lower hierarchy having as a final element the work product concept. We define the features of each work product concept according to the ISO/IEC 29110 [2]:

Input process products. Products required to perform the process and its corresponding source, which can be either another process or an external entity of the project, such as the acquirer.

Output process products. Products generated by the process and its corresponding destination, which can be either another process or an external entity of the project, such as an acquirer or organizational manager.

The second step of our method comprises the definition of equivalences between some constructs of the executable pre-conceptual schema (our domain) and a target set of elements in OWL (classes, object properties, data properties, and SWRL semantic rules). Equivalences are proposed in Table 1.

Figure 2. Executable pre-conceptual schema related to the Basic Profile of the ISO/IEC 29110



Source: Own production

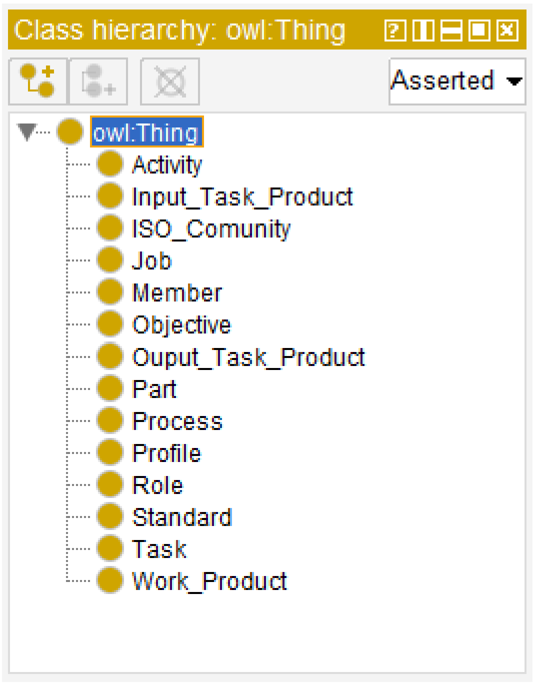
Table 1. Equivalences between pre-conceptual schema elements and OWL classes/properties

Pre-conceptual schema element	OWL class/property
Concept	Class
Structural relationship IS	Sub-class
Structural relationship HAS/Dynamic relationship	Object property
Leaf concept	Data property
Note	Data property value
Conditional/Implication	SWRL semantic rule

Source: Own production

We construct the ontology by using the Protégé framework in the step three. We start by creating in Protégé main classes of the elements described in the pre-conceptual schema as concepts excluding leaf concepts. When two concepts are linked by using a structural relationship IS, child concepts should be created as sub-classes of the parent (see Figure 3). Created classes should be defined as disjoints.

Figure 3. Classes in the Protégé framework

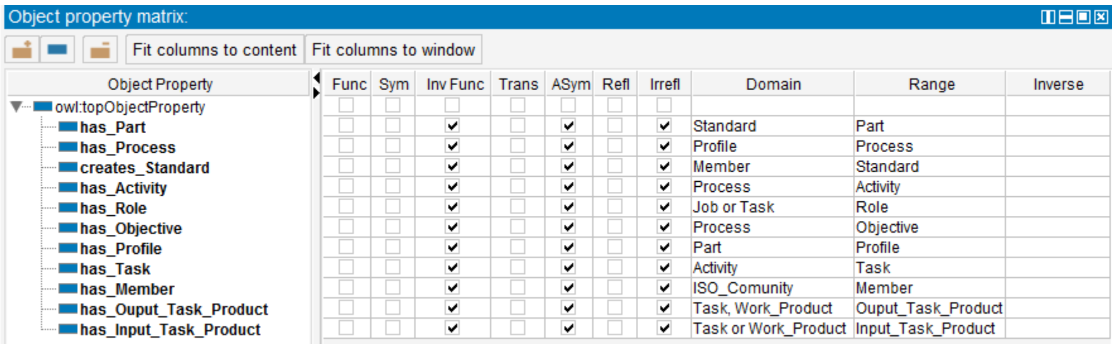


- owl:Thing
  - Activity
  - Input_Task_Product
  - ISO_Comunity
  - Job
  - Member
  - Objective
  - Ouput_Task_Product
  - Part
  - Process
  - Profile
  - Role
  - Standard
  - Task
  - Work_Product

Source: Own production

We design the object properties for creating the relationships between classes. Therefore, we follow the equivalence table searching for structural relationships HAS and dynamic relationships. We create the object properties in a hierarchical way by defining each range and domain according to the concepts and relationships previously established in the pre-conceptual schema as illustrated in Figure 4. For example, the concept STANDARD has a structural relationship HAS with the concept PART, so we create the object property has_Part for specifying the class as the domain standard and the class part as the range. Besides, we propose in Figure 5 the object property matrix, which also includes the object type restrictions—functional, symmetric, inverse functional, transitive, asymmetric, reflexive, irreflexive, and inverse—defined in order to establish restrictions between objects. We create the data properties searching for leaf concepts in the pre-conceptual schema. Similar to object properties, we define a range and a domain. A main concept linked with a leaf concept by a static relationship "has" become a domain as illustrated in Figure 5. Likewise, the range is specified with the datatype of the value used in such a data property. In the creation of data properties, we specify the leaf concepts names taken from the pre-conceptual schema prefixing the verb HAS. Such an implementation is used to design an ontology with the closest possible representation to natural language.

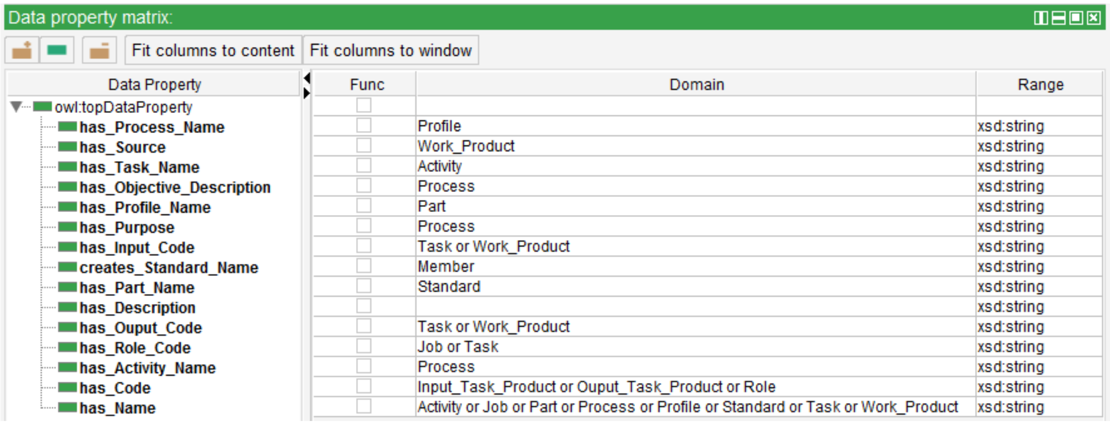
Figure 4. Object Property Matrix in Protégé framework



Object Property	Func	Sym	Inv Func	Trans	ASym	Refl	Irrefl	Domain	Range	Inverse
owl:topObjectProperty										
has_Part			✓		✓		✓	Standard	Part	
has_Process			✓		✓		✓	Profile	Process	
creates_Standard			✓		✓		✓	Member	Standard	
has_Activity			✓		✓		✓	Process	Activity	
has_Role			✓		✓		✓	Job or Task	Role	
has_Objective			✓		✓		✓	Process	Objective	
has_Profile			✓		✓		✓	Part	Profile	
has_Task			✓		✓		✓	Activity	Task	
has_Member			✓		✓		✓	ISO_Comunity	Member	
has_Ouput_Task_Product			✓		✓		✓	Task, Work_Product	Ouput_Task_Product	
has_Input_Task_Product			✓		✓		✓	Task or Work_Product	Input_Task_Product	

Source: Own production

Figure 5. Data Property Matrix in Protégé framework

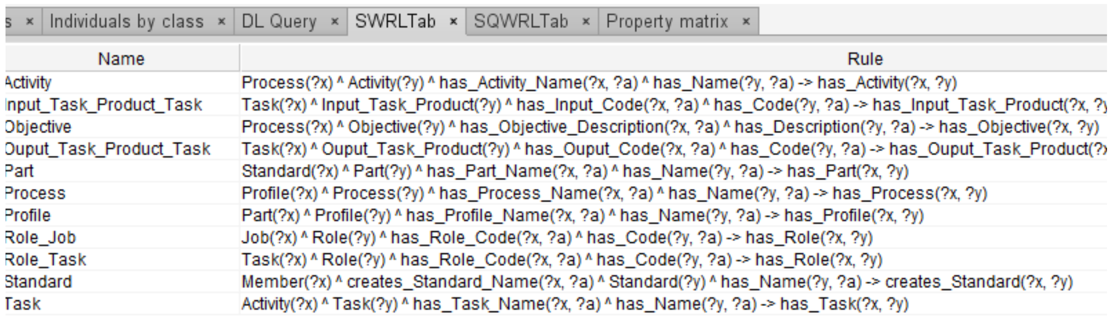


Data Property	Func	Domain	Range
owl:topDataProperty			
has_Process_Name		Profile	xsd:string
has_Source		Work_Product	xsd:string
has_Task_Name		Activity	xsd:string
has_Objective_Description		Process	xsd:string
has_Profile_Name		Part	xsd:string
has_Purpose		Process	xsd:string
has_Input_Code		Task or Work_Product	xsd:string
creates_Standard_Name		Member	xsd:string
has_Part_Name		Standard	xsd:string
has_Description			xsd:string
has_Ouput_Code		Task or Work_Product	xsd:string
has_Role_Code		Job or Task	xsd:string
has_Activity_Name		Process	xsd:string
has_Code		Input_Task_Product or Ouput_Task_Product or Role	xsd:string
has_Name		Activity or Job or Part or Process or Profile or Standard or Task or Work_Product	xsd:string

Source: Own production

The fourth step is related to the development of the semantic rules used to create restrictions in the open world. We create some rules in SWRL by using classes, data properties, variables, and constants as we show in Figure 6. For example, we establish an association between the class Process and the class Activity with the semantic rule Activity. We also establish the rule each process has activities in the association and we assign such a rule to the object property has_Activity. We then develop inferences understandable by the reasoner—Pellet—for deduction of relationships between classes by using object properties in the OWL semantic rules.

Figure 6. OWL semantic rules established in the SWRL Tab in the Protégé framework



Name	Rule
Activity	Process(?x) ^ Activity(?y) ^ has_Activity_Name(?x, ?a) ^ has_Name(?y, ?a) -> has_Activity(?x, ?y)
Input_Task_Product_Task	Task(?x) ^ Input_Task_Product(?y) ^ has_Input_Code(?x, ?a) ^ has_Code(?y, ?a) -> has_Input_Task_Product(?x, ?y
Objective	Process(?x) ^ Objective(?y) ^ has_Objective_Description(?x, ?a) ^ has_Description(?y, ?a) -> has_Objective(?x, ?y)
Ouput_Task_Product_Task	Task(?x) ^ Ouput_Task_Product(?y) ^ has_Ouput_Code(?x, ?a) ^ has_Code(?y, ?a) -> has_Ouput_Task_Product(?
Part	Standard(?x) ^ Part(?y) ^ has_Part_Name(?x, ?a) ^ has_Name(?y, ?a) -> has_Part(?x, ?y)
Process	Profile(?x) ^ Process(?y) ^ has_Process_Name(?x, ?a) ^ has_Name(?y, ?a) -> has_Process(?x, ?y)
Profile	Part(?x) ^ Profile(?y) ^ has_Profile_Name(?x, ?a) ^ has_Name(?y, ?a) -> has_Profile(?x, ?y)
Role_Job	Job(?x) ^ Role(?y) ^ has_Role_Code(?x, ?a) ^ has_Code(?y, ?a) -> has_Role(?x, ?y)
Role_Task	Task(?x) ^ Role(?y) ^ has_Role_Code(?x, ?a) ^ has_Code(?y, ?a) -> has_Role(?x, ?y)
Standard	Member(?x) ^ creates_Standard_Name(?x, ?a) ^ Standard(?y) ^ has_Name(?y, ?a) -> creates_Standard(?x, ?y)
Task	Activity(?x) ^ Task(?y) ^ has_Task_Name(?x, ?a) ^ has_Name(?y, ?a) -> has_Task(?x, ?y)

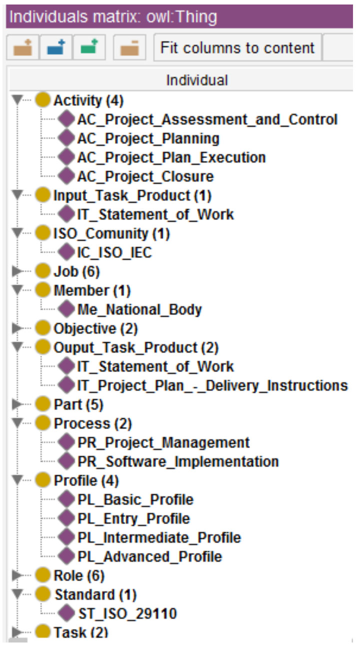
Source: Own production

We finally create individuals of every class for testing the ontology functionality and rule-based inference deduction of the reasoner as illustrated in Figure 7. We associate the respective data properties to each individual created. Besides, we define the data type and the value of every data property linked.

Results

After developing the OWL ontology based on the executable pre-conceptual schema, we present how the reasoner deducts conjunctions and relationships between individuals. In the previous Section, we specified each individual has data properties and it is associated with a specific class. Each class has a domain and a range and it is associated with SWRL semantic rules to object properties. Object Properties also have defined a range and a domain. With such elements and rules specified, we can observe—by starting the Pellet reasoner—how individuals are completely linked to each other and how new assertions of the object property are automatically constructed.

Figure 7. Individuals by Class in the Protégé framework.

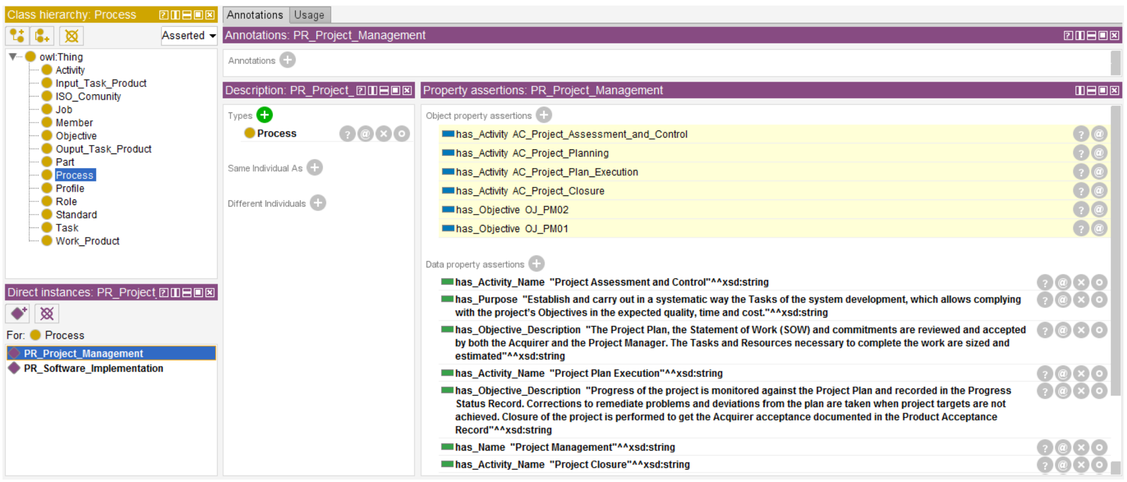


```
▼ Activity (4)
    AC_Project_Assessment_and_Control
    AC_Project_Planning
    AC_Project_Plan_Execution
    AC_Project_Closure
▼ Input_Task_Product (1)
    IT_Statement_of_Work
▼ ISO_Comunity (1)
    IC_ISO_IEC
► Job (6)
▼ Member (1)
    Me_National_Body
► Objective (2)
▼ Ouput_Task_Product (2)
    IT_Statement_of_Work
    IT_Project_Plan_-_Delivery_Instructions
► Part (5)
▼ Process (2)
    PR_Project_Management
    PR_Software_Implementation
▼ Profile (4)
    PL_Basic_Profile
    PL_Entry_Profile
    PL_Intermediate_Profile
    PL_Advanced_Profile
► Role (6)
▼ Standard (1)
    ST_ISO_29110
► Task (2)
```

Source: Own production

Likewise, we have the possibility to evaluate each inference explanation with the active reasoner. We show in Figure 8 some individuals created for each domain class. For example, the selected individual PR_Project_Management has four data objects defined—has_Name, has_Purpose, has_Objective_Description, and has_Activity_Name—with their respective string type values. We also show in Figure 8 how the reasoned infer six assertions related to object properties—four of them referred to the object property has_Activiy, and two of them referred to the object property has_Objective—by applying the previously created semantic rules. Such assertions are painted in yellow color.

Figure 8. Inferences deduced by the reasoner Pellet shown in yellow for the individual PR_Project_Management in the Protégé framework



```
has_Activity  AC_Project_Assessment_and_Control
has_Activity  AC_Project_Planning
has_Activity  AC_Project_Plan_Execution
has_Activity  AC_Project_Closure
has_Objective  OJ_PM02
has_Objective  OJ_PM01

has_Activity_Name  "Project Assessment and Control"^^xsd:string
has_Purpose  "Establish and carry out in a systematic way the Tasks of the system development, which allows complying
with the project's Objectives in the expected quality, time and cost."^^xsd:string
has_Objective_Description  "The Project Plan, the Statement of Work (SOW) and commitments are reviewed and accepted
by both the Acquirer and the Project Manager. The Tasks and Resources necessary to complete the work are sized and
estimated"^^xsd:string
has_Activity_Name  "Project Plan Execution"^^xsd:string
has_Objective_Description  "Progress of the project is monitored against the Project Plan and recorded in the Progress
Status Record. Corrections to remediate problems and deviations from the plan are taken when project targets are not
achieved. Closure of the project is performed to get the Acquirer acceptance documented in the Product Acceptance
Record"^^xsd:string
has_Name  "Project Management"^^xsd:string
has_Activity_Name  "Project Closure"^^xsd:string
```

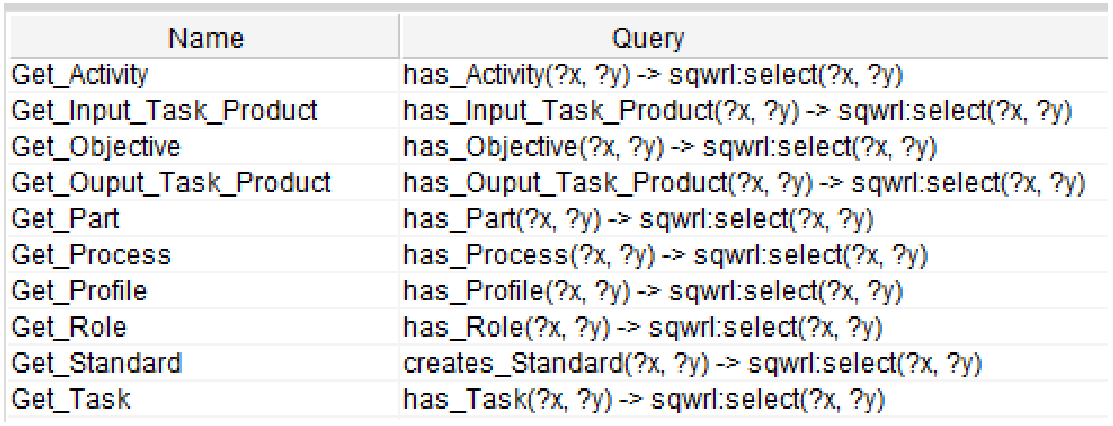
Source: Own production

We can create queries in OWL language by using the SQWRL tab in order to ask specific questions related to the ontology. The initial set of questions structured at the start point of the ISO/IEC 29110 basic profile are:

- Which is the standard information?
- What are the parts of the ISO/IEC 29110?
- What are the profiles of each part?
- What processes have the ISO/IEC 29110 Basic Profile?
- What are the objectives of each process?
- Which activities are related to each process have?
- Which tasks are related to each activity?
- What are the roles of each task and job?
- What are the input work products of each task and work product?
- What are the output work products of each task and work product?

We define the restrictions for answering such questions by creating the semantic rules in the SWRL tab. Besides, the SWQRL tab can be used for extracting information with simple queries as we propose in Figure 9.

Figure 9. OWL queries created for answering the initial set of questions in the SQWRL tab in the Protégé framework



Name	Query
Get_Activity	has_Activity(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Input_Task_Product	has_Input_Task_Product(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Objective	has_Objective(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Ouput_Task_Product	has_Ouput_Task_Product(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Part	has_Part(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Process	has_Process(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Profile	has_Profile(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Role	has_Role(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Standard	creates_Standard(?x, ?y) -> sqwrl:select(?x, ?y)
Get_Task	has_Task(?x, ?y) -> sqwrl:select(?x, ?y)

Source: Own production

As a summary, we are able to represent the ISO/IEC 29110 basic profile domain in a graphical way by constructing an executable pre-conceptual schema. We can interpret the profile core, hierarchy, priority, flow of processes, activities, tasks, roles, etc. with such a representation. Likewise, we create an equivalence table as a guide for obtaining ontology classes and properties by using previously established symbols of the executable pre-conceptual schema—i.e., concepts, relationships, implications, and conditionals. Finally, we are able to develop an OWL ontology in Protégé including SWRL semantic rules by following the statements on the equivalence table in order to restrict the open world.

We can automatically deduce new object property assertions by following the previous SWRL semantic rules. Besides, the reasoner can show inference explanation for analysis of the assertions deducted. We can also select each individual in the ontology for verifying the assertions deduced by the reasoner as a way to prove the consistency of the ontology. We are able to construct queries using the SQWRL tab in order to answer previous questions established about the ISO/IEC 29110 basic profile. Finally, we can add queries in order to answer new structured questions of the ontology by using the SQWRL tab.

We found some issues involving the Protégé framework along this study. We had visualization problems in certain tabs huddling new information specified, and issues with the reasoner when executing queries. Such issues can be solved by restarting the software. However, we can highlight the possibility of refactoring for avoiding inconsistencies in the ontology. The graphical environment is very intuitive and structured in a harmonic way, easing class and property construction. The SWRL and SQWRL tabs include an OWL debugger detecting invalid statements and guiding the creation of either a correct rule or query statement. Likewise, Protégé has a main debugger for rule inference with automated consistency checking for a new object, property, rule, or query created, allowing the immediate correction of any error.

Conclusions

We proposed an OWL ontological representation of the ISO/IEC 29110 basic profile standard domain by using Protégé. Previously, we developed an executable pre-conceptual schema as a graphical representation, and we proposed a set of equivalences between such a schema and OWL objects. We also developed a complete ontology containing a set of SWRL semantic rules for restrictions and SQWRL queries in order to respond established questions. The resulting ontology includes a set of individuals of each class with asserted relationships deduced by a reasoner. The results of this paper can be used by VSEs as a guide for implementing the basic profile of the ISO/IEC 29110 standard. This ontology can be also used in academia for achieving a better understanding and acknowledgment of such a standard [25, 26].

Future work should be related to tailoring the method used in this paper for constructing new knowledge representations based on ontologies and executable pre-conceptual schemas. Besides, we should consider the representation of related ISO standards in Protégé for harmonization purposes. The main goal should be the construction of a general domain suitable for general and specific elements common to different standards.

Acknowledgment

This paper is part of the results of the M.Sc. project entitled “A pre-conceptual-schema-based method for eliciting requirements in the context of ISO/IEC 29110,” from the Universidad Nacional de Colombia.

Bibliographic references

1. C. Laporte, S. Alexandre, R. O’Connor, “A Software Engineering Lifecycle Standard for Very Small Enterprises,” *European Conference on Software Process Improvement EuroSPI 2008*, vol. 16, pp. 129–141, Dublin, Sep. 2008. DOI: https://doi.org/10.1007/978-3-540-85936-9_12
2. ISO, “ISO/IEC TR 29110-5-1-2: 2011—Software Engineering—Lifecycle profiles for Very Small Entities (VSES)—Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile,” Geneva, 2011.
3. A. Díaz, C. de Jesús, K. Melendez, A. Dávila, “ISO/IEC 29110 Implementation on two Very Small Software Development Companies in Lima. Lessons Learned,” *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2504–2510, May 2016. DOI: https://doi.org/10.1109/TLA.2016.7530452
4. C. Y. Laporte, “The Development and Experimentation of an International Standard for Very Small Entities Involved in Software Development,” *International Council on Systems Engineering (INCOSE) Workshop*, Phoenix, AZ, Jan. 2011. DOI: https://doi.org/10.1002/j.2334-5837.2014.tb03167.x
5. M. L. Sánchez-Gordón, R. V. O’Connor, “Understanding the gap between software process practices and actual practice in very small companies,” *Software Quality Journal*, vol. 24, no. 3, pp. 549–570, Sep. 2016. DOI: https://doi.org/10.1007/s11219-015-9282-6
6. C. Y. Laporte, R. V. O’Connor, “Implementing Process Improvement in Very Small Enterprises with ISO/IEC 29110 A Multiple Case Study Analysis,” 1*0th International Conference on the Quality of Information and Communications Technology, Lisbon*, pp. 125–130, Sep. 2016. DOI: 10.1109/QUATIC.2016.033
7. L. García, C. Y. Laporte, Y. Arteaga, M. Bruggmann, "Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru," *Software Quality Professional*, vol. 17, no. 2, pp. 16–29, 2015.

8. V. Siddoo, N. Wongsai, R. Wetprasit, "An Implementation Approach of ISO/IEC 29110 for Government Organizations," *Lecture Notes in Computer Science,* vol. 7983, pp. 5–19, 2013.

9. R. O'Connor, "Early Stage Adoption of ISO/IEC 29110 Software Project Management Practices: A Case Study," *Software Process Improvement and Capability Determination, 14th International Conference, SPICE 2014 Vilnius. Communications in Computer and Information Science,* Lithuania, A. Mitasiunas, T. Rout, R. O'Connor, A. Dorling (Eds.), Springer-Cham, vol. 477, pp. 226–237, Nov. 2014. DOI: https://doi.org/10.1007/978-3-319-13036-1_20

10. S. Galvan, M. Mora, R. V. O'Connor, F. Acosta, F. Alvarez, "A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process," *Procedia Computer Science*, vol. 64, pp. 188–195, Oct. 2015. DOI: 10.1016/j.procs.2015.08.480

11. K. Villela, G. Santos, L. Schnaider, A, Rocha, G. Travassos, "The use of an enterprise ontology to support knowledge management in software development environments," *Journal of the Brazilian Computer Society*, vol. 11, no. 2, Nov. pp. 45–59, 2005. DOI: https://doi.org/10.1007/BF03192375

12. B. Henderson, C. Gonzales, T McBride, G. Low, "An ontology for ISO software engineering standards: 1) Creating the infrastructure," C*omputer Standards and Interfaces*, vol. 36, no. 3, pp. 563–576, Mar. 2014. DOI: 10.1016/j.csi.2013.11.001

13. C. Gonzales, B. Henderson, T McBride, G. Low, X. Larrucea, "An ontology for ISO software engineering standards: 2) Proof of concept and application," *Computer Standards and Interfaces*, vol. 48, pp. 112–123, Nov. 2016. DOI: https://doi.org/10.1016/j.csi.2016.04.007

14. R. Eito, "Ontology-based Tailoring of Software Process Models," *Terminology and Knowledge Engineering* 2014, Berlin, Jun. 2014.

15. R. Faiano, E. Souza, R. Falbo, M. Barcellos, "Software Testing Processes in ISO Standards: How to Harmonize Them?," *Simpósio Brasileiro de Qualidade de Software (SBQS)*, Rio de Janeiro, Aug. 2017.

16. R. Batres, M. West, D. Leal, D. Price, Y. Naka, "An upper ontology based on ISO 15926," *Computers and Chemical Engineering,* vol. 31, no. 5–6, pp. 519–534, May. 2007. DOI: https://doi.org/10.1016/j.compchemeng.2006.07.004

17. A. Vera. "A Pre-conceptual-schema-based method for eliciting requirements in the context of ISO/IEC 29110," M.Sc. Thesis, *Universidad Nacional de Colombia*, Medellín, Feb, 2019.

18. C. M. Zapata, F. Vargas, "Reglas Sintáctico-semánticas para Relacionar los Objetivos Organizacionales y los Problemas en el Contexto de la Educción Temprana de Requisitos de Software," *Revista latinoamericana de ingenieria de software*, vol. 1, pp. 01–07, Feb. 2013. DOI: https://doi.org/10.18294/relais.2013.01-07

19. C. M. Zapata, A. Gelbukh, F. Arango, "Pre-conceptual schema: a conceptual-graph-like knowledge representation for requirements elicitation", *Lecture Notes in Computer Science,* vol. 4293, pp. 17–27, Nov. 2006. DOI: https://doi.org/10.1007/11925231_3

20. H. Knublauch, "Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Protégé/OWL," *1st International Workshop on the Model-Driven Semantic Web*, pp. 381–401, Monterrey, CA, Sep. 2004.

21. M. O'Connor, H. Knublauch, S. Tu, B. Grosof, M. Dean, W. Grosso, M. Musen, "Supporting Rule System Interoperability on the Semantic Web with SWRL," *Lecture Notes in Computer Science,* vol. 3729, pp. 974–986, Nov. 2005. DOI: https://doi.org/10.1007/11574620_6

22. M. O'Connor, A. Das, "SQWRL a query language for OWL," *5th International Workshop on OWL: Experiences and Directions (OWLED 2009)*, Chantilly, VA, Oct. 2009.

23. C. M. Zapata, G. Giraldo, S. Londoño. "Esquemas preconceptuales ejecutables," *Revista Avances en Sistemas e Informática*, vol. 17, no. 2, pp. 15–23, 2011.

24. N. F. Noy, M Sintek, S. Decker, M. Crubézy, R. Fergerson, M. A. Musen, "Creating Semantic Web Contents with Protégé-2000," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 60–71, Mar. 2001. DOI: https://doi.org/10.1109/5254.920601

25. O. Y. Guerrero Jaimes y R. M. Guerrero Jaimes, "Las empresas de Norte de Santander y su perspectiva acerca de la seguridad y salud en el trabajo", *Investigación e Innovación en Ingenierías*, vol. 5, n.º 2, pp. 26-45, 2017. DOI: https://doi.org/10.17081/invinno.5.2.2755

26. B. E. Barreto Mardach y M. Marchena Rivera, "Incidencia del nuevo marco normativo de auditoría en el ejercicio de la revisoría fiscal en Colombia", *Dictamen Libre*, n.º 18, pp. 23–29, 2016.