

Article

UPCaD: A Methodology of Integration Between Ontology-Based Context-Awareness Modeling and Relational Domain Data

Vinícius Maran ^{1,*},†,‡ , Guilherme Medeiros Machado ^{1,‡}, Alencar Machado ^{2,‡},
Iara Augustin ^{2,‡} and José Palazzo M. de Oliveira ^{1,‡} 

¹ Instituto de Informática, Universidade Federal do Rio Grande do Sul, 91540-000 Porto Alegre, Brazil; gui.medeiros1@gmail.com (G.M.M.); palazzo@inf.ufrgs.br (J.P.M.d.O.)

² Centro de Tecnologia, Universidade Federal de Santa Maria, 97105-900 Santa Maria, Brazil; alencar.machado@ufsm.br (A.M.); august@inf.ufsm.br (I.A.)

* Correspondence: vmaran@inf.ufrgs.br; Tel.: +55-51-3724-8417

† Current address: Coordenadoria Acadêmica, Universidade Federal de Santa Maria, Cachoeira do Sul, Brazil.

‡ These authors contributed equally to this work.

Received: 20 December 2017; Accepted: 26 January 2018; Published: 30 January 2018

Abstract: Context-awareness is a key feature for ubiquitous computing scenarios applications. Currently, technologies and methodologies have been proposed for the integration of context-awareness concepts in intelligent information systems to adapt them to the execution of services, user interfaces and data retrieval. Recent research proposed conceptual modeling alternatives to the integration of the domain modeling in RDBMS and context-awareness modeling. The research described using highly expressiveness ontologies. The present work describes the UPCaD (Unified Process for Integration between Context-Awareness and Domain) methodology, which is composed of formalisms and processes to guide the data integration considering RDBMS and context modeling. The methodology was evaluated in a virtual learning environment application. The evaluation shows the possibility to use a highly expressive context ontology to filter the relational data query and discusses the main contributions of the methodology compared with recent approaches.

Keywords: context-awareness; data query; relational database; linking rules; methodology

1. Introduction

Context-Awareness is a key feature in recent information systems. It was developed from ubiquitous computing scenarios, as described by Mark Weiser [1]. Context-awareness is used in many fields in computer science as recommendation systems or human-computer interaction studies. Context-awareness may be defined as a set of applications or architectures that adapt and customize their execution based on information obtained from the environment [2]. In ubiquitous architectures, context-awareness may directly interfere in several operations, among them [3]: (i) Abstraction of Context Information; (ii) Association of Contexts with Data; (iii) Context-Based Discovery of Resources; (iv) Context-Based Actions; and (v) Context-Based Services Selection.

Recent research [4–6] demonstrated that context modeling based on ontologies is the approach that best meets important requirements for ubiquitous systems, such as extensibility and the level of formality required to represent contexts. Based on the context modeling developed in ontologies, tools and methodologies have been created to persist and recover this information in ubiquitous systems [7–9]. Ubiquitous systems uses many information sources, including ontologies and databases, to retrieve content related to the user [10]. However, not all the information considered in ubiquitous

systems is represented in ontologies, especially with respect to domain-specific information, which is frequently represented and persisted in relational databases were non-ubiquitous information systems were modeled [11]. Therefore to retrieve this domain data in a contextualized way it is necessary to create a form of interconnection between context information, represented in ontologies, and information about the domain of the application, stored in relational databases [11,12]. Some recent research proposed formalisms and tools [11,13] to provide CAR (Context-Aware Retrieval). CAR tools have been proposed in two different approaches: (i) integrate context-awareness in legacy systems to retrieve content; and (ii) integrate context-awareness in the modeling phase of information systems. However, there are no proposals of specific methodologies for the application of these formalisms as proposed in CAR models.

This paper presents the UPcaD (*Unified Process for Integration between Context-Awareness and Domain*) methodology. This methodology was defined based on a well known software engineering methodology (UP—Unified Process [14]) and in an ontology engineering methodology (UPON—Unified Process for Ontology Engineering [15]). Many of the processes used in these methodologies were reused, making the understanding of the methodology easy to the actors involved in the methodology application. The main objective of UPcaD is to guide, through a set of processes, the implementation of CAR models to integrate domain-specific data and context modeling.

The UPcaD methodology reuses the UP definitions of phases. UP is applied in software engineering and the focus of UPcaD is in the integration between context-awareness and domain data. UPON methodology extends UP for application in the ontology engineering field. The UPcaD methodology reuse a set of process originally defined in UPON in one of its workflows (Context Workflow).

To evaluate the proposed methodology, we applied it in a previously validated ontology network [16] that describes context information about students and content in a real MOOC (Massive Open On-line Course) about Motivational Interviewing using a set of formalisms to integrate context information and domain information [13,17]. These formalisms were defined as a set of linking rules, which are used in the workflows of the UPcaD. All the other formalisms of the integration (except the linking rules) are defined in this work.

The paper is structured as follows: In Section 2 the main concepts related to the modeling of context in context-awareness applications and recent research related to integration between context and domain information are presented. In Section 3 the research motivational real-world scenario is described. In Section 4 we present UPcaD methodology and the formalisms and processes that integrates the methodology. In Section 5 the evaluation of the methodology is presented with the application of the methodology in a legacy system, and a comparison of characteristics between UPcaD and related work is described. In Section 6 finally the conclusions and a list future work possibilities are presented.

2. Conceptual Foundation

In this section the main concepts about context-awareness and recent research about the integration between context and domain information are presented.

2.1. Context-Awareness

In ubiquitous Information Systems (IS) the data that describe the environment are collected by sensors, abstracted into sets, the full dataset is called context information. The most current definition of context is the definition of Dey et al. [2], where context was described as “*Any information that can be used to characterize the situation of entities (person, place or object) that are considered relevant for interaction between a user and an application (including the user and the application)*”. To be considered in IS, context must be represented and processed based in an abstract model representation. The most frequently used models for context representation in IS are [18]:

- Key-value model;
- Markup schemes;
- Object-oriented models;
- Graphical models;
- Models based on logic;
- Models based on ontologies.

Strang et al. [19], Knappmeyer et al. [20] compared the most commonly used forms of context representation. Spatial models are more efficient when compared to ontologies, but the models do not present high representation capacity when compared to ontological and object-oriented models. None of the analyzed forms was satisfactory in the question of information imperfection. Recent research [4,18,20] pointed that ontology-based models offers a series of advantages for context representation in IS.

Currently, there are standards to model ontologies, such as: RDF, RDFS, RDFa and OWL (OWL 1 and OWL 2 [21]). Most of these standards are managed by W3C [22]. The representation and inference capacity varies according to the representation language [19]. Several ontologies have been proposed to represent contexts in ubiquitous systems. The comparison presented in [23] summarized the concepts that represent context elements (A context element is equivalent to a context entity. It represents a type of element in the environment. An example of context element is a specific device, inserted in the environment [23]), represented in each ontology. It was observed in the comparative analysis that: (i) The SWRL language is widely adopted for the representation of first-order logical rules as a complement for OWL-DL ontologies; (ii) Recent ontologies use more abstract concepts, if compared to the first proposed ontologies for context-awareness; (iii) In parallel to the use of more generic expressions, there is a greater use of constraints and conditions of construction of individuals in more recent ontologies.

Rodriguez et al. [24] performed a comparison between ontologies that represent contexts and human activities. The comparison was made using a set of criteria, described as: (i) Learning curve; (ii) Definition of techniques and methods; (iii) Representation of social interaction; (iv) Sensor infrastructure and (v) Scalability. The comparison pointed that PiVOn ontology [25,26] and MiO! ontology [27] offer a more complete model to represent context information, compared to other proposed ontologies.

Context information can be used in several ways in IS. Context-Aware Retrieval (CAR) is one of this forms and defines that Information Retrieval (IR) and Information Filtering (IF) operations must be based on context information. In this way, stored information, whether in a structured or unstructured form, can be retrieved according to the informed context.

As the context involves large sets of information, defined in fields, it may also be considered a document in a collection. Contexts must be able to be queried as a common document related to the domain [28], therefore context may be used in IR [28] to: (i) Derive a query that returns the documents that best fit the context informed in the time that the query is made; (ii) Treat context as a document, that is, the context becomes the source of information to be queried.

Specifically in relation to item (i), the integration between context and domain-specific information allows an integration between what is known (knowledge modeling related to the application domain and the application itself) and what is done (modeling and use of the environment context) by IS [29]. Explicitly or implicitly, knowledge modeled in computational systems has contextual components; these components may be used to filter, change the focus or reduce information in queries [29].

2.2. Linking Context-Awareness and Relational Data

Several research works proposed models to integrate context-awareness modeling and domain information modeled in RDBMS [11,13,16]. CAR approaches can be divided into the following clusters [30]:

- **Use of visions in data modeling:** uses context information to avoid ambiguities in data interpretation under different conditions, allowing entities to assume different values according to context. Into this category are included the works proposing extensions for the data model and extensions of query operators to create visions according to the context information [11];
- **Database partitioning:** uses context information to classify groups of entities according to the context information. It is a strategy considered old and used only by the first proposals [30];
- **Information filtering:** uses context information to select relevant data in databases. This type of approach is related to recommendation systems, usually with the implementation of techniques that allow users to associate classifications to a context-based recommendation, and if a classification is considered acceptable similar recommendations are made when the context is similar to the one reported previously. This type of approach is the most used in CAR systems and are generally proposed as extensions of query languages that uses context information [13,17,31].

CAR approaches contemplate the execution of a set of two-step steps that are common in context-aware systems [30]:

- **Modeling phase:** The context model is defined according to scenarios that describe the possible contexts where the application will be inserted. Afterwards, the context model is associated with domain model and they are related to contexts of interest. The domain model describes the specific knowledge of an application, for example, entities in a database, business rules, relevant services, among others;
- **Execution phase:** The system recognizes the current context of the environment and if this context corresponds to one of the contexts of interest of the application. If the current context corresponds to a previously modeled context, the knowledge associated with the context is used by the system.

2.3. A CAR Model

A model to integrate context and domain information was presented in [13,17]. This model has as main advantages compared to other proposals:

- Provides the support to ontology-based context models, even with models of context in use (without the necessity to redefine the context model);
- Provides support to legacy IS, through the use of linking rules that are used with the relational queries predefined in the IS;
- Provides support to a semi-automatized form of linking between context and domain information.

Figure 1 presents the components of the CAR model. It is composed by ontologies, linking rules (explained in Section 2.3.3) and a set of algorithms, associated with each linking rule.

To perform a context-based query, a Query Extension Algorithm [13] receives: (i) A relational query, described in the SQL language, and (ii) A context of interest, described in a set of axioms in the OWL-DL language. The algorithm performs an extension of the SQL query, execute the query in the RDBMS and return a set of contextualized data. This extension process is performed for each query and use the relational query and the ontological description of the context of interest.

To contextualize the queries, the model needs the previous execution of three processes in the modeling stage: one related to the context modeling, other related to domain modeling and another related to the creation of linking rules.

In relation to the context, is suggested that a **Generic Context Ontology** should be used. This kind of ontology is extended by designers to represent contextual information and contexts of interest required for contextualized queries. After this extension, the **Extended Context Ontology** is processed in a **Context Integration Algorithm**. This algorithm integrates the extended context ontology and the concepts defined in the CI (Context Integration) Ontology [16]. The algorithm has as output a **Resultant Context Ontology**, which in turn is applied in the expansion of queries and in the definition of linking rules.

an algorithm integrates this ontology and the Domain Integration (DI) ontology. This algorithm [17] produces a new ontology, which in turn is used by the query extension algorithm.

In CAR model, context-aware queries were classified into three types: (i) Domain as Context; (ii) Domain for Context element and (iii) Domain for context value. Each one of these types is related to a linking rule (the definition of each rule is presented in Section 2.3.3).

2.3.3. Linking Rules

A set of linking rules were created to define relations between context and domain information using the formalism defined in [13].

These rules are used by an algorithm in the execution phase to extend relational queries. To clarify the use of the linking rules in the UPCaD methodology, we present a summary of each rule and the consequence of the use of the rules.

Let us consider D as a relational schema, R as a relation of D , t a tuple of R , C a class of context of an ontology that describe context, ci_p a data property of C , $rae(D)$ a relational algebra expression over D , val a specific value for ci_p , di_p as an attribute of the domain, , where $di_p \in \{rae(R) \vee NULL\}$ and ind as an individual of C type.

- The **Domain and Context** (\leftrightarrow) rule implies that domain information can be also considered as a context information by an IS. The $R(t) \leftrightarrow C$ rule implies that an individual will be created in the ontology to represent the information related to the tuple of domain schema. The domain information may be considered as a context information.

As an example of the rule, let us consider a relation module, presented in Table 1. This relation represents the information about a part of a Massive Open On-line Course (MOOC). Each module is identified by an id attribute, has a $display_name$, which represents the module name, a sorted attribute, which represents a required object or a module to complete the course, a parent attribute, which represents whether the module is a submodule of another, more generic, one $format$ attribute, which represents the format without which the module is, an $attribute$ type, which represent the module is the main type (main) or is of another type.

Table 1. Relation module in a Massive Open On-line Course (MOOC) platform [13].

Tuple	ID	Display_Name	Graded	Parent	Format	Type
1	100	Clinical Case 1	1	NULL	html	main
2	101	Clinical Case 2	1	NULL	html	main

To make the integration between context and domain, the designers need to associate new information to each module, which was not previously modeled in the relation. In this example, we will consider that each module must have, in addition to the information presented in the relation, a data that indicates which specific concept the module is dealing with. Thus, designers previously modeled a *CoursePart* class on the context ontology, which is related to the *LearningFocus* class.

Thus, the definition of the connection rules $module(tuple1) \leftrightarrow CoursePart$ and $module(tuple2) \leftrightarrow CoursePart$ results in the classes and individuals presented in Figure 2. After the definition of the rules, it was possible to create a semantic relation between $module101_$, which represent a module in the domain database and the concept $general_concept_learning$, related to the class *LearningFocus* in an ontology that represent context information.

- The Domain for a Specific Value of Context (\rightsquigarrow) rule was created to relate an expression in relational algebra with a data property of a class that represent a context entity in the ontology of context. The expression $C_{(ci_p, val, di_p)} \rightsquigarrow rae(D)$ is similar to $\sigma(di_p = val)(rae(D))$.
- The Domain for a Specific Context Element (\rightarrow) rule was created to associate an expression of relational algebra with a specific element of context, represented by an individual in the context

ontology. The expression $C_{(ci_i)} \rightarrow rae(D)$ defines that $rae(R)$ is incorporated to the original query when the context individual ci_i is part of the informed context of interest.

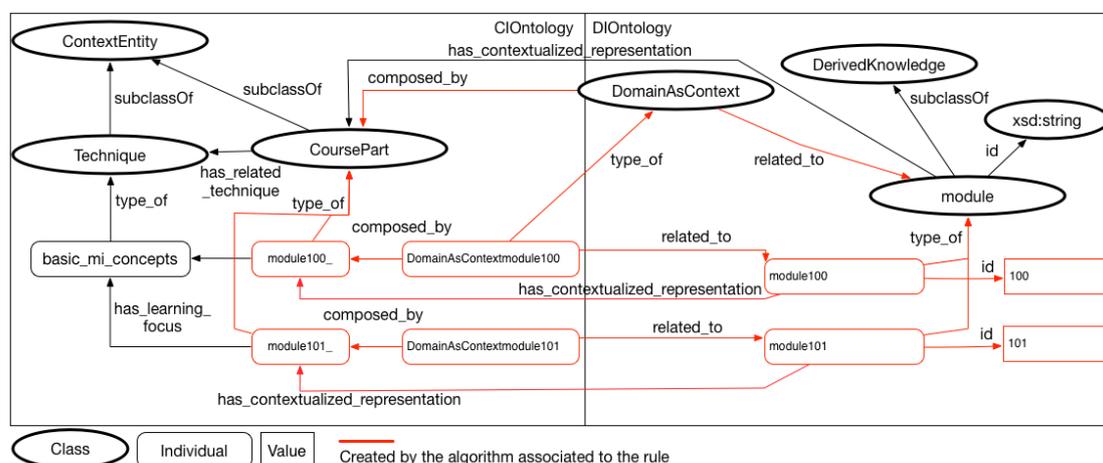


Figure 2. Ontology fragment after the creation of domain and context linking rule [13].

Considering the relation presented earlier in Table 1, let us imagine that for a decision of design, only the module of type main should be presented to users that uses smartphones to access MOOC courses. To make this filtration, the modeler defined the rule $Device_{(smartphone)} \rightarrow \sigma(type = 'main')(module)$.

To clarify the application of a CAR model in an existing environment, we present in the next section the motivational scenario of this research, presented earlier in a position paper [17]. This scenario is used in the evaluation section and evolves the usage of real systems.

3. Research Motivation—Recommendation of Resources in Smart Universities

Education is evolving constantly in relation to teaching processes. Recently, technologies such as mobile computing have contributed to a greater dissemination of information and educational support materials [34]. As there is a large information volume and materials available to teachers and students, it is necessary the addition of filters in the queries on this information. This procedure will restrict the results to the student’s field of study and to the context at the query time [34].

Currently, teachers assemble and distribute materials in Virtual Learning Environments (VLEs). These materials are distributed and activities are performed in these environments by students always in the same way, disregarding the importance of the context variables involved in performing these tasks. This is know as the model “one size fits all”. The problem is accentuated in MOOCs, mainly because there are larger variations in relation to cultural and geographical issues when compared to a traditional distance course. This variety of factors directly influence the low completion rate in these courses, which has been 5 to 9% [35]. Thus, it is necessary that contextual information be taken into account in the way these materials are distributed to the students of these courses [36].

Recent work [37,38] presented intelligent university environments, which are composed by a set of resources and intelligent systems that manages these resources and recommend these resources to students according their necessities. These system are linked to an ubiquitous middleware, which provides abstractions for applications to utilize features and technologies present in ubiquitous environments.

In general the ubiquitous middleware is responsible to manage context information that describes the environment [6]. Figure 3 shows an overview of the entities and technologies used in a scenario based on the use of CARLO recommender system [38].

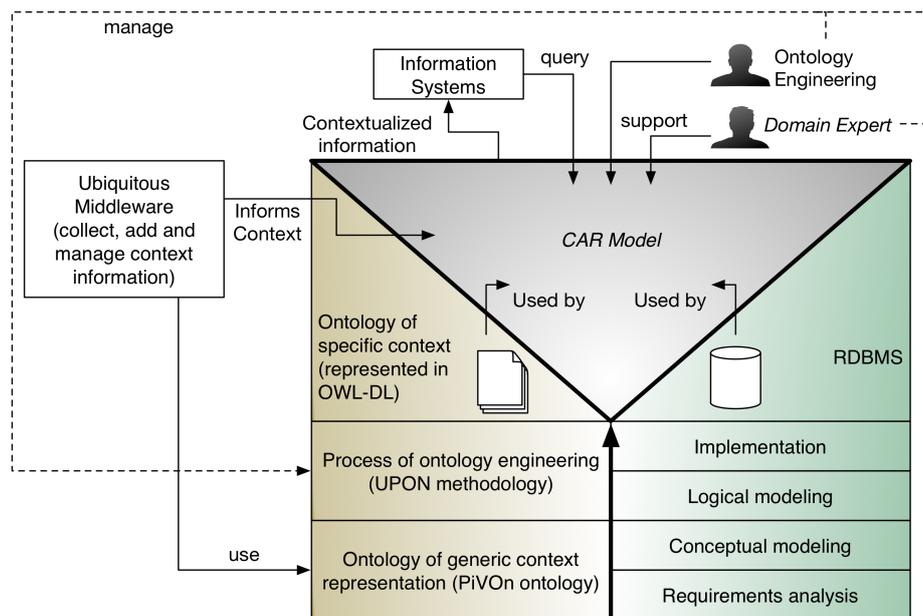


Figure 3. Overview of a scenario of recommendation of resources in a smart university campus.

As may be observed, CARLO uses the context information provided by an ubiquitous middleware. The context is modeled in an ontology network, based on PiVOn Ontology [25]. The ontology was extended from PiVOn to represent specific context of interest of CARLO. In parallel a MOOC platform, used by the university, uses a RDBMS to store content about MOOC courses and statistics about the use of the courses by students. To use the context information provided by CARLO in the MOOC platform, integration models as presented in [11,13,17] and are used to avoid a total re-engineering in the IS.

Motivational Scenario

Based on the concepts and technologies presented earlier, let us imagine the following scenario, based on a description of a real-world application in [16,17]:

“A university uses an ubiquitous middleware to manage the educational resources present. This middleware uses an ontology as a context representation model. This ontology contains a set of axioms that, together with a set of processes managed by the middleware and allows data collected from the environment to be aggregated, and after inferences, considered as context information. Thus, the context management of interest in this ubiquitous middleware infers the situations in which students and teachers are. This university also has a recommendation system that presents warnings to students about educational resources that may be of interest. Therefore, the context of interest of each student and teacher is inferred and send to the recommendation system, which recommends to the student or teacher an interesting resource available at the university.

A MOOC in the area of psychology and medicine is composed of cases of Motivational Interview (MI). Motivational Interview is a patient-centered approach to demonstrate behavioral changes to increase effectiveness in treating diseases such as obesity, smoking, and depression. The group of techniques that make up a MS has been constantly studied in a wide range of health-related behaviors [39]. To prepare the MOOC on motivational interview, the technical staff assembled the course with a structure composed of topics, where each topic consists of a supporting material (video, document or presentation) and a set of questions. If the student correctly answer the questions related to that topic, the progression to the next one is allowed and new motivational interview situations are introduced. The student completes the course when has finished all the tasks related to

the basic concepts and has finished more than 50% of the exercises related to the specific motivational interview cases applied to different contexts (illness to be treated, motivation, among others).

Johan is a student of the university's collective health course who has the ubiquitous middleware and resource recommendation system. At the time of enrollment, Johan registered his smartphone in the system of the university and from there began to receive recommendations of resources. Johan lives in the city of Smartville. Currently, Johan attends the 6th semester, and in a class of Applied Psychology to Health, the system of recommendation used by the university sends an alert to Johan in his smartphone recommending to enroll in the MOOC on Motivational Interviews. During the break of class, John visualizes the alert and enrolls in the Motivational Interview MOOC. At this point, the MOOC asks Johan for permission to receive context information managed by the university's ubiquitous middleware. Johan authorizes and the MOOC begins to receive constant updates of contexts of interest."

During a class break, Johan accesses the MOOC using a smartphone and notes the existence of 3 conceptual introduction topics.

4. UPcad Methodology

The UPcad (Unified Process for Integration between Context-Awareness and Domain) methodology is defined as an extension of UPON [15] and UP (Unified Process) [14] methodologies. UPON was defined for the creation of large scale ontologies. As the representation of context information is frequently made using ontologies, UPCaD is proposed as a methodology to guide the execution of models that uses context information in data querying.

4.1. Methodology Overview

The methodology is composed of evolutionary implementation cycles and composed of workflows that describe a series of operations. The number of implementation cycles varies according to the designers' need to associate: (i) New context elements in information filtering; or (ii) A new source of information (database) is inserted in the filtering process.

Each workflow has inputs and outputs. In each deployment cycle, five workflows, named: Context Workflow, Domain Workflow, Alignment Workflow, Serialization Workflow and Query Test Workflow, must be executed. The methodology overview is presented in Figure 4.

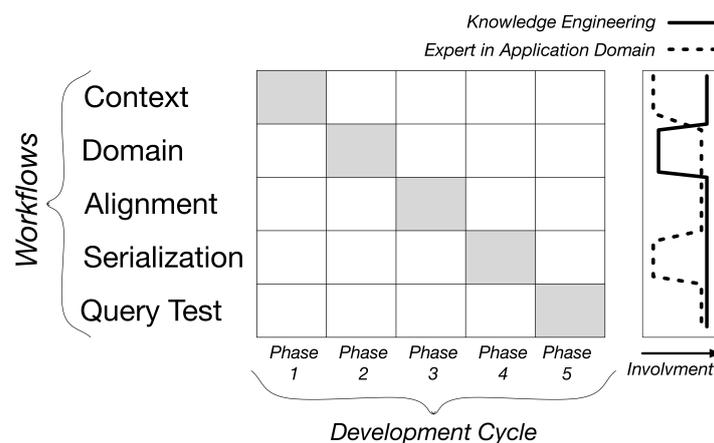


Figure 4. Overview of Unified Process for Integration between Context-Awareness and Domain (UPCaD) phases.

The support of experts in the field of application and ontology engineers varies according to each one of the workflows that compose the phases of the methodology. The structure of workflows for each phase with inputs and outputs was based on the UPON methodology [15], which is used directly only in Phase 1 (Context Workflow).

4.2. Context Workflow

Context Workflow is a set of steps to define or extend an ontology that describe context to be used in query extension by applying a query contextual model. This workflow follows the steps defined in the UPON Methodology, with some modifications in the order of execution and creation of new process in this methodology.

An overview of the Context Workflow processes is presented in Figure 5. The processes not highlighted in the figure were defined in the UPON Methodology, and are described as follows:

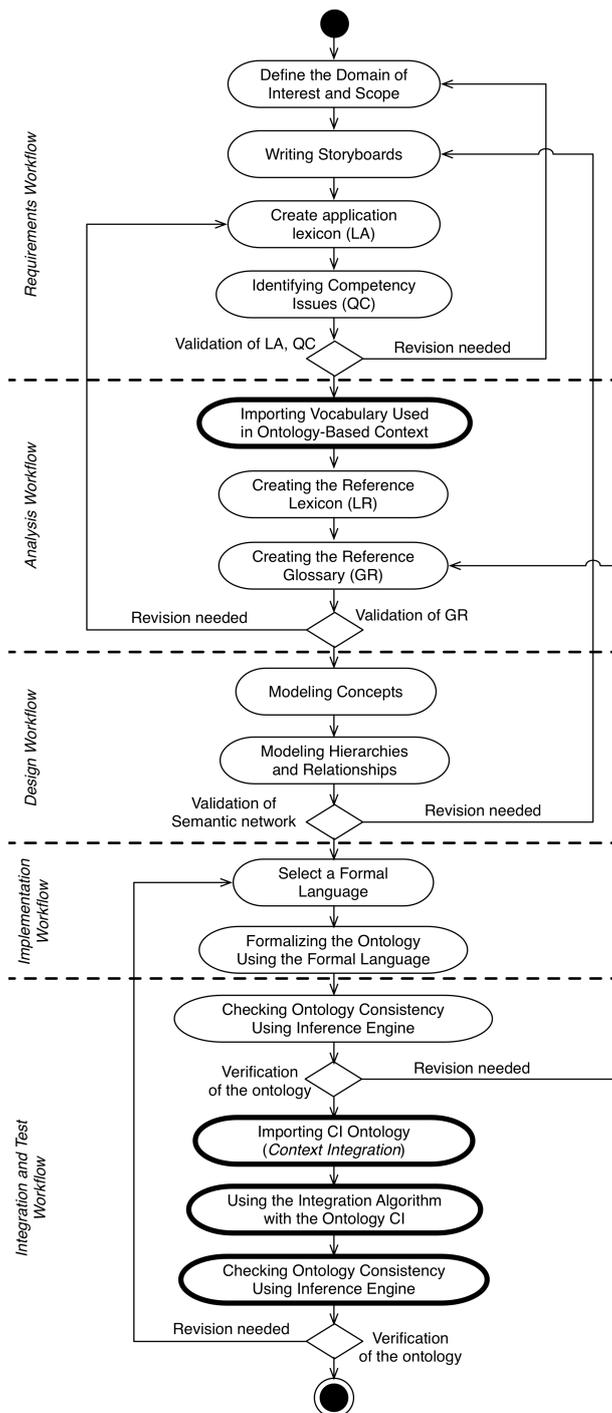


Figure 5. Overview of Context Workflow.

- **Definition of the Domain of Interest and Scope:** It consists of the identification of the main concepts that must be represented, their characteristics and the definition of the domain's scope that will be represented. For this, ontological commitments must be made in the process.
- **Writing Storyboards:** In this step the application domain expert wrote one or a series of *Storyboards* that describe sequences of activities in a given scenario. This scenario describes a set of application scenarios and contexts;
- **Creation of the Application Lexicon (LA):** The lexicon of the application is a set of the main terms contained in documents, collected from the storyboards developed in the previous process;
- **Identifying Competency Issues (QC):** Competency issues are conceptual-level issues that the resulting ontology must be able to respond to. They are defined based on interviews with domain experts, users, and developers. They define the coverage and depth of the ontology representation scope over the modeled domain. In the UPON Methodology, there are two types of Competency Issues defined: (i) oriented to the discovery of resources or content; and (ii) semantic interoperability between different schemes. In the developed methodology only questions of the first type are considered;
- **Importing Vocabulary Used in Ontology-Based Context Modeling:** It is envisaged to import the vocabulary from a context representation ontology at this stage to avoid the redundancy of concepts in the modeling phase;
- **Creating the Reference Lexicon (LR):** The reference lexicon is defined from the set of concepts described in the application lexicon, in the imported vocabulary of the context ontology and in the information of the documents used in the previous stages;
- **Creating the Reference Glossary (GR):** The reference glossary is defined by adding informal definitions (sentences about the concept) to the LR using natural language;
- **Modeling Concepts:** Each concept is categorized through the association of type with the concept. The concepts are categorized using the context ontology as top ontology;
- **Modeling Hierarchies and Relationships:** At this stage the categorization of concepts described in a taxonomy is complemented with domain-specific relations, aggregation relations and generalization;
- **Formalization of the Ontology Using the Formal Language:** In this step the concepts and relationships modeled in the design workflow are formalized in the OWL-DL language. This formalization is accomplished through the definition of an ontology, which imports the concepts of the context ontology, forming a network of ontologies.

To perform the ontology validation a check-list must be performed, according to different characteristics: (i) Syntactic quality; (ii) Semantic quality; (iii) Pragmatic quality and (iv) Social quality [15];

- **Checking Ontology Consistency Using Inference Engine:** Checking the consistency of the ontology must be performed using an inference engine. Here the existence of contradictions in the definition of the ontology are discovered. If there is any contradiction, the inference engine informs the ontology engineer, who should review the process of creating the ontology;
- **Importing CI Ontology:** In addition to importing the context ontology, it is necessary to import the CI ontology. Thus, the concepts created in the extended context ontology should be sub-concepts of the classes of the CI ontology;
- **Using the Integration Algorithm with CI Ontology:** After the importation of the ontologies' vocabularies, the designers must integrate these ontologies, generating a net of ontologies. This network of ontologies will be used in the next steps of the workflow.

After the application of the Context Workflow, a network of ontologies is generated, with verified consistency through the use of an inference engine and of ontology engineers through the processes of verification of the ontology coverage and the answer of questions of competence. This ontology network is used as input to the Domain Workflow, shown below.

4.3. Domain Workflow

Domain Workflow defines a series of processes for integrating domain information in the model. An overview of the workflow is presented in Figure 6. The workflow consists of the following steps:

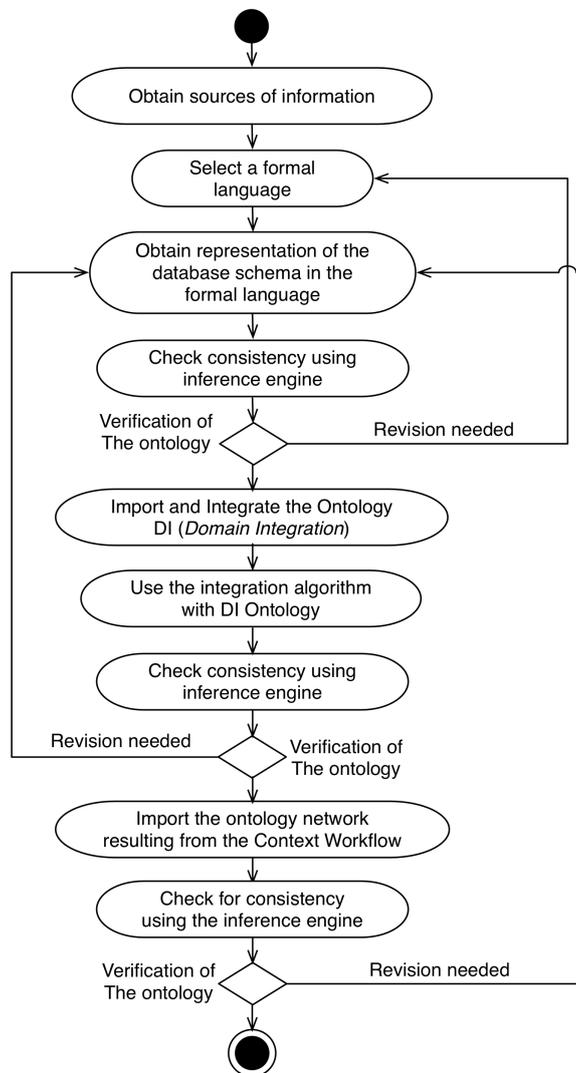


Figure 6. Overview of Domain Workflow.

- **Obtain sources of information:** At this stage the domain expert must obtain the information sources that will be used in the model development. As the focus of this research is the query extension model, a database schema must be employed as the source of information in this step;
- **Select a formal language:** In this step, the ontology engineer makes the choice of a formal language that will represent the database schema. As the ontology network generated in Context Workflow is represented in the OWL-DL language, it is recommended to choose the OWL-DL or RDF languages in this step for compatibility purposes;
- **Obtain representation of the database schema in the formal language:** Because the context representation in the model is made in the OWL-DL language, the database schema used in the model must be represented in the same language. Currently, there are tools capable of converting relational schemes to OWL-DL files through R2RML mappings. RDBToOnto is an example of a tool that is used in several projects to do this [33];
- Check consistency using inference engine;

- **Import and Integrate the DI Ontology:** After the first verification, the ontology engineer imports the DI ontology;
- Check the consistency using the inference engine;
- **Import the ontology network resulting from the Context Workflow:** The ontology network generated in the Context Workflow is imported and integrated in the ontology network generated in the Domain Workflow;
- Check for consistency using the inference engine.

After these steps, the context and domain definitions are interconnected through a network of ontologies. This allows binding rules to be set to interconnect the settings at the time of the data query.

4.4. Alignment Workflow

In Alignment Workflow the stages of defining a series of contexts are taken into account in the process of extending queries and the queries involved in these processes. An overview of the workflow is presented in Figure 7.

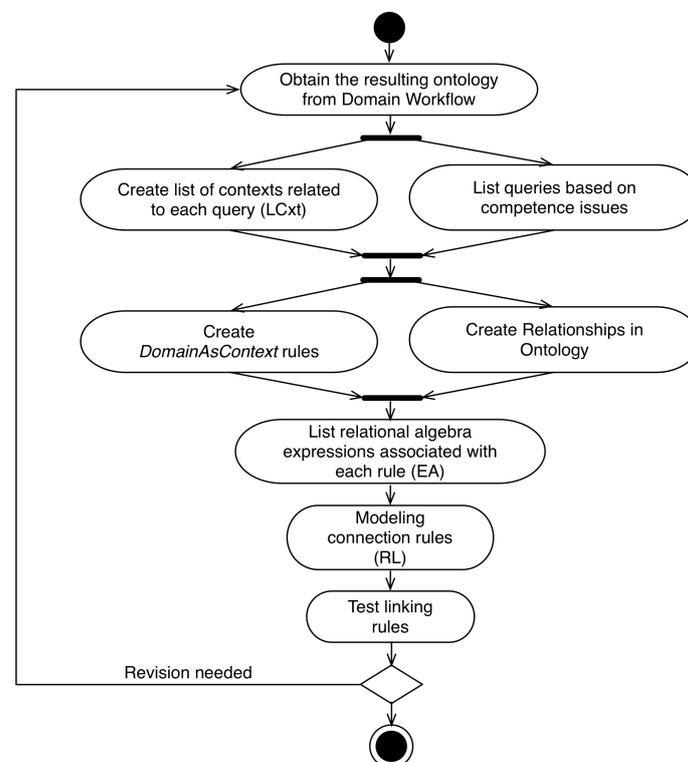


Figure 7. Overview of Alignment Workflow.

The workflow consists of the following steps:

- Obtain the resulting ontology from Domain Workflow;
- **Create list of contexts related to each query (LCxt):** Based on the competence issues, the application domain expert and the ontology engineer create a list of: (i) context entities; (ii) context attributes and (iii) semantic contextual relations, which are related to the competence issues defined earlier. These elements will be used in the definition of linking rules;
- **List queries based on competence issues:** In this step, the expert in the application domain will create a list (LC), the set of queries that are used in the system and that may be contextualized;
- **Create DomainAsContext linking rules:** In this step, if necessary, the modeler creates the rules of DomainAsContext type. These rules are different in relation to other rule types because they

create new individuals in the network of ontologies, allowing new semantic relationships to be created;

- **Create Relationships in Ontology:** If domain and context rules are created, new individuals are created in the ontology network. After the creation of these individuals, new semantic relationships can be defined in the network of ontologies with these new individuals;
- **List relational algebra expressions associated with each rule (EA):** From LCxt and LC, the expert in the application domain defines a series of relational algebra (EA) expressions that represent the filtering that must be performed according to each context element, for each query;
- **Modeling linking rules (RL):** Using LCxt, LC and EA, the domain expert models the linking rules, using the definitions presented previously in [13,17];
- **Test linking rules:** After the definition of the linking rules by the modeler, the set of defined rules and the network of ontologies generated in the process are serialized in a DB instance. In this step, the modeled rules are tested. In this testing step, rules are tested only in terms of syntax and semantics. These tests are done through contexts of interest, entered manually, and verification of the return of the queries by the domain expert.

4.5. Serialization Workflow

Serialization Workflow deals with the set of processes required to assure the persistence of the definitions made previously (definition of the connection rules and the network of ontologies). A view of the Serialization Workflow is presented in Figure 8.

The workflow consists of the following steps:

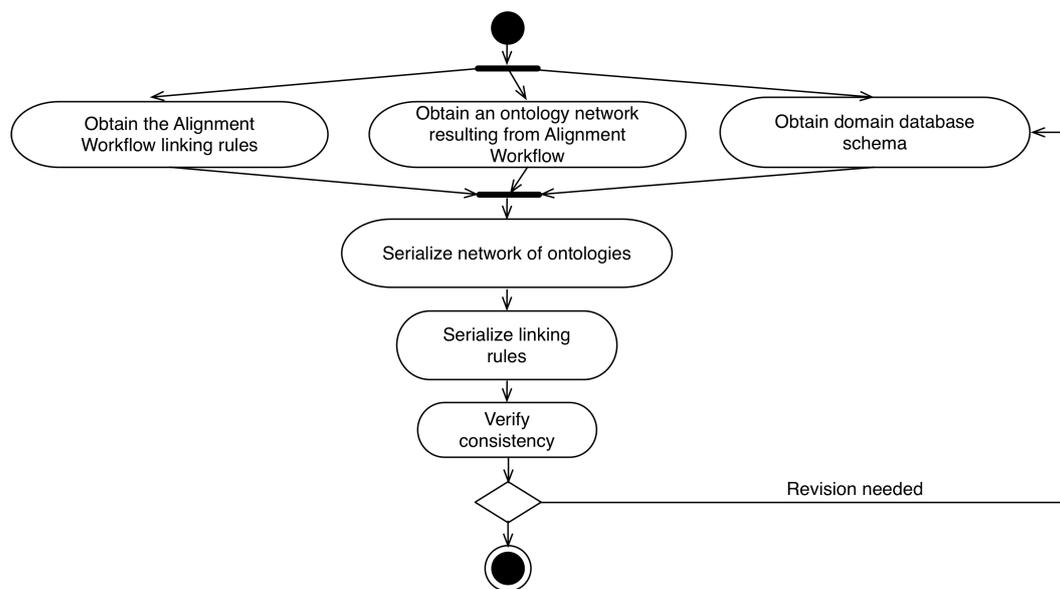


Figure 8. Overview of Serialization Workflow.

- **Obtain the Alignment Workflow linking rules;**
- **Obtain an ontology network resulting from Alignment Workflow;**
- **Obtain domain database schema;**
- **Serialize network of ontologies:** After the step of collecting the information generated in previous processes, the network of ontologies is serialized in a format compatible with the model. In the prototypes developed to support the model, we used the serialization of ontologies in the JSON-LD format;

- **Serialize linking rules:** The serialization of binding rules occurs in the execution of the algorithms associated with each rule. Each of the algorithms generates a JSON-LD file that is persisted in the same instance of the relational database that stores the domain information;
- **Verify consistency:** After realizing the persistence of the ontology network used by the model, one must verify the consistency of the information. To perform this check, it is recommended to use an inference engine.

After the persistence of information, the methodology provides for the execution of tests, to evaluate the application of the model in the system. The workflow that describes the evaluation is presented in the next section.

4.6. Query Test Workflow

To perform the evaluation of the application of the model in the system/application scenario, it is necessary to integrate the query extensor algorithm to the system and carry out the evaluation. Two processes must be performed after applying the model to the system:

- **Apply query expansion algorithm:** The application of the query expansion algorithm is performed by associating the integrator module with the information system;
- **Verify answers to competence questions:** In this step, the expert in the application domain verifies, through the accomplishment of queries or verification with the users, if the queries answer the questions of competence listed in the first stage of the methodology.

An overview of the workflow is presented in Figure 9.

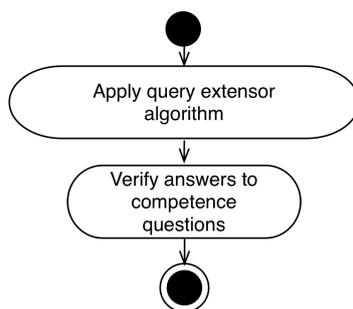


Figure 9. Overview of Query Test Workflow.

5. Evaluation

To evaluate the UPCaD methodology, we applied it in the scenario presented in Section 3, based on the use of CARLO recommender system and the MOOC platform used in a university. Figure 10 presents the overview of the process that composes the workflow. Each process was assigned with a number that are referenced in the text that describes the evaluation. The evaluation process of the methodology was based in the evaluation used by related works [11,15]. To evaluate the methodology, we implemented the SQL-eCO plugin for Protégé. Figure 11 shows an example of interface in SQL-eCO plugin (More information about the developed plug-in is available at: <https://github.com/viniciusmaran/SQL-eCO-plugin-public>. The plugin offers a set of services to modelers create linking rules and test it in relational queries. The plug-in was developed in Java, using the set of Protégé software API to manage and verify consistency of ontological models. The plug-in is composed by four user interfaces, which are used to configure the three types of linking rules and configure and test relational queries using the extension of the queries. After defining and testing the rules in the plug-in, modelers can export the definitions for a particular RDBMS schema).

To make the evaluation of the methodology, five actors developed the implementation of the methodology. Three actors are related to the development of the CARLO recommender system.

They are ontology engineers who maintain CARLO ontologies. The other two actors are RDMBS administrators, who makes queries and maintain the RDBMS with the information about the MOOCs.

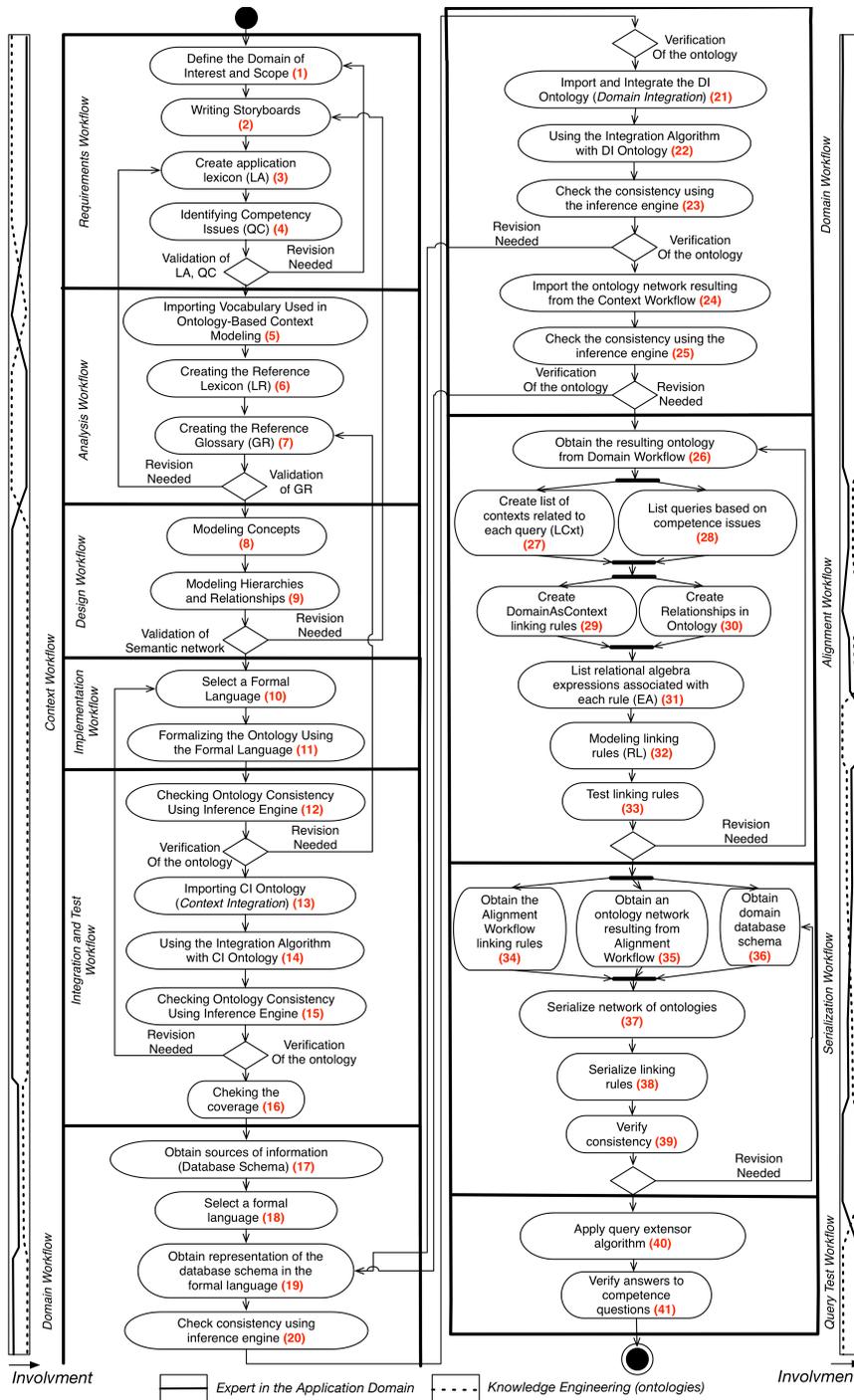


Figure 10. All process that composes the Unified Process for Integration between Context-Awareness and Domain (UPcad) Methodology.

It is important to notice that we not used recall and precision measures in this paper as the focus of the paper is the methodology and not the evaluation of the linking rules or the CAR model. The evaluation of the model, considering specifically the linking rules was presented in [13].

The first step in the integration process is to determine the domain of interest and the scope of the definition of the extended context ontology (1) from the use of the ontology employed in the ubiquitous middleware.

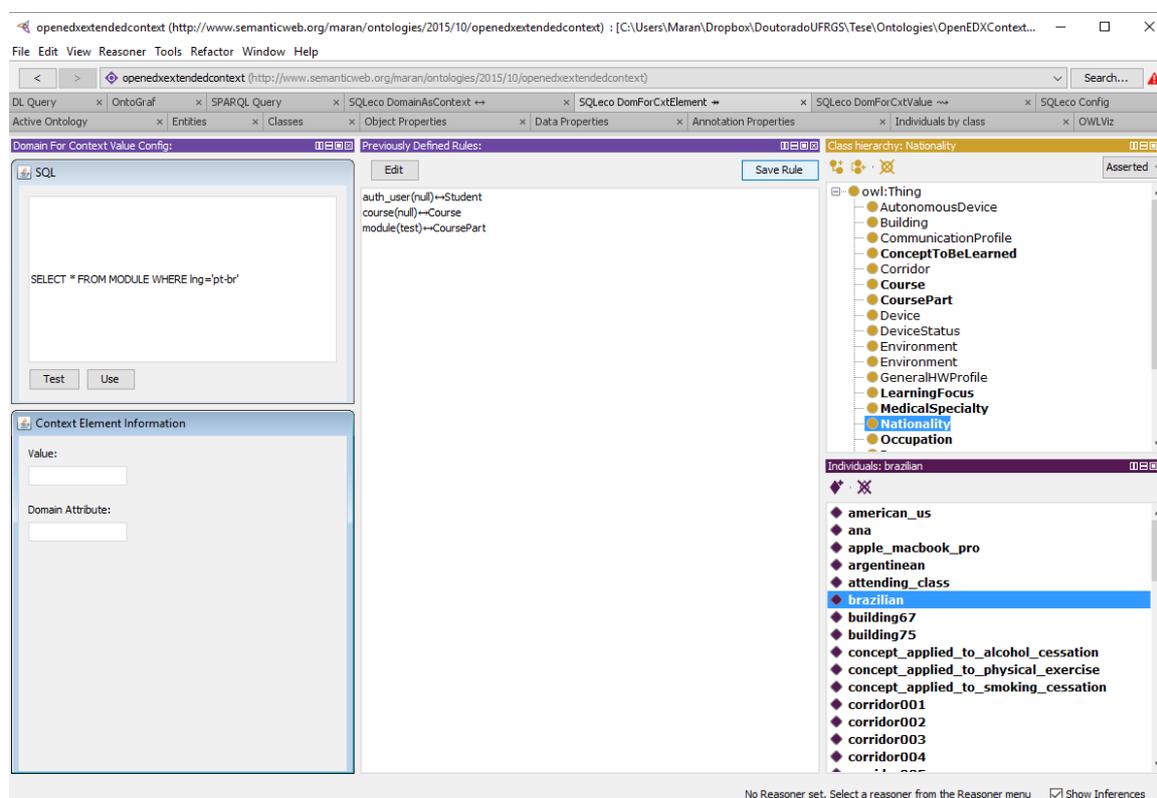


Figure 11. An interface to create linking rules in SQL-eCO (SQL-extension based on Context-Awareness) plugin.

5.1. Application of the Context Workflow

Considering the application scenario presented in Section 3, the domain of interest of the context ontology is the presentation of modules of the motivational interview course according to context information from the ontology used by the ubiquitous middleware. Scope is defined as contextualized selection of modules and support materials according to specific student situations, context information related to profiles, devices, location, student interests and learning focus.

From the definition of the domain of interest and scope, a set of storyboards (2) was defined that describes the situations of use of the system, with the context involved in each of these situations. The storyboards are taken from the description of the application scenario.

- **Storyboard 1:** Johan is a public health student at the university campus. He registers and accesses the motivational interview MOOC during the break of one of the classes through his smartphone while staying at the cafeteria. As Johan enrolled in the course and makes the first access, he is presented only to course modules related to learning basic concepts of motivational interviewing. Since Johan is in the range of classes, he uses a device with limited viewing capabilities and the range is 15 min, no backing material with videos longer than 10 min is displayed;
- **Storyboard 2:** Johan is a public health student on the university campus. He enrolled in the MOOC on motivational interviewing, and during one of his undergraduate classes, he introduced the MOOC to some of his colleagues. As Johan is in class, the MOOC presents only general information about the course, such as the presentation about the course and the discussion forum;
- **Storyboard 3:** Johan is a public health student on the university campus. Johan is interested in areas such as promoting physical activities for patients and combating smoking. This interest is due to two main

factors: (i) Johan's father smokes; and (ii) John has a close friend with behavioral obesity. Johan goes to the computer lab after lunch and accesses the motivational interview MOOC through one of the lab computers. Johan has already completed activities related to learning basic concepts of motivational interviewing. Thus, the parts of the course related to the concepts of "listening to patient's motivation", "resistance to correction reflex" and "empowering the patient" are presented to John. We present only the cases where the focus is the promotion of physical exercises or the fight against smoking.

After defining the storyboards, the Application Lexicon (3) was defined, presented in Table 2. The application lexicon was defined based on queries about the concepts related to the course of MS and the context ontology.

Table 2. Application Lexicon.

Names				
Student	Course	Device	MOOC	
Motivational Interview	Classroom	Interval	Building	
Classroom	Class	Laboratory	Course Part	
Smartphone	Desktop	Computer	Video	
Empowering the Patient	Device	Duration	Smoking Combat	
Correction Reflection Resistance	Hall	Presentation	Discussion	
Promotion of Physical Activity	Forum	University	Basic concepts	
Listen to the patient's motivation	Campus	Quiz	Support material	

Based on the previously created storyboards and the definition of the application logic, it was defined as competence issues able to respond (4) after the application of the linking rules. The set of competence and evolution issues in Table 3.

Table 3. Competence questions.

Question ID	Description
QC1	What support materials from the motivational interview course should be presented to students when they are in the class interval?
QC2	What support materials from the motivational interview course should be presented to students when they are accessing the course using smartphones?
QC3	Which cases of the motivational interview course should be presented to students who are interested in the topics "combating smoking" and "incentive to exercise"?
QC4	Which course cases should be presented to the student when he/she is accessing the course during a face-to-face class?
QC5	Which cases of the motivational interview course should be presented to the student in the first three months of the course?

After defining competence issues the context ontology vocabulary is imported (5). The vocabulary of the context ontology consists of the names of the represented classes, object properties, and data properties. Table 4 presents the vocabulary terms imported from the ontology.

Table 4. Vocabulary imported from PiVOn ontology (limited to facilitate the visualization).

Imported Concepts			
Ability	Activity	Contact	Device
Entity	Publication	System	User
Work	Role	Learning Object	Expertise
GPS	Organization	Academic	

The next step in the implementation of the methodology is the creation of the reference lexicon (6). The lexicon is generated by the intersection of the lexicon of the application with the vocabulary of the context ontology. From the reference lexicon, the reference glossary (7) was defined, with a description of the meaning of each of the terms.

The reference glossary generated (Table 5) is employed in the modeling of concepts (8) and relations (9) in the network of ontologies. Each glossary term is classified according to its type of information and related to the terms already defined in the context ontology. The semantic network of concepts is presented in Figure 12. As can be seen in the figure, some of the concepts are modeled as classes, while other concepts as individuals in the semantic network. In addition, relationships are created between the new definitions and the existing definitions in the context ontology. The settings created in this process is displayed in the Figure 12.

Table 5. Glossary of reference.

Term	Description
MOOC	Online courses made available to a large audience, which is generally not geographically limited.
Course	Course offered on a MOOC platform.
Course Part	A part of course offered on a MOOC platform. Parts of courses are composed of support elements defined on the platform
Motivational Interviewing	Main course topic used in the application scenario
Basic Concepts	Student's learning focus on the course during the first access
Smoking Combat	Subject of interest of the student about cases of motivational interview that present situations of combat to smoking
Promotion of Physical Activity	Subject of interest of the student about cases of motivational interview that present situations promotion of physical activity
Correction Reflection Resistance	One of the techniques used in motivational interviews
Listen to the patient's motivation	One of the techniques used in motivational interviews
Empowering the Patient	One of the techniques used in motivational interviews
Presentation	One of the types of support material associated with a part of the course on the MOOCs platform
Movie	One of the types of support material associated with a part of the course on the MOOCs platform
Discussion Forum	One of the types of support material associated with a part of the course on the MOOCs platform
Quiz	One of the types of support material associated with a part of the course on the MOOCs platform

The language (10) for the representation of the semantic network is the OWL-DL language, to maintain compatibility with the representations used by the context ontology. The ontology network is formalized using Protégé (11). After the formalization of the ontology network the consistency of the ontology network is verified using the Pellet inference engine (12).

The CI ontology was imported (13) and integrated (14) to the ontology network generated in the previous process using the SQL-eCO plugin. This network of ontologies is used in the next stages of the methodology.

The ontology coverage check (16) is done by the expert in the application domain and by the ontology engineer. This verification is performed by verifying whether all contexts of interest were modeled on concepts in the ontology. After defining the network of context representation ontologies, the workflow related to the application domain is executed.

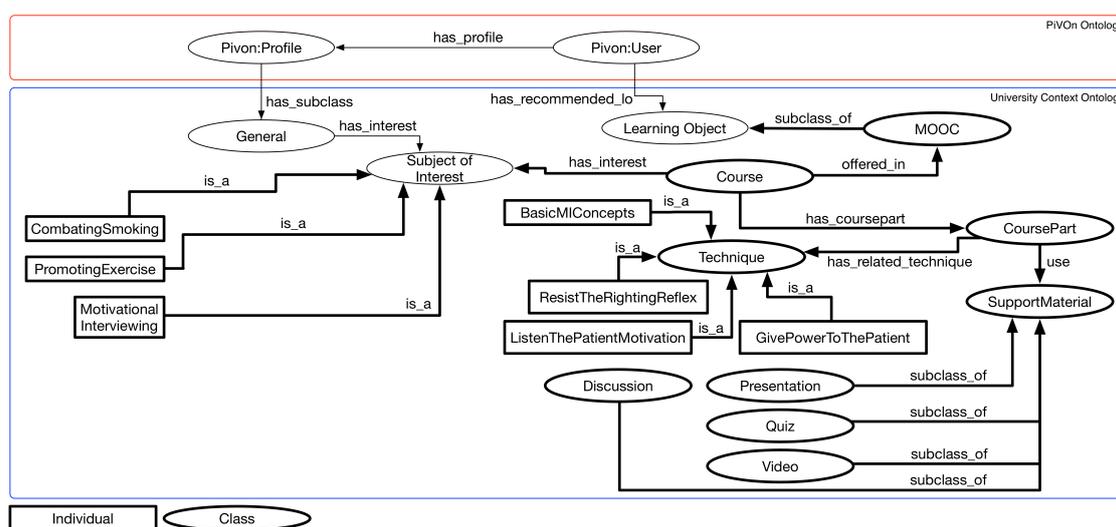


Figure 12. Semantic Network generated in the Context Workflow.

5.2. Application of the Domain Workflow

THE access the database schema (17) through the MySQL is configured using an SSH connection. The language for the formalization of the schema (18) is the OWL-DL language to maintain compatibility with the context ontology used in the middleware. The OWL-DL representation of the database schema (19) is obtained using the RDBToOnto tool. Table 6 presents an overview, in number of elements, of the OWL-DL resulting from the conversion process.

Table 6. Overview (in number of occurrences) of the OWL-DL file generated.

Classes	Object Properties	Data Properties	Axioms
130	262	620	3654

The OWL-DL representation is validated syntactically and semantically (20) using the Pellet inference engine. After this verification, the SQL-eCO tool is used to import (21) and perform the integration (22) of the DI ontology with the OWL-DL representation of the database schema. As output from this process, an ontology network is generated and stored. After completing this step, the consistency of this network of ontologies is verified with the Pellet inference engine (23).

The ontology network resulting from this process contains only representations relative to the application domain. To be able to create binding rules considering an ontology network it is necessary to integrate the ontology network resulting from Context Workflow (24). This integration is accomplished through the SQL-eCO plugin defining the location of the ontology networks resulting from Context Workflow and the integration between the DI ontology and the OWL-DL representation of the domain database schema. As a result of this process, a network of ontologies is generated. This network of ontologies has the consistency verified (25). The ontology network is used as an Alignment Workflow to create linking rules. The process of creating the rules is presented in the next section.

5.3. Application of the Alignment Workflow

From the ontology network generated in the previous workflow and the domain database, the binding rules can be defined and tested. To do this, the file representing the resulting ontology network was opened in the Protégé software (26) and a connection to the domain database is configured with the SQL-eCO plugin.

To model the rules of connection it is necessary to list the context elements of interest related to each competence issue (27). Table 7 presents the context elements related to each of the competence questions and their classification in the context ontology (class, object property, data property or individual).

Table 7. Context elements of interest related to each competence question.

Competence Question	Elements of Context of Interest	Context Element Classification
QC1	in_class_break	Individual of UserSituation class
QC2	limited_viewing = true	Data property of Device class
QC3	smoking_cessation, exercise_promotion	Individuals of SubjectOfInterest class
QC4	in_class	Individual of UserSituation class
QC5	basic_ML_concepts	Individual of Technique class

In addition to the context elements listed, the consultations related to each issue of competence are listed (28). These queries are listed in this step as they are used by the information system, without modifications. Table 8 presents these queries.

Table 8. Queries used in the system related to each issue of competence.

Competence Question	Related SQL Query	Query ID
QC1, QC2	<pre>SELECT t1.idmodule AS lev1, t2.idmodule AS lev2, t3.idmodule AS lev3, t4.idmodule AS lev4, def.definition AS lev4definition FROM module_child AS t1 LEFT JOIN module_child AS t2 ON t2.idmodule = t1.idchild LEFT JOIN module_child AS t3 ON t3.idmodule = t2.idchild LEFT JOIN module_child AS t4 ON t4.idmodule = t3.idchild JOIN module ON t4.idmodule = module.id JOIN module_definition AS def ON module.definition=def.id WHERE t1.idmodule = '57bb8878e4efae083b63c157'</pre>	Query1
QC3, QC4, QC5	<pre>SELECT course.id, t1.idmodule AS lev1, t2.idmodule AS lev2, module_definition.definition FROM auth_user JOIN student_courseenrollment ON (auth_user.id = student_courseenrollment.user_id) JOIN course ON (student_courseenrollment.course_id = course.id) JOIN module_child AS t1 ON (t1.idmodule = course.module_id) LEFT JOIN module_child AS t2 ON (t2.idmodule = t1.idchild) JOIN module ON (t2.idmodule = module.id) JOIN module_definition ON module.definition = module_definition.id WHERE auth_user.id = 5 AND course.id = 'course-v1:UnivTest + EM101+2016_1'</pre>	Query2

Queries are repeated on different competence issues, as some of the competence questions are related to the same portion of domain data, but vary in context elements. In the application scenario, the QC1 and QC2 competency questions use the same query, which returns the tree structure of the modules and their submodules, as well as the JSON definition of these modules.

The QC3, QC4, and QC5 questions also use the same query returning the basic course structures, with the main modules of the course carried out by the student that has their id informed in the WHERE clause of the query. Since none of the course modules were related to any MS technique and the definition of individuals in the Technique class, it was necessary to define the DomainAsContext rule (29) for each of the main modules of the motivational interview

course , associating them with the CoursePart class. An example rule created is shown below: *module(846d7c85e3654c4691910c8831fcc7f5) ↔ CoursePart*.

After defining the rules and applying them using the SQL-eCO plugin, the ontology network is modified by inserting the individuals that represent the modules in the context ontology. With the creation of these individuals, it was possible to associate each course part with a Technique class individual, who represents a MS technique (30). Figure 13 shows an example of the has_interest relationships performed between the individuals of the CoursePart class and the Technique class.

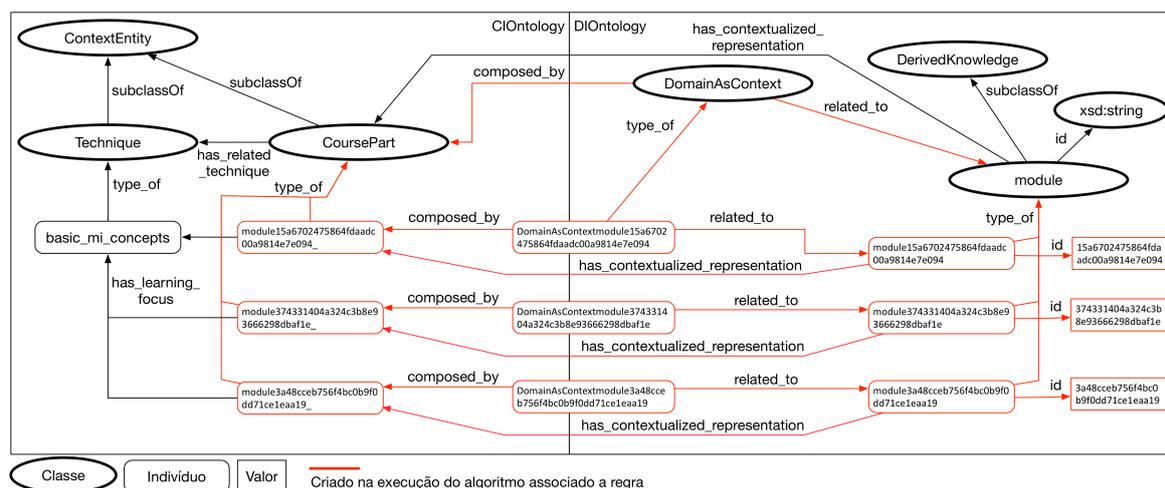


Figure 13. Relationships between individuals in the CoursePart and Technique classes.

After defining DomainAsContext rules and realizing the necessary links in the ontology, the linking rules to the context elements of interest related to competence issues are modeled (31).

Each query has a query identifier, which is used later in the definition of linking rules.

Query1 has been set to allow all course modules to be displayed, but limiting the display of modules containing videos that are less than 10 min (600 s) long.

Query2 was defined to limit only the presentation of modules that do not have associated videos, due to the limited capacities of devices that are in the context associated with competence issue 2.

Queries3 and 4 were defined to present the modules that represent motivational interview cases related to the promotion of physical exercise and the fight against smoking. The Query5 has been set to only allow the visualization of the modules of type type, vertical or overview. Thus, the basic structure of the course is presented to the student.

The SQL-eCO plugin is used to model the connection rules (32). The rules are modeled selecting the context element of interest related to each rule and the definition of the query in SQL. Table 9 shows the modeled linking rules.

After modeling the linking rules, they are executed with the SQL-eCO plugin (33). The tests related to the binding rules are performed using OWL-DL files that represent a context of interest based on the context ontology resulting from the Context Workflow and the realization of queries associated to each of the contexts of interest. To perform the tests in this phase of the methodology, the contexts of interest related to each of the storyboards describing the application scenario are modeled. Contexts of interest are modeled with Protégé. The first context of interest (Figure 14) is related to the situation presented in storyboard 1.

Table 9. Linking rules used in the evaluation.

Linking Rule	Function of the Rule
<p><i>UserSituation(in_class_break)</i> → SELECT module.*,FROM module JOIN module_definition ON (module.definition = module_definition.id) WHERE module_definition.definition-> “.block_type” <> ‘video’ OR module_definition.definition-> “.block_info.duration” <600</p>	Return all the modules with videos with less then 10 min
<p><i>Device(limited_viewing,true,NULL)</i> ~→ SELECT module.*,FROM module JOIN module_definition ON (module.definition = module_definition.id) WHERE module_definition.definition-> “.block_type” <> ‘video’</p>	Return all the modules with no videos
<p><i>SubjectOfInterest(smoking_cessation)</i> → SELECT module.*, FROM module JOIN module_definition ON (module.definition = module_definition.id), WHERE module_definition.definition-> “.block_type” = ‘overview’ OR module_definition.definition ->“.block_type” = ‘vertical’ OR module_definition.definition->“.block_type” = ‘course’, OR module.id = ‘3a4ce1eaa19’</p>	Return all the modules related to smoking cessation
<p><i>UserSituation(in_class)</i> → SELECT module.*,FROM module JOIN module_definition ON (module.definition = module_definition.id), WHERE module_definition.definition ->“.block_type” = ‘overview’ OR module_definition.definition -> “.block_type” = ‘course’,OR module.id = ‘3a4ce1eaa19’ OR module.id = ‘3e9a6124059c62’</p>	Return only the modules related to course main structure

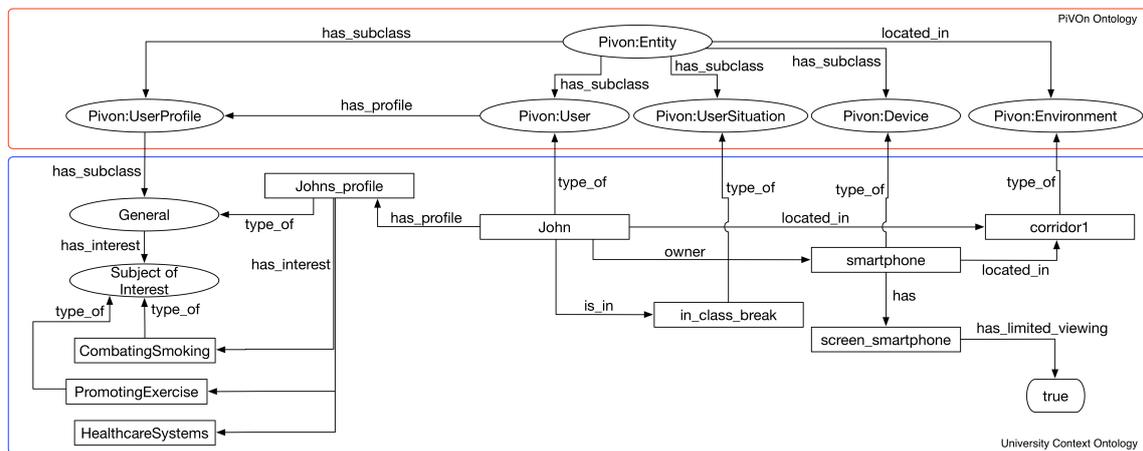


Figure 14. Context of interest related to Storyboard 1.

The second context of interest (Figure 15) is related to the situation presented in storyboard 2.

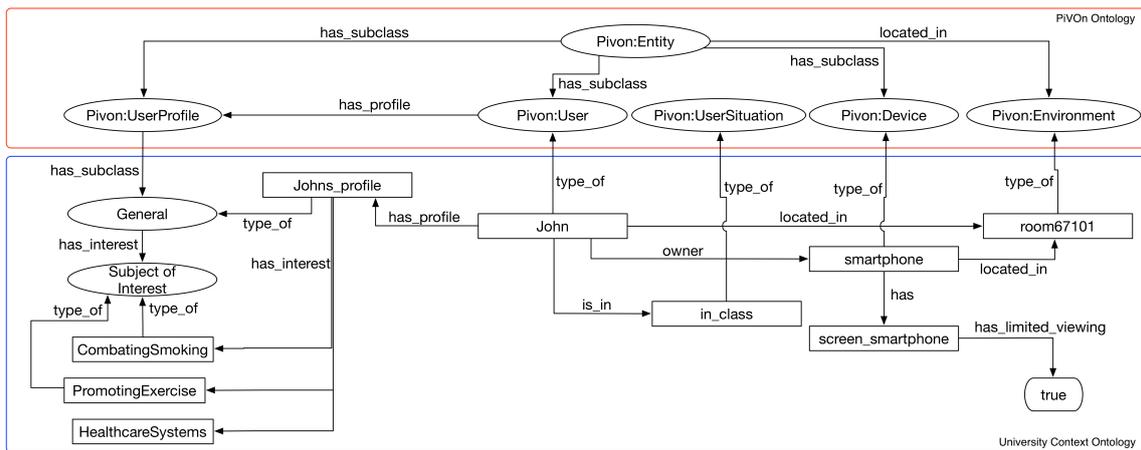


Figure 15. Context of interest related to Storyboard 2.

The third context of interest (Figure 16) is related to the situation presented in storyboard 3.

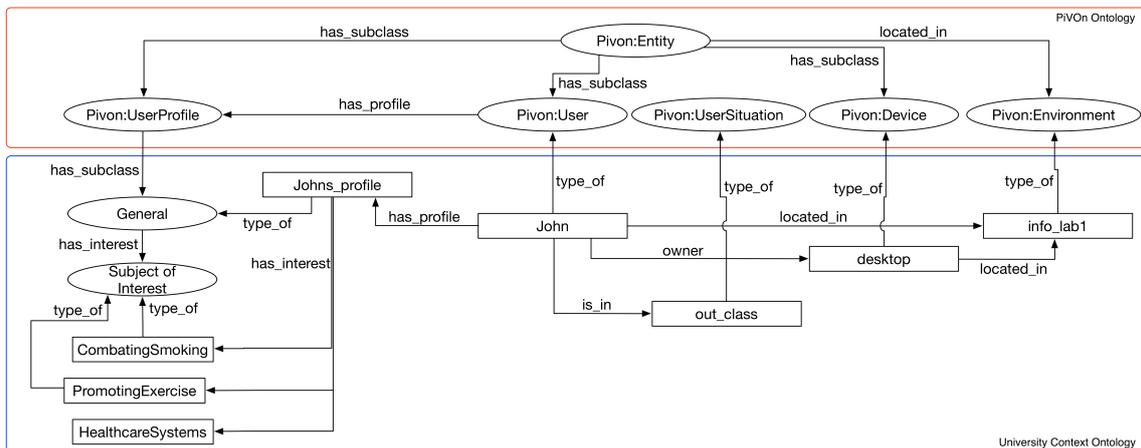


Figure 16. Context of interest related to Storyboard 3.

In this phase, only the absence of empty sets is verified as a result of the queries. To carry out this verification, the queries Query1, Query2, Query3, Query4 and Query5 are tested to verify if they represent the expressions related to each of the competence issues.

In addition, Cons1 and Cons2 queries are also tested. Verifications are carried out with the three contexts of interest in conjunction with the two relational queries. Verification of competency responses is performed with the Query Test Workflow. After verifying the connection rules, the Serialization Workflow is performed.

5.4. Application of Serialization Workflow

The process of serializing the definitions used by the framework is executed with the SQL-eCO plugin. In the configuration tab, the ontology locations used in the ontology network (35) are marked and the database connection (36) is configured. In addition, the file containing the definitions of the connection rules (34) is used. The serialization of the JSON (37) format is performed and then persisted in the MySQL relational database. In addition, the linking rules (38) and persistence in the same database instance are serialized.

To verify the consistency of the definitions after their persistence in the relational database (39), the SQL-eCO plugin is used in conjunction with the Pellet inference engine. No consistency errors

were found. After completing this step, the query extension tests were performed to evaluate the returned tuples after the application of the extended query algorithm. The testing process is presented in the next section.

5.5. Application of Query Test Workflow

The application testing and evaluation workflow is performed in two parts: (i) Applying the expansion query algorithm (40) and (ii) Verifying responses to competence questions (41).

To apply the query expansion algorithm, an API must be imported into the information system project in which it is desired to perform the query extension. Thus, the queries used in the system are not performed directly through the JDBC driver originally used, but through the API that redirects the query to the extender. The API was imported into the prototype that implements a query simulator. After this stage, the evaluation of the consultations and the results of these consultations were carried out after the application of the framework, varying the context of interest.

The extension queries evaluation is performed analyzing the returned tuples in the queries comparing the results of these queries with the queries that would normally be performed and the expected results of these queries. This comparison is made based in competence issues. Initially, we analyzed the quantities of tuples returned in the queries comparing the quantities returned in each query with and without the framework, considering each of the contexts of interest presented in Alignment Workflow. Figure 17 shows the number of resulting tuples in each of the queries, considering the queries performed with and without the framework considering each of the three contexts of interest based on the application scenario.

It may be perceived in Figure 17 that there was a decrease in the number of resulting tuples when the framework was used in conjunction with a context of interest to the extended query algorithm. Regarding the realization of queries in the context of interest (a), there was a decrease of 14.84% in the number of tuples returned in the Cons1 query, while in the Cons2 query there was no change in the number of tuples returned. This is because the query extension, in this case, filters only the video-type course modules, whereas the Cons2 query returns only course structure definition modules.

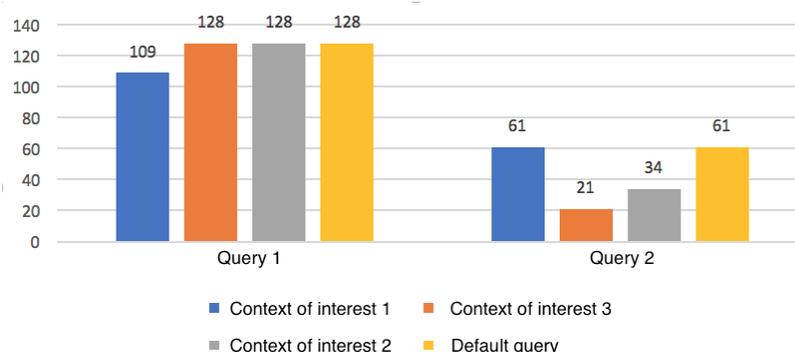


Figure 17. Results of query test.

Considering the contexts of interest (b) and (c), there were changes only in query Cons2. This is due to the fact that this query returns course elements that describe sections and cases of motivational interviewing. The Cons1 query in turn returns only the modules that describe the last level of the course structure tree, with the modules that describe the supporting materials. Considering the context of interest (b); there was a decrease of 65.57% in the number of tuples returned. Considering the context of interest (c), there was a decrease of 44.26% in the number of tuples returned.

In addition to the quantitative analysis of the tuples returned in each query considering the context of informed interest, the results of the queries were analyzed considering the competence issues defined in the Context Workflow. The result of the analysis is presented in Table 10.

Table 10. Analysis of contexts and queries related to each competence issue and the results of each query.

Competence Question	Related SQL Query	Context of Interest	Query Result
QC1, QC2	Query 1	Context of Interest (a)	All video-type modules that have a duration greater than 600 were not included in the query.
QC3	Query 2	Context of Interest (c)	Only the modules and sub-modules of the course associated with the interests of physical activity promotion and smoking control were returned in the consultation.
QC4, QC5	Query 2	Context of Interest (b)	Only the course modules that present the basic structure of the course (overview and discussion) or that are associated with the generic learning of MI techniques were returned in the query.

It is possible to perceive that the results of the queries tested with the contexts of interest derived from the storyboards used in the application methodology are compatible with the competence questions highlighted from the storyboards.

It was possible to observe that some of the defined linking rules were defined based in context of interest that could be used in other application domains. As an example we can cite the linking rules defined for competence question 2 (QC2), which was related to device screen. Based on this fact, we can assume that the application of the methodology can be applied in other application domains.

In the evaluation process, it was possible to observe that some of the competence questions (QC3 and QC4) were defined to filter educational content according to student's context.

It is important to note that in the evaluation of UPCad, no tests were performed in relation to the evolution of the data scheme. As predicted in other works related to the CAR area [7,11], the evolution of the data schema causes a revision in the forms of integration.

6. Conclusions and Future Work

Context-awareness has been applied in several ways in ubiquitous systems, such as in choosing services for execution, adapting graphical interfaces and retrieving content. The sensitivity to context modeled in ontologies presents a series of advantages over other models of representation and for this reason, it has been constantly used in works related to the ubiquitous computing.

In this context, this paper presented the UPCaD methodology, based on well known software engineering methodologies to guide the integration between the modeling of context and RDBMS data querying. The evaluation of the methodology in the scenario using the CARLO recommender system and the RDBMS which stores the information about MOOC courses in the university showed the validation of the methodology as a guide for integration between context and domain data.

It was observed during the evaluation of the methodology that a great effort was made by the team of ontology engineers and experts in the application domain to define the connection rules.

According to the team who applied the methodology, which have experience with the development of ontologies for context-aware applications, and the RDBMS administrators, the UPCaD methodology aid the use of CAR model. It was mentioned that the usage of common process with UP methodology helped in the learning curve of the methodology. As future works, we pretend to verify and measure the learning curve, varying the team who applies the methodology and the application domain.

With the evaluation of UPCaD methodology, the following findings were obtained:

- The strength of the proposed approach lies in the UPCaD being a methodology based on well known process, provided by UP and UPON methodologies, to guide the implementation of recent CAR models;

- The definition of the methodology based on workflows provided the possibility of multidisciplinary teams to work together, with the involvement of each team varying in each workflow;
- The implementation of the workflow was supported by existing and common used tools. Examples of tools were cited in the evaluation, as Pellet reasoner, SQLCO plugin, R2RML tool, Protege and UML;

At this time, the methodology only uses the algorithms previously defined in the CAR model to semi-automatize the implementation of UPCaD. One of the possible future works would be the implementation of algorithms for semi-automation of some of the steps and processes of the methodology reducing the effort of the team in the implantation.

Acknowledgments: The authors would like to thank Ricardo Pietrobon and Gustavo Costa Medeiros for the contributions about Motivational Interviewing, Federal University of Rio Grande do Sul and Federal University of Santa Maria for the support to realize this work. This research was partially supported by the authors' scholarships and individual research grants from CNPq and Fapergs (grant n.17/2551-0000875-8), Brazil.

Author Contributions: Vinícius Maran and José Palazzo M. de Oliveira conceived and designed the methodology and the experiments; Vinícius Maran, Guilherme Medeiros Machado and Alencar Machado performed the experiments and analyzed the data; Vinícius Maran Maran, Guilherme Medeiros Machado, Alencar Machado, Iara Augustin and José Palazzo M. de Oliveira wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Weiser, M. The computer for the 21st century. *Sci. Am.* **1991**, *265*, 94–104.
2. Dey, A.K.; Abowd, G.D.; Salber, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum. Comput. Interact.* **2001**, *16*, 97–166.
3. Chalmers, D. Contextual Mediation to Support Ubiquitous Computing. Ph.D. Thesis, University of London, London, UK, 2002.
4. Bettini, C.; Brdiczka, O.; Henriksen, K.; Indulska, J.; Nicklas, D.; Ranganathan, A.; Riboni, D. A survey of context modelling and reasoning techniques. *Pervasive Mob. Comput.* **2010**, *6*, 161–180.
5. Makris, P.; Skoutas, D.N.; Skianis, C. A survey on context-aware mobile and wireless networking: On networking and computing environments' integration. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 362–386.
6. Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Context aware computing for the internet of things: A survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 414–454.
7. Zhang, X.; Hou, X.; Chen, X.; Zhuang, T. Ontology-based semantic retrieval for engineering domain knowledge. *Neurocomputing* **2013**, *116*, 382–391.
8. Lee, M.H.; Rho, S.; Choi, E.I. Ontology based user query interpretation for semantic multimedia contents retrieval. *Multimed. Tools Appl.* **2014**, *73*, 901–915.
9. Samwald, M.; Freimuth, R.; Luciano, J.S.; Lin, S.; Powers, R.L.; Marshall, M.S.; Adlassnig, K.P.; Dumontier, M.; Boyce, R.D. An RDF/OWL knowledge base for query answering and decision support in clinical pharmacogenetics. *Stud. Health Technol. Inf.* **2013**, *192*, 539.
10. Forte, M.; de Souza, W.L.; do Prado, A.F. Using ontologies and Web services for content adaptation in Ubiquitous Computing. *J. Syst. Softw.* **2008**, *81*, 368–381.
11. Bolchini, C.; Quintarelli, E.; Tanca, L. CARVE: Context-aware automatic view definition over relational databases. *Inf. Syst.* **2013**, *38*, 45–67.
12. Adomavicius, G.; Tuzhilin, A. Context-aware recommender systems. In *Recommender Systems Handbook*; Springer: New York, NY, USA, 2011; pp. 217–253.
13. Maran, V.; Machado, A.; Machado, G.M.; Augustin, I.; Lima, J.C.L.; de Oliveira, J.P.M. Database Ontology-Supported Query for Ubiquitous Environments. In Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web, Gramado, Brazil, 17–20 October 2017; ACM DL: New York, NY, USA, 2017.
14. Jacobson, I.; Booch, G.; Rumbaugh, J.; Rumbaugh, J.; Booch, G. *The Unified Software Development Process*; Addison-Wesley: Reading, MA, USA, 1999; Volume 1.

15. De Nicola, A.; Missikoff, M.; Navigli, R. A software engineering approach to ontology building. *Inf. Syst.* **2009**, *34*, 258–275.
16. Maran, V.; de Oliveira, J.P.M.; Pietrobon, R.; Augustin, I. Ontology Network Definition for Motivational Interviewing Learning Driven by Semantic Context-Awareness. In Proceedings of the 2015 IEEE 28th International Symposium on Computer-Based Medical Systems (CBMS), Sao Carlos, Brazil, 22–25 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 264–269.
17. Maran, V.; Machado, A.; Augustin, I.; de Oliveira, J.P.M. Semantic Integration between Context-awareness and Domain Data to Bring Personalized Queries to Legacy Relational Databases. In Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016), Rome, Italy, 25–28 April 2016; Volume 1, pp. 238–243.
18. Chen, G.; Kotz, D. *A Survey of Context-Aware Mobile Computing Research*; Technical Report TR2000-381; Department of Computer Science, Dartmouth College: Hanover, NH, USA, 2000.
19. Strang, T.; Linnhoff-Popien, C.; Frank, K. CoOL: A context ontology language to enable contextual interoperability. In *Distributed Applications and Interoperable Systems*; Springer: Berlin, Germany, 2003; Volume 2893, pp. 236–247.
20. Knappmeyer, M.; Kiani, S.L.; Reetz, E.S.; Baker, N.; Tonjes, R. Survey of context provisioning middleware. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1492–1519.
21. W3C OWL Working Group. {OWL} 2 Web Ontology Language Document Overview. Available online: <https://www.w3.org/TR/owl2-overview/> (accessed on 30 January 2018).
22. Berners-Lee, T. Linked Data, in Design Issues: Architectural and Philosophical points. Available online: <https://www.w3.org/DesignIssues/> (accessed on 30 January 2018).
23. Maran, V.; de Oliveira, J.P.M. Uma Revisão de Técnicas de Distribuição e Persistência de Informações de Contexto e Inferências de Situações em Sistemas Ubíquos. *Cad. Inform.* **2014**, *8*, 1–46.
24. Rodríguez, N.D.; Cuéllar, M.P.; Lilius, J.; Calvo-Flores, M.D. A survey on ontologies for human behavior recognition. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 43.
25. Hervás, R.; Bravo, J.; Fontecha, J. A Context Model based on Ontological Languages: A Proposal for Information Visualization. *J. UCS* **2010**, *16*, 1539–1555.
26. Hervás, R.; Bravo, J. COIVA: Context-aware and ontology-powered information visualization architecture. *Softw. Pract. Exp.* **2011**, *41*, 403–426.
27. Poveda Villalon, M.; Suárez-Figueroa, M.C.; García-Castro, R.; Gómez-Pérez, A. A Context Ontology for Mobile Environments. Available online: <http://ceur-ws.org/Vol-626/regular3.pdf> (accessed on 30 January 2018).
28. Brown, P.J.; Jones, G.J. Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. *Pers. Ubiquitous Comput.* **2001**, *5*, 253–263.
29. Orsi, G.; Tanca, L. Context modelling and context-aware querying. *Datalog Reloaded* **2011**, *1*, 225–244.
30. Colace, F.; De Santo, M.; Moscato, V.; Picariello, A.; Schreiber, F.A.; Tanca, L. *Data Management in Pervasive Systems*; Springer: Berlin, Germany, 2015.
31. Martinenghi, D.; Torlone, R. A logical approach to context-aware databases. In *Management of the Interconnected World*; Springer: Berlin, Germany, 2010; pp. 211–219.
32. Diaz, A.; Motz, R.; Rohrer, E. Making ontology relationships explicit in a ontology network. *AMW* **2011**, *1*, 749.
33. Laclavík, M. RDB2Onto: Relational database data to ontology individuals mapping. In *Proceeding of Ninth International Conference of Informatics*; Slovak Society for Applied Cybernetics and Informatics: Bratislava, Slovak, 2007. Available online: http://nazou.fiit.stuba.sk/home/files/itat_nazou_rdb2onto.pdf (accessed on 30 January 2018).
34. Cooper, S. MOOCs: Disrupting the university or business as usual? *Arena J.* **2013**, *39*, 182.
35. Dillenbourg, P.; Fox, A.; Kirchner, C.; Mitchell, J.; Wirsing, M. Massive Open Online Courses: Current state and perspectives. In Proceedings of the Dagstuhl Perspectives Workshop 14112, Dagstuhl Manifestos, 10–13 March 2014; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Wadern, Germany, 2014; Volume 4.
36. Gutiérrez-Rojas, I.; Alario-Hoyos, C.; Pérez-Sanagustín, M.; Leony, D.; Delgado-Kloos, C. Scaffolding self-learning in MOOCs. In Proceedings of the Second MOOC European Stakeholders Summit, EMOOCs, Lausanne, Switzerland, 10–12 February 2014; pp. 43–49.

37. Bueno-Delgado, M.; Pavón-Marino, P.; De-Gea-Garcia, A.; Dolon-Garcia, A. The smart university experience: An NFC-based ubiquitous environment. In Proceedings of the 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Palermo, Italy, 4–6 July 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 799–804.
38. Machado, G.M.; de Oliveira, J.P.M. Context-aware adaptive recommendation of resources for mobile users in a university campus. In Proceedings of the 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Larnaca, Cyprus, 8–10 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 427–433.
39. Cole, S.; Bogenschutz, M.; Hungerford, D. Motivational interviewing and psychiatry: Use in addiction treatment, risky drinking and routine practice. *Focus* **2011**, *9*, 42–54.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).