

## A Context Model for Business Process Adaptation Based on Ontology Reasoning

Leila Kord Toudeshki<sup>a</sup>, Mir Ali Seyyedi<sup>a,\*</sup>, Afshin Salajegheh<sup>a</sup>

<sup>a</sup>Department of computer engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran.

### ARTICLE INFO.

#### Article history:

Received: 5 April 2020

Revised: 16 December 2020

Accepted: 3 January 2021

Published Online: 24 January 2021

#### Keywords:

Ontology, Context Information, Business Process Adaptation, Context Awareness, Adaptation, Context Model

### ABSTRACT

The surrounding context information of business processes is unpredictable and dynamically changing over time. Therefore, they should dynamically adapt themselves to different changes in context information such as business rules or computational changes in available resources. For example, we may add a particular delivery service for golden customers, or provide new payment services in case of unavailability of service providers, or change a service invocation based on available bandwidth. Unlike other methods which provide a context Meta model or a shallow context taxonomy in a specific limited scope of the business domain, we focused on ontology engineering methods not only to propose a general multipurpose context ontology but also to present our proposal as an underlying ontology for other researchers to extend and customize it for their applications. In this paper, the business process adaptation knowledge is modelled in the form of concepts, relations, and axioms which comprises time, resources, performers, locations, environment, and rules to model and record whole context information of adaptation mechanism. We characterized our work in comparison with related studies to show its completeness and demonstrated it by using an online learning management system and virtual class case studies.

© Research Article, 2020 JComSec. All rights reserved.

## 1 Introduction

Business processes (BP) are defined as a series or network of value-added services, performed by their relevant roles or collaborators, to purposefully achieve a common business goal [1]. To provide adequate services for users, business processes should be aware of

their contexts and automatically adapt themselves to their changing contexts known as context-awareness.

Business processes should be equipped with the required mechanisms to adapt themselves quickly to changes in the context, particularly at runtime. When context changes, it might affect the processes and/or quality metrics. Hence business process requires to identify and codify the relevant context information such that it can be monitored and exploited to trigger adaptation requirements. Put it simply, context is the relevant subset of the entire situation of a business process that requires a business process to adapt to potential changes in the context variables [2].

Context information could be divided into Business,

\* Corresponding author.

Email addresses: [l.kordtoudeshki@gmail.com](mailto:l.kordtoudeshki@gmail.com) (L. Kord Toudeshki), [ma\\_seyyedi@azad.ac.ir](mailto:ma_seyyedi@azad.ac.ir) (M. A. Seyyedi), [a\\_salajegheh@azad.ac.ir](mailto:a_salajegheh@azad.ac.ir) (A. Salajegheh)

<https://dx.doi.org/10.22108/jcs.2021.124660.1056>

ISSN: 2322-4460 © Research Article, 2020 JComSec. All rights reserved.



Computing, User, Infrastructure, and Ambient categories [3]. For instance, the business context includes stakeholders, regulations, business trends, and user context includes end-user preferences in addition to user activities. Each context category (e.g., user) consists of several context types (e.g., role) and each context type is composed of several context atoms (e.g., normal, loyal, VIP). Context atoms present the status of monitored data in a higher abstraction view [4].

Despite innovative works proposed by the business process community, there is a lack of general context information taxonomy/model, which has identified related adaptation concepts and their relationships precisely. Meanwhile, the construction of domain-specific context models is a time-consuming process and, in some cases, hard to achieve due to inaccessibility to business experts or rich corpora. Furthermore, the previous works have not provided an integration method to embed context information model into business processes. Thus, business processes are not allowed to be dynamic, flexible, and able to express a variety of behavior based on various situations and contexts.

In this paper, we considered business needs, in addition to collecting business process adaptation requirements and mechanisms to propose a general context information model. Since we applied ontology to propose our contribution, it supports extensibility and customizability according to the domain of usage. Moreover, it enables business rules interpretation and runtime reasoning using SWRL. From an adaptation point of view, an adaptation designer can change the adaptation requirements via the Jena library or other standard ontology integration interfaces.

The rest of the paper is organized as follows. The related works are presented in Section 2. Section 3 describes the proposed context ontology in detail. Section 4 describes the evaluation method using an online learning management system and virtual class case studies and provides a comparison table. Finally, the paper is concluded in Section 5.

## 2 Related Works

To develop related works, we have followed the principles and guidelines of systematic literature reviews (SLR) that are proposed by Kitchenham [5]. SLR is a method for combining the best quality scientific studies on a specific topic or research question. Nevertheless, the goal in this paper is not to develop an exhaustive SLR with all the work available in the literature, but to report systematically on the list of relevant contributions similar to our work, focusing on the context information model. We have performed a manual search with the terms “context information”,

“context-aware business process”, “context ontology”, “context-aware”, “business process adaptation” and “context model” on top-ranked journals and conferences. The terms have been applied to title, abstract, and keywords. Applying this search protocol, we found 87 papers covering the search criteria. 29 papers were discarded by title, 21 by abstract, and 15 papers were discarded after skimming, leading to a total of 22 papers that present different approaches.

In [6][7], the Apriori algorithm is used to learn from contextual information that is modelled based on a 3-layered Metamodel [8] to identify new unforeseen situations that impact business process adaptation. The Metamodel includes context, process, and domain layers from top to bottom. The authors presented a generic Metamodel which is instantiated by the domain of use, but they did not mention context elements related to business process adaptation. Moreover, the lack of a unified model for adaptation concepts and relations remain.

Saidani et. al. [9] proposed the role-driven business process modelling (RBPM) and CM4BPM, which cover a wide range of context-related knowledge (CRK) to adapt the behavior of business processes. They used a first-order predicate calculus to formalize the atomic context model and operations such as conjunction, disjunction, negation, and implications of contexts to formalize the complex context model. CRK is produced based on a context tree. This tree is identified by domain experts in a complex time-consuming process.

In [10][2], a context-awareness framework is proposed that includes immediate, internal, external, and environmental contexts. The immediate context is mostly related to core business process concepts including data, organizational constructs, and resources. This work is noticeable due to considering context stimuli for adaptation and comparing different BP modelling languages and notations.

In [11], the authors studied context modeling approaches including key-value pairs, mark-up schema (hierarchical structure) models, object-oriented models (graphical UML models), and ontology-based models. They claimed that ontologies are best of all because they support interoperability, distributed collaboration, sufficient reasoning, and provide sharing structures. Also, they identified evaluation criteria to compare and analyze context models including information (what to observe), structure (how to represent and formulate), capture (how to get), maintenance (how to update), reasoning (whether it is interpretable or not) and action (application of context).

To address the context in BPM modelling, adaptation, and evolution phases, Santoro et. al. [12] de-



scribed a context-aware BPM lifecycle. They proposed specific concepts, structures, and rules for abstract modelling of context-aware BP. And besides, they provided solutions for monitoring and reasoning unexpected runtime situations.

Focusing on separation of concern, Song et. al. [13] proposed a context-aware BPM in which decision model and notation (DMN) is used for modelling decision requirements and decision logic. They also extended BPMN by adaptation options and variants to change the workflow at runtime, in response to decisions.

In [14], the authors extended the ontology Meta-model to consider the dependent context of each concept. They realized their idea by defining contexts as situations in which a situation is defined as a new relation between concepts. This work is in the early stages and further descriptions and demonstrations are required to align BPM adaptation requirements.

To solve the ambiguity of context definition and context-awareness for business processes, Song et. al. [15] presented a pyramid model for different definitions of context information. Based on that model, a top-level ontological model, consisting of concepts and relationships is presented for context-aware business processes. Likewise, a three-step model is presented which considers the need for flexibility, need for personalized services, need for knowledge-intensive tasks in a BP, and injects context-dependent decision activities, and points for adaptation purposes.

SOUPA [16] and CONON [17] ontologies are proposed for context modelling in pervasive environments and smart home environments respectively. SOCAM [18] middleware has adopted CONON OWL-DL ontological model for context-awareness. In [19], a context-aware ontology is prototyped to generate an SLA document and dynamically adapt it to unexpected situations that a cloud service may encounter. They merged consumer and provider ontologies in SLA ontological model and used SWRL- inference techniques to automatically re-negotiate the SLA.

### 3 Proposed Context Ontology

In this section, we describe the construction process of our proposed context ontology and how to use it for adaptation purposes.

#### 3.1 Motivation

To bring adaptability to business processes, an adaptation unit needs to be considered which controls and manages the current static structure of processes

(e.g., BPMN). The adaptation unit gets raw monitored events and interprets whether an adaptation requirement is triggered or not. In former cases, an adaptation strategy is realized to modify the business process structure or substitute services within it. The adaptation strategies might be embedded in business process structure at design time by pre-defined structures (for example extending BPMN with adaptive notations), or be realized during runtime. In both cases, it is recommended to separate the adaptation unit from running BP instances to achieve more flexibility in changing business rules and BP workflow. Similarly, several heterogeneous raw data are coming from different resources (e.g., computational, infrastructure, etc.), performers (e.g., users, systems, organizations, etc.), and locations that need to be understood and interpreted. For better analysis, semantic identification of monitored data is essential which assigns multiple ambiguous forms of data to unified standard concepts and relations. On the other hand, decision-making is a crucial activity in BP adaptation that is done based on rules, relations, and axioms, and on top of that, we are facing distributed business process execution during the age of the Internet of Things (IoT). Hence share ability of any proposed contextual model is a must. RDF and OWL are available tools to define declarative representation and to publish and share ontologies [20].

Understanding, analyzing, and modelling the data of sensors that are attached to objects in IoT, is a challenging issue. In [21], a comprehensive survey is proposed which leads to a context-aware IoT framework. It includes practical Taxonomy based on identified features, models, techniques, functionalities, and approaches in past related studies, Context Acquisition, Context Modelling, Context Reasoning, and Context Distribution. The need for a generic context ontology is emphasized in reference [22]. They presented a three-level context ontology with the goal of reusability, extensibility, and adaptation. The first two levels are modelled in general and specific context information related to concepts like a resource, agent, time, environment, etc. The third level is used for domain specific ontologies like home, restaurant, vehicle etc. All in all, we applied ontology to cover all context modelling requirements as mentioned above. Last but not least, ontologies are easy to change via standard libraries namely Jena to modify rules, relations and runtime ontology population (instantiation).

#### 3.2 Ontology for Context Modelling

Ontologies are mostly applicable in pervasive computing. Ontologies represent a description of the concepts and relationships. Therefore, ontologies are a very promising instrument for modeling contextual infor-



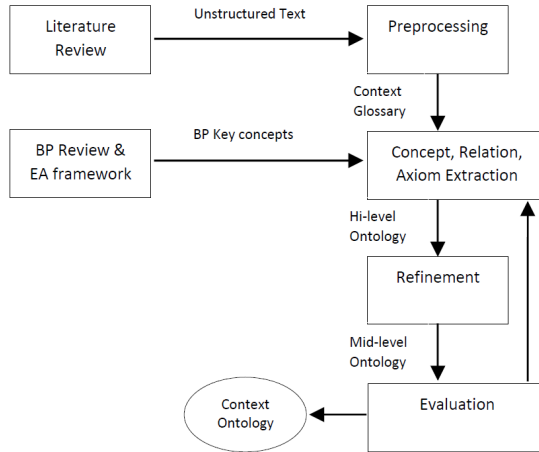


Figure 1. Ontology Construction Process.

mation due to their high and formal expressiveness and the possibility for applying ontology reasoning techniques [20]. In [23] a context ontology is presented to capture changing ambient and user contextual information about each service. They extended Web Service Modelling Ontology (WSMO) to attach adaptation rules and adaptation actions. In another work [18], an upper-level context ontology is presented which can be related to a lower-level ontology to cover all domain-specific concepts. They also provided quality ontology and reasoning to prevent context conflicts. Ontology Learning techniques are used in [24] to make patterns based on the history of users' activities and their contextual conditions. Then these patterns are added to context ontology to adapt themselves properly to different contextual conditions and predict new situations. COCCC [25] is a context ontology, specifically for Android mobile applications. It is constructed based on the following concepts for standardization purposes: Android Application, Activity (the functionality of the application that manipulates context data), Mobile device (portable devices that run the application), Context (different types of context information), and Application manifest (system data).

We applied the ontology model to cover “reusability and share ability”, extensibility and adaptability. Ontology models can be easily reused and shared using the URI of the OWL model by any business parties. Also, there is no limitation for extending ontology with domain-specific entities or modifying the model based on software design concerns. Finally, ontology models are the best way to maintain contextual information in adaptive software systems or adaptive BPMS since they support rules and reasoning between concepts and relations.

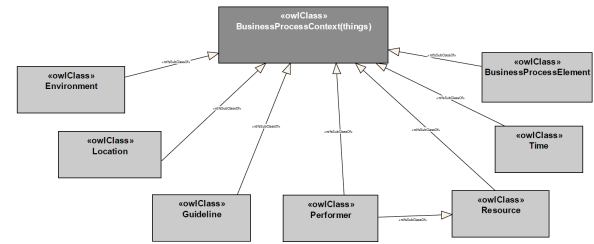


Figure 2. Highest Abstraction Level of Proposed Context Ontology.

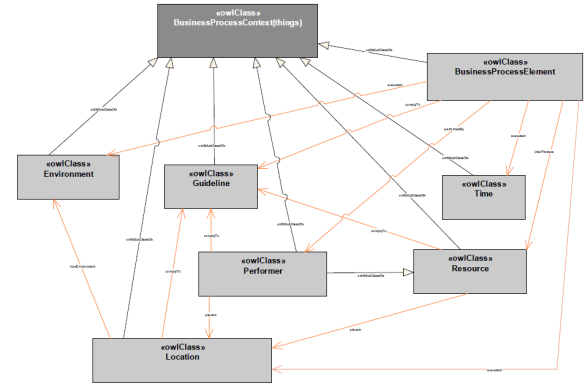
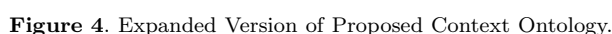


Figure 3. Context Ontology With Object Properties.

### 3.3 Construction

Ontology learning (construction) methodology includes following activities: Unstructured text preprocessing, Terms/concept extraction, Relation extraction, Axiom definition and Evaluation [26]. As depicted in Figure 1, we applied this methodology to propose our context ontology. First of all, in the literature review step, we collected and studied related papers and extracted text snippets that contained context models, context taxonomy, or context information. Then, in the pre-processing step, we extracted context entities, context attributes, and context values in text snippets and provided a context glossary. Next, since our goal was to construct a context ontology for BP adaptation, after reviewing enterprise business processes and Enterprise Architecture (EA) frameworks, we considered five fundamental questions: Who, What, Why, How, When and Where, to define business process key concepts [9]. As a result of our reviews, both context glossary and BP needs are used in the extraction step to identify Concepts, Relations, and Axioms. The output of this step is a high-level ontology in response to five mentioned questions for business process context information. Next, in the refinement step, we extended high-level ontology with context attributes, context values (instances), relations, and annotations. Finally, a repetitive process of evaluation was done to produce a clear generic Context Ontology which covers most





### 3.4 Proposed Context Ontology

The relationships among extracted concepts are depicted in Figure 3. Object property is described in the following sections.

```
classDiagram
    class BusinessProcessContext {
        +things
    }
    class Environment
    class Pollution
    class Noise
    class Humidity
    class Temperature
    class Pressure
    class Lighting
    class MeasurementUnit

    BusinessProcessContext <|-- Environment
    Environment <|-- Pollution
    Environment <|-- Noise
    Environment <|-- Humidity
    Environment <|-- Temperature
    Environment <|-- Pressure
    Environment <|-- Lighting
    Environment --> MeasurementUnit : hasMeasurementUnit
```

**Figure 5.** Environment Concept.



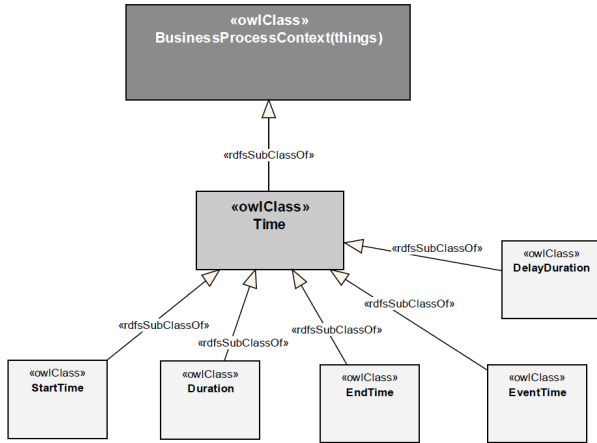


Figure 6. Time Concept.

ships is depicted in Figure 4. The expanded version contains 138 concepts and 90 object properties.

**Environment:** The surrounding environment of each business process is changing unpredictably and unexpectedly. As depicted in Figure 5, by the environment, we mean the following influencing concepts that may change BP execution: Pollution, Noise, Humidity, Temperature, Pressure, Lighting, and Weather Condition. The Measurement Unit concept is considered to define how to measure mentioned concepts, for example, the temperature is measured in Centigrade or Fahrenheit.

**Time:** It is an imperative concept in business process elements. As depicted in Figure 6, the Time concept identifies when a BP element is started/ended (Start Time/End Time) in addition to the duration of its execution (Duration) and duration delay (Delay Duration). Also, business process elements are influenced by the time of an incoming event (Event Time), whether BP should react to the raised event or miss it.

**Guideline:** Business processes are modelled and performed to achieve business goals and objectives. Business processes should be aligned with Regulation by following Rules and Standards. Furthermore, business Policies comprise Strategies which are realized by Tactics respectively. In Figure 7, we covered all these concepts concerning the Guideline concept.

**Resource:** Since each BP uses one or more resources or produce them, the Resource concept is one of the most important context information in BP execution. As depicted in Figure 8, the Resource concept includes Information, Energy, Material, Performer, and Equipment concepts that are described in the following text.

*Information* is extended by domain-specific concepts and relations. This is done by domain experts since each domain has its specific requirements and

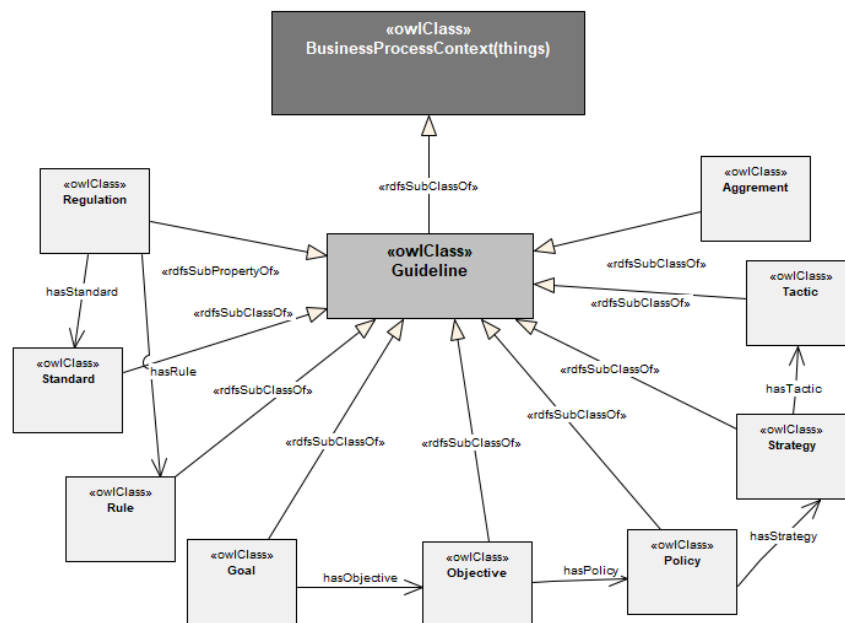
definitions. For example, flight information in an aviation system or merchandise information in a shopping system are subclasses of the Information concept.

*Equipment* is a subclass of Resource concept and includes Network, Process node, Network node, Machine, Storage subclasses. All these concepts could be extended by domain experts based on BP adaptation requirements and are described as below:

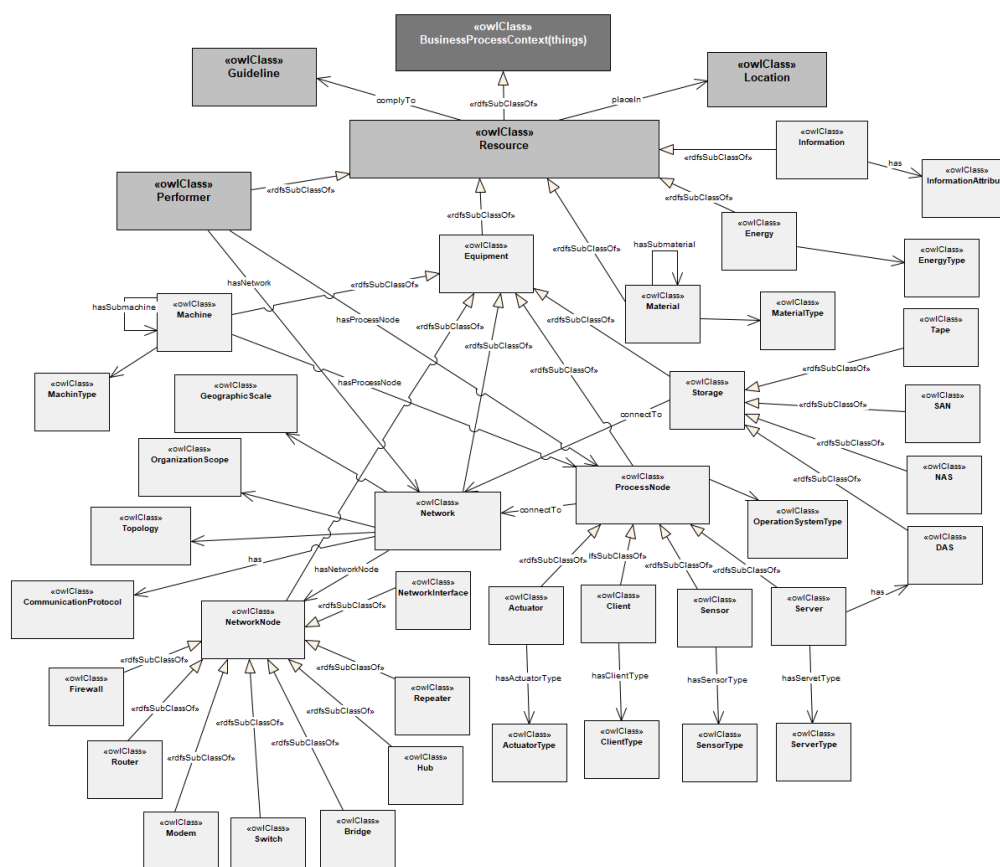
Each resource can place in a Location that is identified by placeIn object property. Also, resources follow guidelines, policies, and rules that are defined by complyTo object property.

- *Network* is used to share resources and information among several interconnected devices. The Network is an infrastructure for interconnecting devices and transmitting information among them at the lowest level of BP execution. It is important since network failure may interrupt BP execution. Consequently, context information monitoring at the network level is critical to adapt BPs in case of failures or needs for quality optimization. The Network concept includes several data type properties such as bandwidth utilization, packet loss, net flow, jitter, and the number of errors and discards. GeographicScaleType includes categories such as LAN, WAN, MAN, Internet, etc. that is defined by an object property. Also, we considered OrganizationScope like the Internet, Extranet and CommunicationProtocol like TCP/IP, UDP, HTTP and different network topologies such as Line, Ring, Mesh, Star, Tree, Bus, etc. that are related to Network concept by "has" object properties.
- *Storage* is used to identify all context information related to HDD or tape-based data storage. Domain experts can extend the storage concept by defining datatype properties such as storage type, storage capacity, used storage, read/write speed, etc. Since storage can connect to a network, an object property is considered to define assigned IP and port. Storage subclasses are Tape, DAS, NAS, and SAN.
- *ProcessNode* refers to any processing device that includes data type properties such as node ID, operating system type, model, CPU utilization, memory utilization, mobility, availability, etc. Each processing node could be connected to a network node via connectTo object property with assigned IP and port. Server, Sensor, Client, and Actuator are subclasses of the processing node. The Client includes all customer-side devices, for example, mobile phones, PC, etc. that can be defined by the ClientType concept and be related to the Client concept as an object property. Also, the





**Figure 7. Guideline Concept.**



**Figure 8.** Resource Concept.

client could include several data type properties such as web browser types, local time, locations, etc. The Server concept refers to several types of database servers, application servers, monitoring,

and control servers. `ServerType` refers to different types of service providers and is related to `Server` as an object property. Likewise, service providers could include several data type properties such

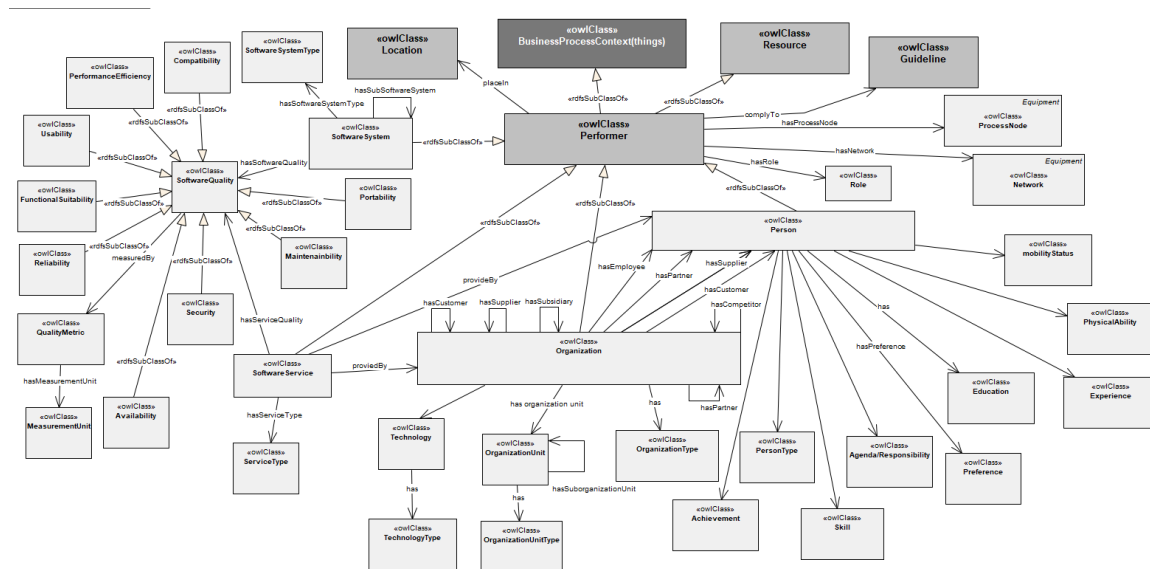


Figure 9. Performer Concept.

as available memory, number of CPU cores, and number of HDDs. If a server has DAS storage, it is addressed by using the “has” object property with the DAS concept. The Sensor concept is considered to model all types of captured data from sensors such as temperature, voltage, LDR, humidity, mobility, etc. The SensorType is used to identify the type of sensors and is related to the Sensor concept by object property. Besides, a sensor includes different data type properties such as precision, accuracy, etc. The Actuator concept is considered to model context information related to electrical, pneumatic, hydraulic, and all other types of actuators. ActuatorType is considered to define the type of actuator and is related to it by an object property. The actuator is composed of different data type properties related to the actuator concept. Moreover, ClientType, ServerType, SensorType, and ActuatorType could be considered as a subclass of its corresponding parent concept, based on adaptation requirements and reviews of domain experts.

- *NetworkNode* is another subclass of the Equipment concept. Network nodes are interconnected using hardware elements such as NetworkInterfaces, Repeaters, Hubs, Bridges, Switch, Routers, Modems, and Firewalls that are considered subclasses of a network node.
- To model context information related to all devices and machines that are applied to perform business processes, we considered the Machine concept. It includes crane, lifting, weighbridge, medical and industrial devices. MachineType is used to identify the type of machine and is related to the Machine concept by object property.

Also, a machine includes different data type properties such as speed, precision, scale, status of the machine. Meanwhile, each machine may include several machines and process data or perform operations automatically defined using object properties.

- *Material* is a subclass of resource concept. Context information of materials such as coal, oil, oxygen, etc. that are used/produced during the execution of BP, are identified and modelled in the Material concept. MaterialType is used to identify the type of material and is related to the Material concept by object property. This includes different data type properties such as material type, color, weight, temperature, density, and state. Meanwhile, each material may include several materials that are defined using object properties.
- *Energy* as a subclass of resource concept includes context information of energies such as electricity, water, sun, wind, etc. that are used/produced during the execution of BP. They are identified and modelled in the Energy concept. EnergyType is used to identify the type of energy and is related to the Energy concept by object property. This consists of different data type properties such as consumed energy, produced energy, energy capacity, etc.

**Performer:** As depicted in Figure 9, a Performer includes any individual (Person/Organization), SoftwareService, or the SoftwareSystem. Since performers are the agents of accomplishing BP activities, its related context information is important for BP adaptation. For example, consider a loan investigation task in which the task is delayed due to the absence of its





responsible staff for more than a couple of days, in this case, the task is delegated to another person with the same authorities and access level to be handled.

Each performer has a location that is related to the Location concept by the *placeIn* object property. Performing agents comply with policies, rules, and regulations by their relations to the Guideline concept via the *complyTo* object property. Since a performer can play several roles during BP execution, we considered its relationship with the Role concept via *hasRole* object property.

Person, Organization, SoftwareService, and SoftwareSystem are subclasses of the Performer concept. The Person includes all context information surrounding an individual which plays a role in BP execution. We identified context information for Person as follows: Mobility (running, jogging, stationary, etc.), Physical ability (healthy, deaf, blind, etc.), Experience (junior, senior, professional, etc.), Education (bachelor's degree, master's degree, PhD., etc.), Agenda/Responsibility, Preference (quality needs, wishes, etc.), Skill, Achievement (awarded, talented, stand out, etc.) and PersonType (aging, managerial levels, etc.).

*SoftwareSystem* concept refers to any software system such as educational, sales, inventory, maintenance, etc. which plays a role in performing a BP. In some cases, a SoftwareSystem comprises other software systems. For example, if a software system is not available, another similar one (duplicate) can handle its responsibilities.

*SoftwareService* concept refers to service providers' web services that are consumed during BP execution. Web services are provided by individuals or organizations and have different types like sale services or payment services. Service providers in either case of SoftwareSystems or SoftwareServices follow SLA for quality assurance. The *hasSoftwareQuality* object property is considered for making relations between software qualities and software systems. The *hasServiceQuality* object property is considered for making a relation between software qualities and software services. Quality attributes such as FunctionalSuitability, Security, Availability, Maintainability, Portability, Compatibility, Usability, Reliability, PerformanceEfficiency are monitored and in case of any SLA violation, BP adaptation is realized. The SoftwareQuality concept similar to performance is measured by the QualityMetric concept (like response time) via the *measuredBy* object property.

*Organization* is another performer subclass which refers to companies, organizations, enterprises, foundations, etc. OrganizationUnit concept shows the en-

closing departments or units of an organization such as financial, procurement, sale, etc. To consider applied technologies of each organization, the Technology concept is related to the Organization concept via an object property.

Each of these performer subclasses has different types. Therefore, SoftwareSystemType, ServiceType, and OrganizationType are considered as object properties to show the relationship with its corresponding concepts.

Each of these performer subclasses have datatype properties. SoftwareSystem data type properties include software availability, platform, programming language, etc. SoftwareService data type properties include service binding protocol, service publication, and discovery, service description standard, etc. Organization data type properties contain years of organization activities, number of staff, capital, organization scale, etc. For example, organizations may have partners, suppliers, customers, competitors, employees, and comprise subsidiary parts that are related to OrganizationType (educational, trading, industrial, etc.). OrganizationUnit datatype properties include the number of employees, unit average cost, unit average income/profit, etc.

**Location:** Business processes perform in a specific location. It is essential to know the location of performing elements and resources, in addition to the changes in their location.

As depicted in Figure 10, location has environmental conditions, that is defined by the *executeIn* object property in relation to the Environment concept. Rules and regulations of each location are defined by its relation with Guideline via the *complyTo* object property. Moreover, other properties like latitude and longitude are considered as data type properties.

There are the following subclasses for Location concept: Point (e.g. the precise geo point of a device), Line (includes multiple points), Surface (includes points and lines), GeoFeature (is related to GeoFeatureType such as a mountain, lake, ocean, etc.), RegionOfWorld (e.g. continents), Country (countries of continents), Region (is related to RegionType to show districts, zones, etc.), RealProperty (is related to RealPropertyType to show ground, apartment, real estate, etc.). The RealProperty has the RealPropertyUnit which is related to RealPropertyUnitType to show different parts of a property such as floors, basements, parking areas, corridors, rooms, etc. Each of the described concepts can be extended to include data type properties.

**BusinessProcessElement:** Business processes consist of several process elements that are interconnected to be performed by BP engines. Since process



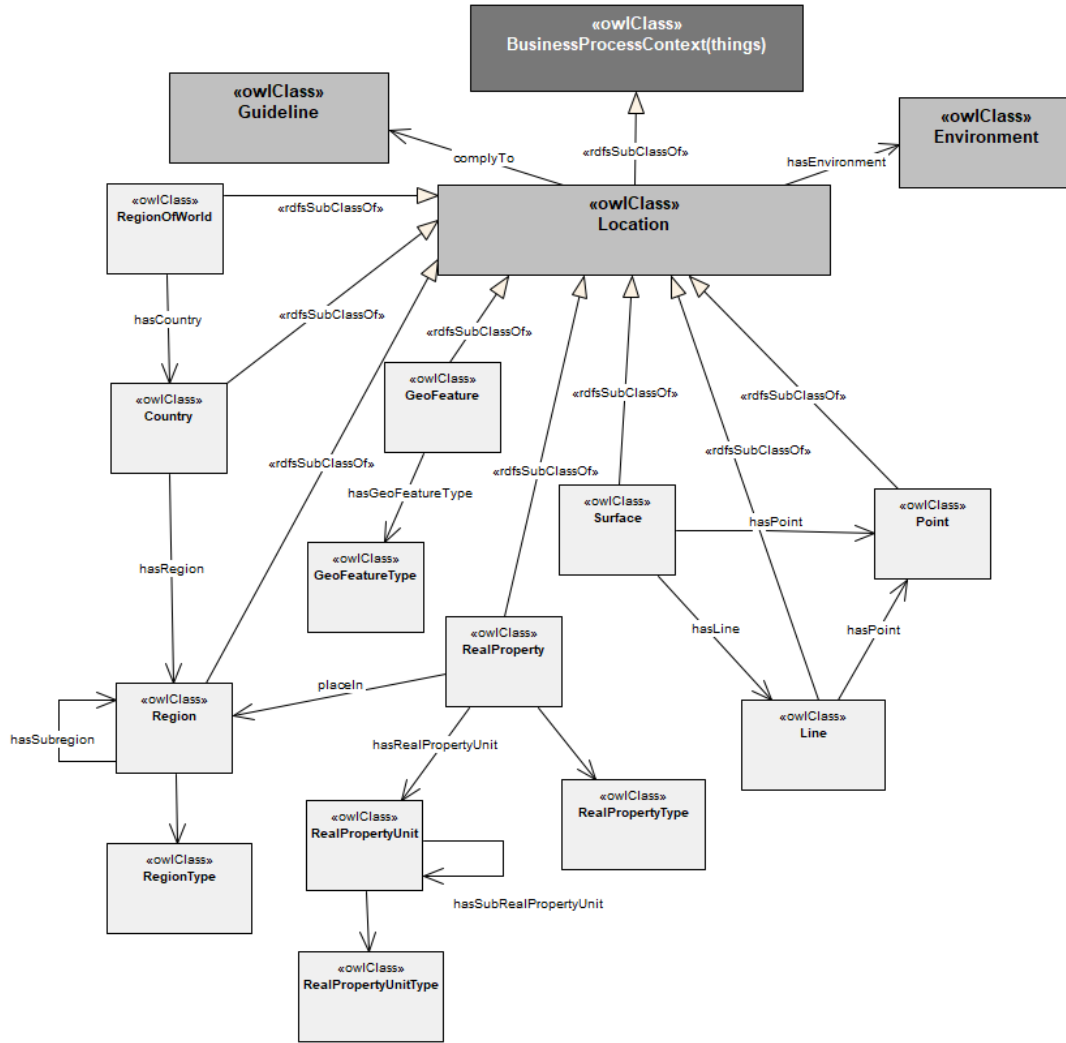


Figure 10. Location Concept.

elements may be affected by different types of changes, as depicted in Figure 11, we considered them as context information with their appropriate relations with other contextual concepts in our proposed context ontology.

Although there are many types of BP notations, BPMN is a prominent notation that covers mostly all BP structures such as Activity, Event, Gateway, Swimlane, Data, and FlowObject. Hence we identified BPElement concepts based on BPMN standard.

As an example, if a junior sales specialist does not handle a task within 24 hours (permitted hours to handle a task), the task will be moved to his/her supervisor with 5-year experience in risk investigation and handling. In this case, a relation between BP element and Time for defining the permitted period of doing a task and a relation between BP element and Person for defining the responsible supervisor with 5-year' experience and risk managing skill is established.

Moreover, a BP element is related to the organization or precisely to the organization unit, which performs BP. An organization unit specifies a swimlane in which the adaptation unit can switch the task from one swimlane to another.

### 3.5 Context Ontology Manipulation

In this section, we will describe how to integrate the proposed context ontology with an application and how to change the ontology at runtime. In [22], a context-aware architecture is presented to manage and exploit context ontology based on SWRL rules, queries, and data schemas. The QoS monitoring sensors and GPS get raw data provided by current context data. On the other hand, users/agents define what to collect and evaluate the ontology using competence questions. SWRL rules originate from business requirements, as a result, domain experts can easily define and modify them with the help of In this section, we



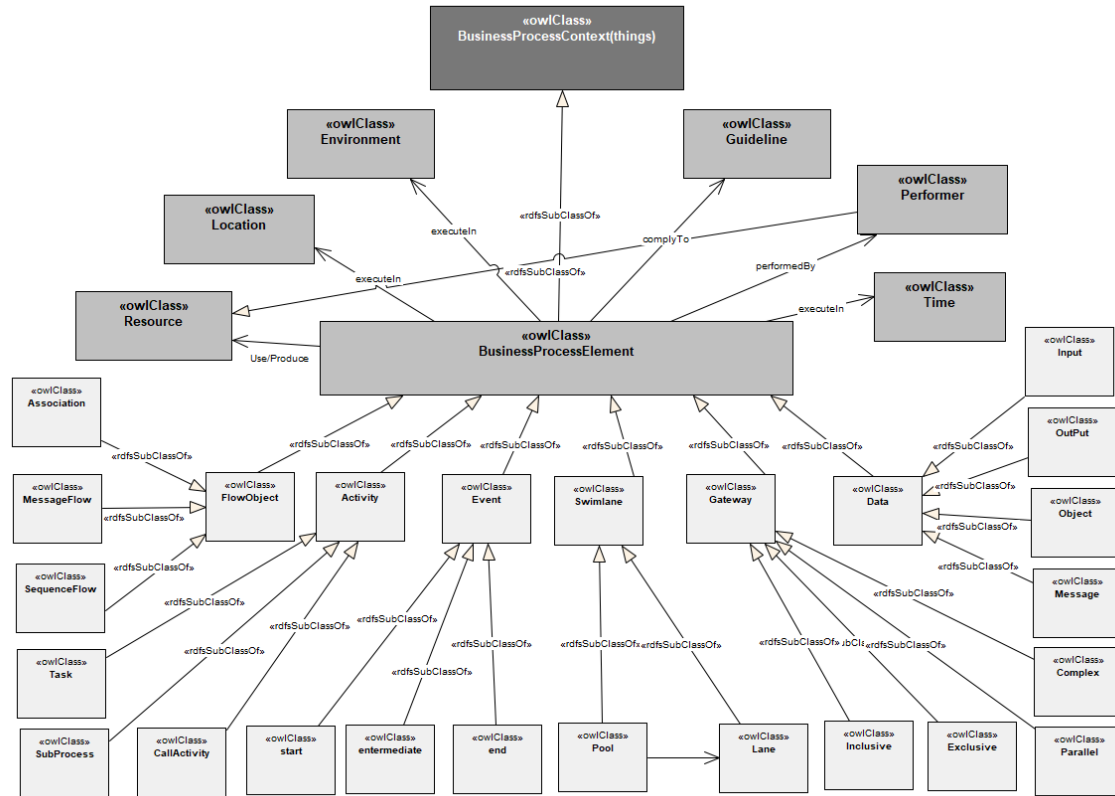


Figure 11. Business Process Element Concept.

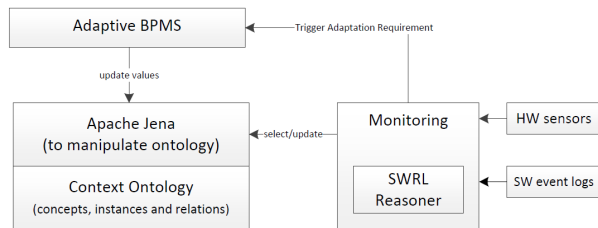


Figure 12. Context Ontology Manipulation.

will describe how to integrate the proposed context ontology with an application and how to change the ontology at runtime. In [22], a context-aware architecture is presented to manage and exploit context ontology based on SWRL rules, queries, and data schemas. The QoS monitoring sensors and GPS get raw data provided by current context data. On the other hand, users/agents define what to collect and evaluate the ontology using competence questions. SWRL rules originate from business requirements, as a result, domain experts can easily define and modify them with the help of suitable tools. Suppose an online learning management system in which new students are added to classes at beginning of a term (a new instantiation is required to context ontology) and graduate by passing all courses (a change in student education state is required in context ontology). Additionally, their

mobile phone bandwidth might change during online class attendance (a new change in network context information has occurred). To face all these changes that are not limited to what we mentioned, the context ontology needs to be modified easily and quickly at runtime. Since our work business processes (static structure part) are separated from context ontology (knowledge-based reasoning part), a mechanism is required to modify context ontology. This will trigger some adaptation requirements and realizes adaptation strategies respectively. In our example, to respond to lack of sufficient bandwidth, video streaming class changes to audio class to inhibit class interruption. As depicted in Figure 12, we used Apache Jena API [27] tool to manipulate context ontology. Jena is a programming toolkit, using the Java programming language and support OWL or RDFS ontology languages. The input values for Apache Jena come from Monitoring and Adaptive BPMS units. The monitoring unit gets raw data from hardware sensors or software event logs and makes proper updates on context ontology. The Adaptive BPMS unit modifies context ontology with running business processes. In our example, a student education state changes to graduate, when “passing all courses” event is arrived.

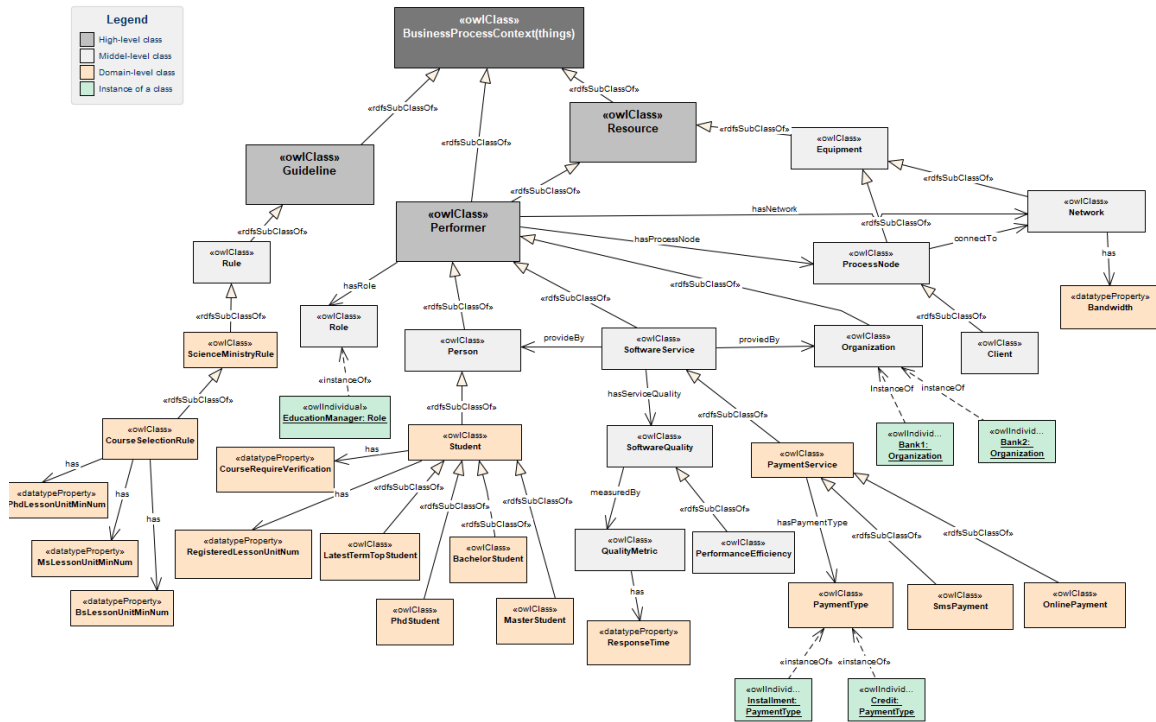


Figure 13. LMS Context Ontology.

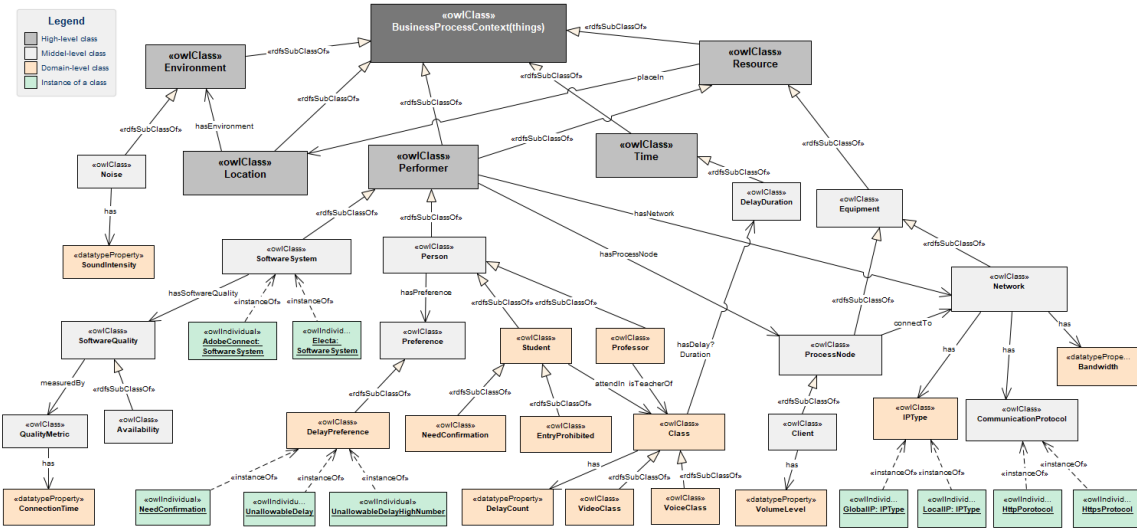


Figure 14. Virtual Class Context Ontology.

## 4 Evaluation

In this section, an online learning management system and virtual class case studies are applied to demonstrate our proposed ontology. Then, we characterized our work in comparison with related studies to show its completeness.

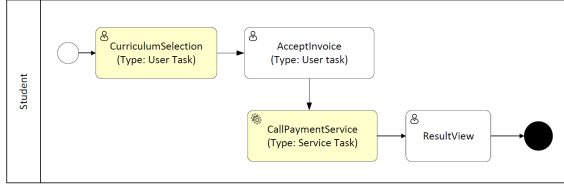
### 4.1 Case Study

To better understand our proposed ontology, we applied the business processes of the online learning

management system (LMS) and virtual class (VC) as case studies. We used 9 scenarios to describe how to extend the proposed generic ontology to cover adaptation requirements and context information related to our case studies.

As depicted in Figure 13 and Figure 14, high-level classes are colored in dark grey and middle-level classes are colored in light grey based on the proposed generic ontology. Now, it is time for extending the ontology to cover domain-specific concepts and relations for LMS





**Figure 15.** Curriculum Selection and Payment Process.

and VC case studies. The extended domain-specific concepts, data type properties, and object properties are colored in orange, and instances (individuals) are colored in green, which are described in the following scenarios.

While adaptive BPMS is performing the LMS and VC business processes, the monitored data and performing BP results are saved/updated in an ontology using manipulation mechanism (refer to Section 3.4), and SWRL semantic reasoner does decision making based on updated data.

#### 4.1.1 Curriculum Selection and Payment Process

As represented in Figure 15, this process is supposed to be performed via a mobile phone application of undergraduate students. First, a student selects available courses for the current term which should be above the minimum required number of courses defined in Ministry of Science rules. Then, he/she confirms the invoice and proceeds to check out for payment which is available in credit via online payment gateways. Finally, the process is terminated by showing the payment results.

In Figure 13, we extended the proposed ontology to support required contexts in described scenarios, and in the following paragraphs, we will describe how to use SWRL for reasoning on this ontology.

**First Scenario:** If a student wants to submit a new term with the number of lessons less than permitted, as per a rule which defines the minimum number of required lessons per term, confirmation of the educational manager is required. This adaptation requirement is applied to the CurriculumSelection task.

The Student is a type of BP performer and is considered as a subclass of the Person concept. For the Student concept, we added RegisteredLessonUnitNum and CourseRequiredVerification data type properties. RegisteredLessonUnitNum is an integer data field to save the number of registered lessons and CourseRequiredVerification is a boolean data field that refers to whether educational manager confirmation is required or not. Also, the Ministry of Science rules is a subclass of rules that contain rules related to term

**Table 1.** Extended Ontology Elements for First Scenario.

<b>Concept</b>	1-Student: Person
	2-ScienceMinistryRule: Rule
	3-CourseSelectionRule: ScienceMinistryRule
<b>Data Type Property</b>	1-RegisteredLessonUnitNum Domain: Student , Range: int
	2-BsLessonUnitMinNum Domain: CourseSelectionRule , Range: int
	3-MsLessonUnitMinNum Domain: CourseSelectionRule , Range: int
	4-PhdLessonUnitMinNum Domain: CourseSelectionRule , Range: int
	5-CourseRequiredVerification Domain: Student , Range: boolean
<b>Individual</b>	1-EducationManager: Role

**Table 2.** Extended Ontology Elements for Second Scenario.

<b>Concept</b>	1-Student: Person
	2-LatestTermTopStudent: Student
	3-PaymentService: SoftwareService
	4-PaymentType
<b>Object Property</b>	1-hasPaymentType, Domain: PaymentService, Domain: PaymentService
<b>Individual</b>	1-Credit: PaymentType
	2-Instalment: PaymentType

registration. Additionally, three data type properties of integer type are considered to define minimum required courses per term for undergraduate, graduate, and Ph.D. students. Furthermore, the educational manager is an instance of a role. A summary of what we described is shown in Table 1.

Based on the proposed ontology, the SWRL of the first scenario is as follows:

$$\begin{aligned}
 & Bachelorstudent(?bs) \wedge \\
 & \quad Student(?s) \wedge \\
 & \quad subclassOf(?bs, ?s) \wedge \\
 & RegisteredLessonUnitNum(?rlun) \wedge \\
 & \quad Person(?p) \wedge Performer(?per) \wedge \\
 & \quad subclassOf(?p, ?per) \wedge Role(?r) \wedge \\
 & \quad has(?per, ?r) \wedge \\
 & swrl : equal(?r, "EducationManager") \wedge \\
 & BsLessonUnitMinNum(?blumn) \wedge \\
 & \quad swrl : lessThan(?rlun, ?blumn) \Rightarrow \\
 & \quad CourseRequireVerification(?s, true)
 \end{aligned}$$

**Second Scenario:** If a student stands out as a top student in the last term, he/she is granted to pay via





**Table 3.** Extended Ontology Elements for the Third Scenario.

<b>Concept</b>	1-Student: Person
	2-PaymentService: SoftwareService
	3-PaymentType
<b>Data Type Property</b>	1-ResponseTime, Domain: QualityMetric, Range: int
<b>Object Property</b>	1-hasPaymentType, Domain: PaymentService, Range: PaymentType
<b>Individual</b>	1-Bank1: Organization
	2-Bank2: Organization

**Table 4.** Extended Ontology Elements for the Fourth Scenario.

<b>Concept</b>	1-Student: Person
	2-PaymentService: SoftwareService
	3- OnlinePayment: PaymentService
	4-SmsPayment: PaymentService
<b>Data Type Property</b>	1-Bandwidth, Domain: Network, Range: int

installment instead of credit payment. This adaptation requirement is applied to the CallPaymentService task. LatestTermTopStudents are considered a subclass of Student concept which was added to the proposed ontology in the first scenario. Payment service is added as a subclass of SoftwareService concept which has hasPaymentType object property to identify Instalment or Credit payment types. A summary of what we described is depicted in Table 2.

Based on the proposed ontology, the SWRL of the second scenario is as follows:

$$\begin{aligned}
 &Student(?s) \wedge \\
 &LatestTermTopStudent(?s) \wedge \\
 &PaymentService(?ps) \wedge \\
 &PaymentType(?pt) \wedge \\
 &hasPaymentType(?ps, ?pt) \Rightarrow \\
 &Installment(?pt)
 \end{aligned}$$

**Third Scenario:** Suppose performance quality attribute is measured by the response time metric. If the response time of a credit payment service exceeded 60 seconds, another payment provider is substituted. This adaptation requirement is applied to the CallPaymentService task.

Student, SoftwareService, PaymentType concepts, and Instalment/Credit instances are used the same as the second scenario. The ResponseTime is added as an integer data type property to the QualityMetric concept. According to the proposed ontology, each software service is provided by an organization. Therefore, in this scenario, Bank1 and Bank2 are instances

of an organization that provide payment services. A summary of what we described is illustrated in Table 3.

Based on the proposed ontology, the SWRL of the third scenario is as follows:

$$\begin{aligned}
 &SoftwareService(?s) \wedge \\
 &SoftwareQuality(?qa) \wedge \\
 &PerformanceEfficiency(?pe) \wedge \\
 &subclassOf(?pe, ?qa) \wedge \\
 &QualityMetric(?qm) \wedge \\
 &measuredBy(?qa, ?qm) \wedge \\
 &ResponseTime(?rt) \wedge \\
 &has(?qm, ?rt) \wedge \\
 &swrl : greaterThan(?rt, '60') \wedge \\
 &PaymentService(?ps) \wedge \\
 &Organization(?org) \wedge \\
 &providedBy(?s, ?org) \wedge \\
 &swrl : equal(?org, 'Bank1') \wedge \\
 &PaymentType(?pt) \wedge \\
 &Credit(?pt) \Rightarrow \\
 &providedBy(?s, 'Bank2')
 \end{aligned}$$

**Fourth Scenario:** If a student's internet bandwidth is less than 300 Kbps, an online credit payment service is provided instead of an SMS (USSD) payment service. This adaptation requirement is applied to the CallPaymentService task.

Student and PaymentService concepts are used here as described in previous scenarios. Also, the OnlinePayment and SmsPayment services are considered by extending payment service. In the proposed ontology, each Performer has ProcessNode and each ProcessNode is connected to a Network. Therefore, a student can be connected to a network via a mobile device (as a Client), and bandwidth is added as an integer data field to the network concept to save mobile device bandwidth.

A summary of what we described is depicted in Table 4.

Based on the proposed ontology, the SWRL of the fourth scenario is as follows:

$$\begin{aligned}
 &Student(?s) \wedge \\
 &Person(?p) \wedge \\
 &subclassOf(?s, ?p) \wedge \\
 &Performer(?per) \wedge \\
 &subclassOf(?p, ?per) \wedge
 \end{aligned}$$

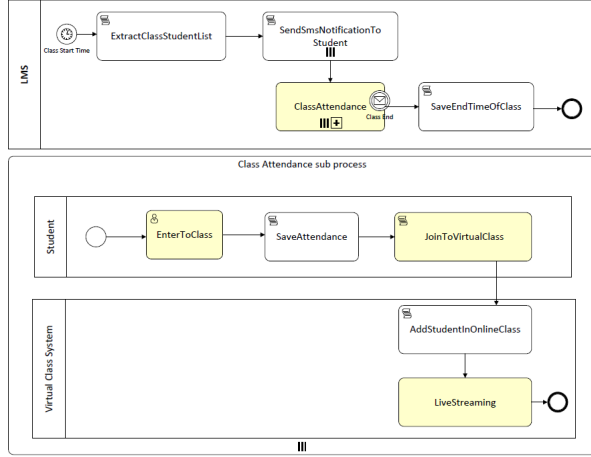



Figure 16. Class Attendance Process.

$ProcessNode(?pn) \wedge$   
 $hasProcessNode(?per, ?pn) \wedge$   
 $Client(?c) \wedge$   
 $subClassOf(?c, ?pn) \wedge$   
 $Network(?net) \wedge$   
 $connectTo(?pn, ?net) \wedge$   
 $Bandwidth(?bw) \wedge$   
 $has(?net, ?bw) \wedge$   
 $swrl : lessThan(?bw, 300) \wedge$   
 $PaymentService(?ps) \Rightarrow$   
 $SmsPayment(?ps)$

#### 4.1.2 Class Attendance Process

As shown in Figure 16, this process is performed for those students who completed their course selection and term registration. The learning management system extracts eligible class students and sends a “start of the class” SMS notification to them.

Then, students can log-in to enter the class and after their successful log-in, LMS saves the time of entrance and is connected to a virtual class system. A VC adds the student to an online class and provides video/voice live streaming facilities. Finally, when the class is terminated by the teacher, LMS sends an end of class notification to all attendances and save class termination (end) time.

In Figure 14, we extended the proposed ontology to support required contexts in described scenarios, and in the following paragraphs, we will describe how to use SWRL for reasoning on this ontology.

**Fifth Scenario:** Although students can attend a class with delays, a professor can select one of these preferences for reacting to students who have a delay in attending a class.

Table 5. Extended Ontology Elements for Fifth Scenario.

<b>Concept</b>	1-Student: Person
	2-PaymentService: SoftwareService
	2-NeedConfirmation: Student
	3-EntryProhibited: Student
	4-Professor: Person
	5-DelayPreference: Preference
<b>Data Type Property</b>	6-Class
	1-DelayPreference, Domain: Preference, Range: string
<b>Data Type Property</b>	2-DelayCount, Domain: Class, Range: int
	1-hasDelayDuration, Domain: Class, Range: DelayDuration
<b>Individual</b>	1-NeedConfirmation
	2-UnallowableDelay
	3-UnallowableDelayHighNumber

Students and the professor as two performing agents are considered two subclasses of the Person concept. Class concept is related to Student with the attendIn object property and is related to Teacher with isTeacherOf object property. For each student who attends a class, a delay time is saved which is extracted via the hasDelayDuration object property. The number of delays of each student for each class is considered as DelayCount data type property. Meanwhile, each professor may have preferences like delay preference which is added as a subclass of preferences and can be instantiated in three options:

- *NeedConfirmation* If a student has more than a 15- minute delay, he/she should be confirmed by the professor. In this case, the delayed student is labeled with NeedConfirmation via a relation between Student and NeedConfirmation instances.
- *UnallowableDelay*: If a student has more than a 15- minute delay, he/she is not permitted to join the class. In this case, the delayed student is labeled with EntryProhibited via a relation between Student and EntryProhibited instances.
- *UnallowableDelayHighNumber*: If a student has delays more than 3 times, he/she is not permitted to participate in the class. In this case, the delayed student is labeled with EntryProhibited via a relation between Student and EntryProhibited instances.

A summary of what we described is depicted in Table 5.

Based on the proposed ontology, the SWRL of the fifth scenario is as follows:

If a student has more than a 15-minute delay, he/she



should be confirmed by a professor.

**Table 6.** Extended Ontology Elements for Sixth Scenario.

<b>Concept</b>	1-Student: Person
<b>Data Type</b>	1-SoundIntensity, Domain: Noise , Range: int
<b>Property</b>	2-VolumeLevel, Domain: Client , Range: int

$Student(?s) \wedge$   
 $Professor(?p) \wedge$   
 $Class(?c) \wedge$   
 $attendIn(?s, ?c) \wedge$   
 $isTeacherOf(?p, ?c) \wedge$   
 $DelayTime(?d) \wedge$   
 $hasDelayTime(?c, ?d) \wedge$   
 $swrl : greaterThan(?d, 15) \wedge$   
 $Preference(?prf) \wedge$   
 $Person(?pr) \wedge$   
 $subClassOf(?p, ?pr) \wedge$   
 $hasPrefrence(?pr, ?prf) \wedge$   
 $DelayPreference(?dp) \wedge$   
 $subClassOf(?dp, ?prf) \wedge$   
 $swrl : equal(?dp,$   
 $"NeedConfirmation")$   
 $\Rightarrow$   
 $NeedConfirmation(?s)$

If a student has more than a 15-minute delay, he/she is not permitted to attend the class.

$Student(?s) \wedge$   
 $Professor(?p) \wedge$   
 $Class(?c) \wedge$   
 $attendIn(?s, ?c) \wedge$   
 $isTeacherOf(?p, ?c) \wedge$   
 $DelayTime(?d) \wedge$   
 $hasDelayTime(?c, ?d) \wedge$   
 $swrl : greaterThan(?d, 15) \wedge$   
 $Preference(?prf) \wedge$   
 $Person(?pr) \wedge$   
 $subClassOf(?p, ?pr) \wedge$   
 $hasPrefrence(?pr, ?prf) \wedge$   
 $DelayPreference(?dp) \wedge$   
 $subClassOf(?dp, ?prf) \wedge$   
 $swrl : equal(?dp,$   
 $"UnallowableDelay")$   
 $\Rightarrow$   
 $EntryProhibited(?s)$

If a student has delays more than 3 times, he/she is not permitted to attend the class.

$Student(?s) \wedge$   
 $Professor(?p) \wedge$   
 $Class(?c) \wedge$   
 $attendIn(?s, ?c) \wedge$   
 $isTeacherOf(?p, ?c) \wedge$   
 $DelayTime(?d) \wedge$   
 $hasDelayTime(?c, ?d) \wedge$   
 $swrl : greaterThan(?d, 15) \wedge$   
 $Person(?pr) \wedge$   
 $subClassOf(?p, ?pr) \wedge$   
 $Preference(?prf) \wedge$   
 $hasPrefrence(?pr, ?prf) \wedge$   
 $DelayPreference(?dp) \wedge$   
 $subClassOf(?dp, ?prf) \wedge$   
 $swrl : equal(?dp,$   
 $"UnallowableDelayHighNumber") \wedge$   
 $DelayCount(?dc) \wedge$   
 $has(?c, ?dc) swrl :$   
 $greaterThan(?dc, 3)$   
 $\Rightarrow$   
 $EntryProhibited(?s)$

**Sixth Scenario:** If the environment sound intensity of the student device is more than 60 decibels, increase device volume by 20 percent. This adaptation requirement is applied to the LiveStreaming task.

The Student concept which is added in the previous scenario is applied in this scenario too. In the proposed ontology, each Performer has ProcessNode and each ProcessNode is connected to a Network. Therefore, a student can be connected to a network via a mobile device (as a Client). SoundIntensity is a datatype property for the Client concept. On the other hand, ProcessNode concept is a subclass of Equipment and Equipment is a subclass of Resource. Each resource is placed in a location with environmental conditions such as Noise (as a subclass of the environment). Thus, SoundIntensity is added as a data type property for the Noise concept.

A summary of what we described is depicted in Table 6.

Based on the proposed ontology, the SWRL of the sixth scenario is as follows:

$Student(?s) \wedge$



```

    Person(?p)∧
    subClassOf(?s, ?p)∧
    Performer(?per)∧
    subClassOf(?p, ?per)∧
    ProcessNode(?pn)∧
    hasProcessNode(?per, ?pn)∧
    Client(?c)∧
    subClassOf(?c, ?pn)∧
    Equipment(?e)∧
    subClassOf(?pn, ?e)∧
    Resource(?r)∧
    subClassOf(?e, ?r)∧
    Location(?l)∧
    placeIn(?r, ?l)∧
    Environment(?en)∧
    hasEnvironment(?l, ?en)∧
    Noise(?n)∧
    subClassOf(?n, ?en)∧
    SoundIntensity(?si)∧
    has(?n, ?si)∧
    swrl : greaterThan(?si, '60') ⇒
        VolumeLevel(?vl)∧
    swrl : add(?newvl, ?vl, 20)∧
    VolumeLevel(?newvl)

```

**Seventh Scenario:** If a student is in the university intranet, then use HTTP, otherwise use HTTPS for data communication (SSL protocol).

The Student concept which is added in the previous scenario is applied in this scenario too. In the proposed ontology, each Performer has ProcessNode and each ProcessNode is connected to a Network so a student can be connected to a network via a mobile device (as a Client). Each network has connection protocol and IPType which is considered “HTTP or HTTPS” and “GlobalIP or LocalIP” in our example, subsequently.

A summary of what we described is depicted in Table 7.

Based on the proposed ontology, the SWRL of the seventh scenario is as follows:

```

    Student(?s)∧
    Person(?p)∧
    subClassOf(?s, ?p)∧
    Performer(?per)∧

```

```

    subClassOf(?p, ?per)∧
    ProcessNode(?pn)∧
    hasProcessNode(?per, ?pn)∧
    Client(?c)∧
    subClassOf(?c, ?pn)∧
    Network(?net)∧
    connectTo(?pn, ?net)∧
    CommunicationProtocolType(?cpt)∧
    has(?net, ?cpt)∧
    IPType(?ipt)∧
    has(?net, ipt)∧
    swrl : equal(?ipt, 'LocalIP') ⇒
        HttpProtocol(?cpt)

```

**Eighth Scenario:** If the bandwidth of a student device is more than 1000 kbps, then provide video streaming; otherwise, provide voice streaming. This adaptation requirement is applied to the LiveStreaming task.

Student and Class concepts are used here as described in previous scenarios. Also, VideoClass and VoiceClass are considered by extending the Class concept. In the proposed ontology, each Performer has ProcessNode and each ProcessNode is connected to a Network as a result of which a student can be connected to a network via a mobile device (as a Client) and bandwidth is added as an integer data field to the network concept to save mobile device bandwidth.

A summary of what we described is depicted in Table 8.

**Table 7.** Extended Ontology Elements for Seventh Scenario.

<b>Concept</b>	1-Student: Person 2-IPType
<b>Individual</b>	1-LocalIP: IPType 2-GlobalIP: IPType 3-HttpProtocol: CommunicationProtocol 4-HttpsProtocol: CommunicationProtocol

**Table 8.** Extended Ontology Elements for Eighth Scenario.

<b>Concept</b>	1-Student: Person 2-VoiceClass: Class 3-VideoClass: Class
<b>Data Type Property</b>	1-BandWidth, Domain: Network, Range: int

Based on the proposed ontology, the SWRL of the eighth scenario is as follows:



$Student(?s) \wedge$   
 $Person(?p) \wedge$   
 $subClassOf(?s, ?p) \wedge$   
 $Performer(?per) \wedge$   
 $subClassOf(?p, ?per) \wedge$   
 $ProcessNode(?pn) \wedge$   
 $hasProcessNode(?per, ?pn) \wedge$   
 $Client(?c) \wedge$   
 $subClassOf(?c, ?pn) \wedge$   
 $Network(?net) \wedge$   
 $connectTo(?pn, ?net) \wedge$   
 $Bandwidth(?bw) \wedge$   
 $has(?net, ?bw) \wedge$   
 $swrl : greaterThan(?bw, 1000) \wedge$   
 $Class(?c) \wedge$   
 $has(?s, ?c) \Rightarrow$   
 $VideoClass(?c)$

$Student(?s) \wedge$   
 $Person(?p) \wedge$   
 $subClassOf(?s, ?p) \wedge$   
 $Performer(?per) \wedge$   
 $subClassOf(?p, ?per) \wedge$   
 $ProcessNode(?pn) \wedge$   
 $hasProcessNode(?per, ?pn) \wedge$   
 $Client(?c) \wedge$   
 $subClassOf(?c, ?pn) \wedge$   
 $Network(?net) \wedge$   
 $connectTo(?pn, ?net) \wedge$   
 $Bandwidth(?bw) \wedge$   
 $has(?net, ?bw) \wedge$   
 $swrl : lessThan(?bw, 1000) \wedge$   
 $Class(?c) \wedge has(?s, ?c)$   
 $\Rightarrow$   
 $VoiceClass(?c)$

**Ninth Scenario:** If Adobe Connect (VC system) is not accessible, that is, it does not respond in 30 seconds, then, substitute it with Electa. This adaptation requirement is applied to the JoinToVirtualClass task. In the proposed ontology, SoftwareSystem is a subclass of Performer and both Electa and Adobe Connect are instances of the SoftwareSystem. Also, each SoftwareSystem has a relation with the SoftwareQuality. Availability is added as a subclass of SoftwareQuality and is measured by ConnectionTime as a metric.

A summary of what we described is illustrated in Table 9.

Based on the proposed ontology, the SWRL of the

**Table 9.** Extended Ontology Elements for Ninth Scenario.

<b>Data Type Property</b>	1-ConnectionTime, Domain: QualityMetric , Range: int
<b>Individual</b>	1-AdobeConnect: SoftwareSystem 2-Electa: SoftwareSystem

ninth scenario is as follows:

$SoftwareSystem(?sw) \wedge$   
 $swrl : equal(?sw, 'AdobeConnect') \wedge$   
 $SoftwareQuality(?qa) \wedge$   
 $hasSoftwareQuality(?sw, ?qa) \wedge$   
 $Availability(?a) \wedge$   
 $subClassOf(?a, ?qa) \wedge$   
 $QualityMetric(?qm) \wedge$   
 $measuredBy(?qa, ?qm) \wedge$   
 $ConnectionTime(?ct) \wedge$   
 $has(?qm, ?ct) \wedge$   
 $swrl : greaterThan(?ct, '30') \wedge$   
 $\Rightarrow$   
 $Electa(?sw)$

## 4.2 Completeness

To characterize our work in comparison with other related context models or ontologies, we defined 8 metrics as described in the following list, according to S-CUBE [28] adaptation and OMG BPM definitions:

- *Type of Model:* What is the type of proposed model including Model, Meta Model, or Ontology Model?
- *Adaptation Actor:* It refers to who is performing the adaptation strategy. Adaptation can be done by a software system in semi/full automatic (self-adaptation) or manually by a person.
- *Adaptation Subject:* It refers to what is under adaptation.
- *Adaptation Requirement:* It refers to why an adaptation is required and under what quality or functional conditions adaptation requirement is triggered.
- *Monitored Property:* It refers to all required raw/rich monitored data for adaptation reasoning or other BP execution needs.
- *Adaptation Strategy:* It refers to how to achieve an adaptation requirement.
- *Extendibility:* Is the proposed model extendible for domain-specific purposes?
- *Business Process Support:* Does the proposed model consider business process elements in any case from Activity, Task to other elements such





Table 10. Comparison Table.

Comparison Metric	[8]	[15]	[10]	[18]	[25]	[11]	[22]	[14]	[23]	Our Work
<b>Type of Model</b>	Meta Model	Ontology	Conceptual Model	Ontology	Ontology	Meta Model	Ontology	Ontology Meta Model	Ontology for WSMO	Ontology
<b>Adapt. Actor (who)</b>	Actor	NA		NA Person, Computing Entity	NA	Actor	Agent	NA	NA	Performer
<b>Adapt. Subject (what)</b>	Activity	Business Process	Core Business Process Concepts	Any Defined Subject	Mobile Devices	Task, Org., Process, Resource	Context Info., Entity	NA	Only Service	BP Element, Resource, Performer, Any Defined Subject
<b>Adapt. Req. (why)</b>	Situation	NA	NA	Quality Constraints, Parameter, Metric	NA	CRK Tree	NA	NA	Situation Adapt. Rules	SWRL, Software Quality, Quality Metric, BP Element
<b>Monitored Property</b>	Contextual Element	Contextual Entities	Context Classifications and Types	Context Entities	Context, Application Manifest	Context Element	Context Info.	Attached situations to concepts	Context Entity	Generic & Specific Onto. Entities including Location, Time, etc.
<b>Adapt. Strategy (how)</b>	If-then Rules	NA	NA	First-order logic predicate	NA	First Order Predicate Rules	NA	NA	Situation-related Adaptation	Reasoning via SWRL
<b>Extendibility</b>	NA	Yes	NA	Yes	Yes	NA	Yes	Yes	Yes	Yes
<b>Biz. Process Support</b>	Yes	Yes	Yes	NA	NA	Yes	No	NA	NA	Yes BP Element

as Events, Conditions, etc.?

As depicted in Table 10, our work covers all 8 defined metrics based on an ontology model. The Performer concept is extended to define adaptation actor as a person (human) or software system for both automatic and manual adaptation cases, respectively. We provided BP Element, Resource, and Performer concepts to support all subjects of adaptations. By applying SWRL, it is possible to define all types of adaptation requirements. Since adaptation requirements are applied to software quality attributes and software functionalities, the former is provided by SoftwareQuality and QualityMetric concepts, and the latter is provided by BP Elements in our ontology. Monitored properties are supported in multiple levels of abstractions in our work; ranging from generic proposed concepts to any considerable domain-specific concepts. There is no limitation for defining desired properties that need to be monitored in our ontology. We defined adapta-

tion actions as concepts and applied SWRL rules to call them. SWRL can be used to define what should be done if an adaptation requirement is triggered and what action is realized to react to changing situations. Any adaptation designer or domain-specific engineer can easily extend the proposed ontology for his/her purposes. “BPElement” concept can be used for making a relation between any type of monitored property (ontology concept) and business process elements such as task, event, lane, pool, etc.

As described in [29], completeness is defined as expected comprehensiveness. Data can be completed even if optional data is missing. So long as the data meets the expectations, the data is considered complete. Regarding this definition, we considered SCUBE for main business process adaptation metrics and put them together to define expectations (metrics) that need to be achieved. Table 10 shows the completeness of our proposed ontology in comparison



with other works. From a measurement point of view, as depicted in Table 10, our proposed ontology fulfills all S-CUBE expectations and can expect to have the maximum measurement value, but for providing more accurate measurement it is possible to create suitable questionnaires and ask experts to complete them.

## 5 Conclusions

In this paper, we presented an ontology for modelling, reasoning, and manipulating context information. In particular, it is generally applicable in the business process adaptation field. As described in Section 3.3, for modelling, we performed an SLR on several related papers and got the benefit of their models and ideas in our work. To meet reusability, share ability and extensibility quality attributes which are instinct needs of pervasive business process adaptation, we considered ontology context modelling. Ontology is not only extensible easily by domain experts and shareable by standard URL but also reasonable using SWRL rule engine. The latter is a compulsory need in adaptive BPMS. Context information which is monitored via sensors is changing at runtime, in this regard, we applied SWRL for manipulating and reasoning on ontology to discover new facts and make the ontology data updated. This work is prototyped with 9 scenarios in LMS and VC case studies to show how our generic ontology could be extended for specific purposes. Also, we considered business and adaptation rules for each scenario and defined them in SWRL format. Considering all together, to characterize our work compared to related ones, we defined 8 metrics based on S-CUBE adaptation and OMG BPM definitions. The result of the comparison is depicted in Table 10 which shows our generic context ontology model not only covers adaptation and business process contextual needs but also is more extendable and reusable in face of new domain requirements.

In the future, we will develop our ontology model based on new research papers to cope with new quality requirements and functionalities. Moreover, we will apply it in an adaptation framework to show its applicability. Since adaptation frameworks are designed based on the separation of concern, we mean, separation of adaptation (control) unit from the dynamic structure (mesh of services, components, etc.), our proposed ontology can be used in adaptation unit without any fundamental changes in dynamic structures. An adaptive software system (or adaptive BPMS) can use the proposed ontology to save incoming context data and reason on them. This can be useful for decision-making in face of any adaptation need and realizes proper adaptation strategies to keep the adaptive system aligned with new context information. Last but

not least, we will ask domain experts to fulfill evaluation questionnaires for measuring our proposed ontology, quantitatively.

## References

- [1] R. Ko. A computer scientist's introductory guide to business process management (BPM). *XRDS: Crossroads, The ACM Magazine for Students*, 15(4):11–18, 2009. doi:10.1145/1558897.1558901.
- [2] M. Rosemann, J. Recker, C. Flender, and P. Ansell. Understanding context-awareness in business process design. In *Proceedings of the 17th Australasian Conference on Information Systems*, pages 1–10. Australasian Association for Information Systems, 2006. ISBN 978-0-9758417-1-6. doi:http://aisel.aisnet.org/acis2006/79.
- [3] M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. *The S-cube research vision," in Service research challenges and solutions for the future internet, Springer*. Springer, 2010.
- [4] Y. Rastegari and F. Shams. A Context-Aware Reflective-State Framework to Reconfigure Service-Based Applications. *Journal of Computing and Security*, 2(4):281–292, 2015. ISSN 2322-4460.
- [5] Staffs Keele et al. Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.
- [6] J. Santo, F. Santoro, and K. Revoredo. A method to infer the need to update situations in business process adaptation. *Computers in Industry*, 71:128–143, 2015. doi:10.1016/j.compind.2015.03.014.
- [7] V. T. Nunes, F. M. Santoro, C. M. Werner, and C. G. Ralha. Real-time process adaptation: a context-aware replanning approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(1):99 – 118, 2016. doi:10.1109/TSMC.2016.2591538.
- [8] T. da Cunha Mattos, F. Santoro, K. Revoredo, and V. Nunes. A formal representation for context-aware business processes. *Computers in Industry*, 65(8):1193–1214, 2014. doi:10.1016/j.compind.2014.07.005.
- [9] O. Saidani and S. Nurcan. Context-awareness for adequate business process modelling. In *2009 Third International Conference on Research Challenges in Information Science*, pages 177–186. IEEE, 2009. ISBN 978-1-4244-2864-9. doi:10.1109/RCIS.2009.5089281.
- [10] M. Rosemann, J. Recker, and C. Flender. Contextualisation of business processes. *International Journal of Business Process Integration and*



- Management*, 3(1):47–60, 2008. ISSN 1741-8771. doi:10.1504/IJBPM.2008.019347.
- [11] S. Najar, O. Saidani, M. Kirsch-Pinheiro, C. Souveyet, and S. Nurcan. Semantic representation of context models: a framework for analyzing and understanding. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, page 1–10, 2009. doi:10.1145/1552262.1552268.
  - [12] F. M. Santoro, F. Baião, K. Revoredo, and V. T. Nunes. Modeling and Using Context in Business Process Management: A Research Agenda. *Journal of Modeling and Using Context*, pages 1–20, 2017. doi:10.21494/ISTE.OP.2017.0130.
  - [13] R. Song, J. Vanthienen, W. Cui, Y. Wang, and L. Huang. What Roles do Decisions Play in Context-Aware Business Process Management? In *ICEIS (2)*, pages 626–633. Science and Technology Publications, 2019. ISBN 978-989-758-372-8. doi:10.5220/0007736506260633.
  - [14] A. Chebba, T. Bouabana-Tebibel, and S. H. Rubin. Context in ontology for knowledge representation. In *Advanced Computational Methods for Knowledge Engineering*, pages 311–320. Springer, 2015.
  - [15] R. Song, J. Vanthienen, W. Cui, Y. Wang, and L. Huang. Towards a comprehensive understanding of the context concepts in context-aware business processes. In *Proceedings of the 11th International Conference on Subject-Oriented Business Process Management*, pages 1–10, 2019. doi:10.1145/3329007.3329020.
  - [16] H. Chen, F. Perich, T. Finin, and A. Joshi. Soupa: Standard ontology for ubiquitous and pervasive applications. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, pages 258–267. IEEE, 2004. ISBN 0-7695-2208-4. doi:10.1109/MOBIQ.2004.1331732.
  - [17] D. Zhang, T. Gu, and X. Wang. Enabling context-aware smart home with semantic web technologies. *International Journal of Human-friendly Welfare Robotic Systems*, 6(4):12–20, 2005.
  - [18] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. *arXiv preprint arXiv:2003.05055*, 2020.
  - [19] T. Labidi, A. Mtibaa, W. Gaaloul, and F. Gargouri. Cloud SLA negotiation and re-negotiation: An ontology-based context-aware approach. *Concurrency and Computation: Practice and Experience*, 32(15), 2019. doi:10.1002/cpe.5315.
  - [20] Matthias, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007. ISSN 1743-8225. doi:10.1504/IJAHUC.2007.014070.
  - [21] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414 – 454, 2013. ISSN 1553-877X. doi:10.1109/SURV.2013.042313.00197.
  - [22] O. Cabrera, X. Franch, and J. Marco. 3LConOnt: a three-level ontology for context modelling in context-aware computing. *Software & Systems Modeling*, 18(2):1345–1378, 2019. doi:10.1007/s10270-017-0611-z.
  - [23] H. Guermah, T. Fissaa, H. Hafiddi, M. Nassar, and A. Kriouile. An ontology oriented architecture for context aware services adaptation. *arXiv preprint arXiv:1404.3280*, 2014.
  - [24] A. Zaguia and R. Alroobaea. Ontological Model to Predict user Mobility. *International Journal of Advanced Computer Science and Applications*, 10(2):407–413, 2019. doi:10.14569/IJACSA.2019.0100253.
  - [25] F. Norki, R. Mohamad, and N. Ibrahim. Context ontology in mobile applications. *Journal of Information and Communication Technology*, 19(1):21–44, 2020.
  - [26] M. Asim, M. Wasim, M. Khan, W. Mahmood, and H. Abbasi. A survey of ontology learning techniques and applications. *Database*, 2018, 2018. ISSN 1758-0463. doi:/10.1093/database/bay101.
  - [27] Apache Jena. Jena Ontology API. <https://jena.apache.org/documentation/ontology/>, Date Accessed: April 1, 2019.
  - [28] M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger. *Service research challenges and solutions for the future internet: S-cube-towards engineering, managing and adapting service-based systems*. Springer, 2010.
  - [29] S. Azimi and C. Pahl. A Layered Quality Framework for Machine Learning-driven Data and Information Models. In *22nd International Conference on Enterprise Information Systems ICEIS’2020*, pages 579–587, 2020. doi:10.5220/0009472305790587.





**Leila Kord Toudeshki** received her B.Sc. and M.Sc. in computer software engineering from Islamic Azad University in Tehran. She is a Ph.D. candidate in computer software engineering at the department of computer engineering, South Tehran Branch, Islamic

Azad University, Tehran, Iran. Her research interests are in context-aware applications, business process adaptation, context ontology, large-scale software systems, software architecture, and software development.



**Mir Ali Seyyedi** received his B.Sc. from the Sharif University of Technology in 1996 and he presented a measurement model on software CMM model as his Ph.D. thesis and graduated Ph.D. in software systems. Currently, he is an Associate Professor at the computer engineering department of Islamic Azad University South of Tehran branch. He

has 2 research laboratories on enterprise and software architecture and big data processing with 40 research team members. His research interests in software architecture, large-scale systems, big data systems, and enterprise architecture.



**Afshin Salajegheh** received his B.Sc. in Computer Science from the University of Tehran, Tehran, in 1995, the M.Sc. in Computer Engineering from Islamic Azad University, Tehran in 1998, and the Ph.D. degree in Computer Software Engineering from Islamic Azad University, Tehran, in 2008. Currently, he is an Assistant Professor in the Department of Computer Engineering at the IAU

University, South Tehran Branch. His research interests include software engineering, data mining, the Internet of things, and evolutionary computing.

