The Pervasive Information System Adaptation: Android Device Context

Fatma Achour, Anis Jedidi, Faiez Gargouri

MIRACL
Multimedia, Information Systems and Advanced Computing Laboratory
Sfax University, Tunisia

Email: fatma.achour@gmail.com, anis.jedidi@isimsf.rnu.tn, faiez.gargouri@isimsf.rnu.tn

***Abstract*—The conceptual step in the pervasive adaptation system is mostly aimed at the success of such a system. It is for this reason that the development of a conceptual adaptation system helps developers to implement their adaptation systems. Accordingly, several categories of contextual information can be presented in a pervasive information system; as such, the network context, the location context, the service context, the application context, the device context, and the person context. Again, a detailed description of each of these contextual information categories allows achieving a better adaptation of the applications to the contexts of use. However, the device context shows the focal point of any information system. In fact, the adaptation of the pervasive information systems, when using contexts, rests on the existing physical constraints in this context. In this paper, we present the architecture of adapting the applications to the pervasive system based on the semantic web services. Specifically, we are interested in the adaptation of the device context. The device contextual data are collected by using an Android program. The adaptation rules are also created using the Jena toolkit.**

Keywords–Adaptation; Model; Android Device; Jena rules; RDF; OWL; OWL-S.

I. INTRODUCTION

The information systems evolution is correlated with telecommunication as well as connectivity hardware and software development. These types of information systems are designed to create a transparent and an inter-operable environment to ensure better information shared between the various types of information systems with heterogeneous information resources.

Thanks to the technological developments and the new technologies integration in all applications of everyday life, as a matter of fact, connectivity enhanced accessibility to the resources. This progress has enormously given the user a free interaction to access the different resources anywhere, anyhow and at any time: the systems have consequently become pervasive and ubiquitous.

The pervasive or the ubiquitous systems are actually designed to make information available anywhere and at any time. These systems, however, must be used in different contexts according to the user's environment and profile as well as the used terminal. One of the major problems of the pervasive system is the adaptation of the applications to the user's situation [1]. Several research works [1][2], indeed, are dedicated to seek out a solution to this problem and to develop an adaptation framework. Nevertheless, the research works in the pervasive system conceptual adaptation domain are too limited. Every researcher in this field seeks to develop a system allowing the implementation of adaptation without taking into consideration the design phase of this adaptation. In this paper, we suggest a complete, a generic and a scalable architecture to conceptually adapt the application to the user's situation. In this, we mainly focus on the Android devices adaptation in the user's situation based on the semantic web services creation. To do this, we propose a generic model to design the device context in the pervasive system. We also integrate the proposed model in the semantic web service description or structure. Eventually, we define a set of rules that can be applied to this description.

In this paper, we present in the first part a state of the Art on the pervasive computing. In the first section, we present the pervasive system adaptation. In the second section, we describe the web service semantic description structure OWL-S Ontology. In the second part, we present our proposed framework. The first section of this part is dedicated to present the proposed description of the device context. The following section deals with the use of this description to adapt the applications in the pervasive system.

II. STATE OF THE ART

Context modeling is a vital aspect in pervasive computing. Because context-aware applications must be adapted to the changing situations, they need a detailed model of the user's activities and entities in the surroundings that let them share the user's perceptions of the real world. One of the basic steps in the development of context-aware applications is, therefore, to provide a formalized representation and standardized access mechanisms to the context information.

In this paper, we present the existing works to adapt the application to the use's context in the pervasive system and we show the existing structure to integrate the contextual information in the semantic web service description.

A. The pervasive system adaptation

Several research works are all for the adaptation of the application to the pervasive system. Since we are inclined to the conceptual adaptation, we present the conception phase (the second phase in the software engineering lifecycle) of two existing adaptation works: SECAS (Simple Environment For Context-Aware Systems) [1] and COCA (A Collaborative Context-Aware Service Platform for Pervasive Computing) [2].

 ISBN: 978-1-61208-353-7

The context models that use the markup scheme approaches are commonly used for the profile data representation. This type of model is used by several research works. Among these works, we can cite the COCA [2] and the SECAS [1] platforms. The COCA platform proposes a semantically rich model for collaboration, representation and context management [2]. It uses a contextual model representation based on a hybrid approach using ontology and relational databases (HCoM/EHRAM). EHRAM is a conceptual context representation meta-model and HCoM is a hybrid model that uses the components of EHRAM in ontology and relational schema. The ontology part represents the semantic aspect of the context data and the relational schema represents the context data itself. The EHRAM model includes: person context, device context, physical environment context, network context, activity context, service context, and location context.

The limitations of this work are inherent in the separation of the various categories of contextual information. For example, there are no semantic relationships between the two contexts: the personal context and the location context. However, we present the relation”locatedIn” between these two contexts (the context personal context is located in the location context).

SECAS attaches a great importance to the context management without showing how to modify the behavior of the application to adapt the context [3]. The application use context is defined as a five-dimensional vector: terminal, communication, user, location, and environment. To store the context parameters, it uses an XML representation based on the CSCP model [4].

In SECAS work, we note the absence of a design general model to conceive the information used to adapt the application in the pervasive system. Also, the adaptation is based on the Petri Nets representation which describes the services of the application and their dependencies. Therefore, the adaptation is functional but is not semantic.

B. The OWL-S Ontology

In the second part of this state of the art, we present the existing works so as to integrate the contextual information in the semantic web service description. In the first place, we present the used structure to describe the semantic web service: OWL-S. In the second place, we present the existing OWL-S extension to integrate the contextual information in order to provide an adapted semantic web service.

1) The OWL-S Ontology Presentation: OWL-S is a Web Services ontology that specifies a conceptual framework to describe the semantic web services. OWL-S is also a language based on the DARPA work of its DAML program and takes the result of DAML-S (DARPA Agent Markup Language Service). It was incorporated into W3C in 2004 within the interest group of semantic web services at the OWL recommendation [5].

The initial purpose of OWL-S is to implement the semantic web services. OWL-S is based on OWL to define the abstract categories of entities and events in terms of classes and properties. OWL-S uses this ontology language description to define a particular ontology for the web services. This ontology is used to describe the web service properties as well as its services available to the public. The OWL-S structure regroups a set of ontology. Each one provides a functionality to describe the web service semantically. The ontology main classes described by OWL-S [6] are defined by the following figure (see Figure 1).

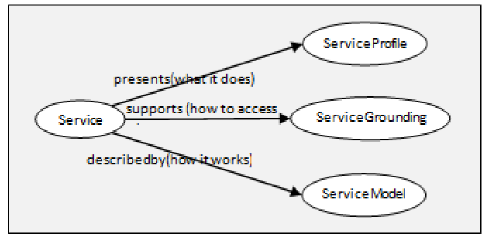

Figure 1. The principal OWL-S classes.

The necessity to use the OWL-S ontology is justified by the creation of a semantic web service that has a dynamic description. This dynamic is provided by the addition of contextual descriptions to the OWL-S structure. The description depends on the use of the context of a pervasive system.

2) The existed OWL-S extension: Several research works take the advantage of the existing OWL-S structure to describe the different contexts. In this paper, we present two research works of Qiu et al. [7][8] and Ben Mokhtar [9]. In [7][8], the authors propose an adaptation system based on the service composition approach. To do this, Qiu et al. [8] offer three context categories : the user’s context, the web service context and the environmental context.

The user’s context (”U-Context”) specifies the context information about the user. In this context, the authors defined two types of contextual information: the user’s static context (profile, interest, and preferences) and the user’s dynamic context (location, current activity and task trying to achieve). The web service context (”W-Context”) includes the not functional contextual information (price, execution time, confidence degree). The environmental context (”E-Context”): this category collects the context information about the user’s environment (time, date, etc.).

Each context category is represented by the OWL ontology and is integrated in the existing OWL-S extension ontology to introduce the OWL-SC (OWL-S for context) [8]. The latter is intended to describe a general contextual information (see Figure 2) based on the users’ description.

The proposed structure focuses only on the user’s context description. However, it presents a vision for the integration or the addition of more information to the OWL-S structure. Ben Mokhtar et al. [9] research works propose a system to adapt the web services to a pervasive environment [9]. The context definition includes the description of four types of contextual information: the context sensors, services, devices and users.

In addition, the contextual adaptation in this work is based on the service representation and the user’s task representation. In the service representation, the authors describe the services using OWL-S extended with context information. This information is divided into a high level context attributes,

 ISBN: 978-1-61208-353-7

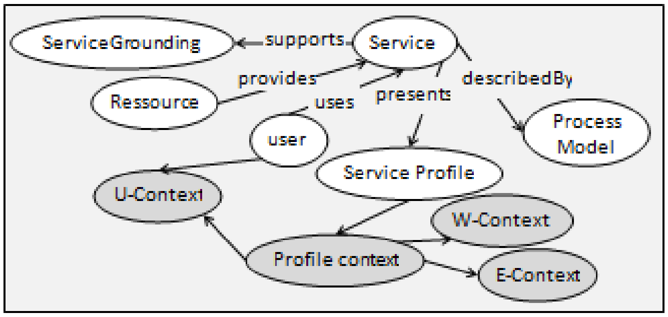

Figure 2. The OWL-SC Ontology.

preconditions and contextual effects. However, via the user's task representation, the user's task representation is performed while extending the OWL-S service model ontology (see Figure 3). To do this, Ben Mokhtar et al. [9] propose to integrate the quality conditions service descriptions and the context conditions required by the user's task in the OWL-S structure.

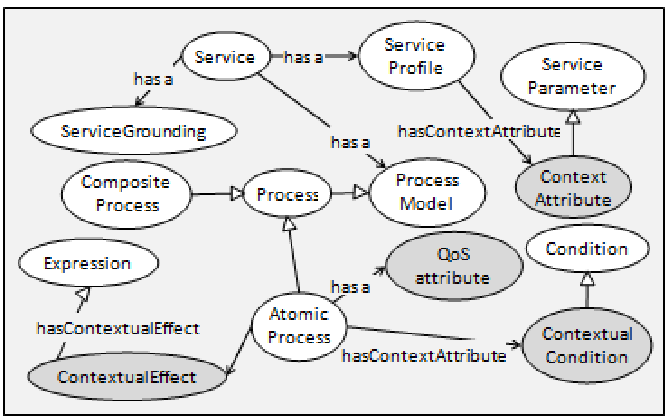

Figure 3. The OWL-S ontology extension for the pervasive system.

III. THE PROPOSED FRAMEWORK

In the proposed framework, we are interested in the semantic web service to adapt the application to the pervasive system. The target audiences of our proposed framework are the developers in the pervasive system. Therefore, we proposed two description levels (see Figure 4):

- **The generic level:** At this level, we create a semantic web service. In this semantic web service description, we create OWL-S ontology. In this ontology, we integrate the highest level of the pervasive system description. This level is created to collect the context description corresponding to the user's needs and to apply the user's rules to the collected information.
- **The specific level:** At this level, we create a set of six semantic web services corresponding to the pervasive information categories: network, device, user, location, service and application. In each semantic web service, we integrate the context description. These semantic web services are created to instantiate and to describe the user's situation.

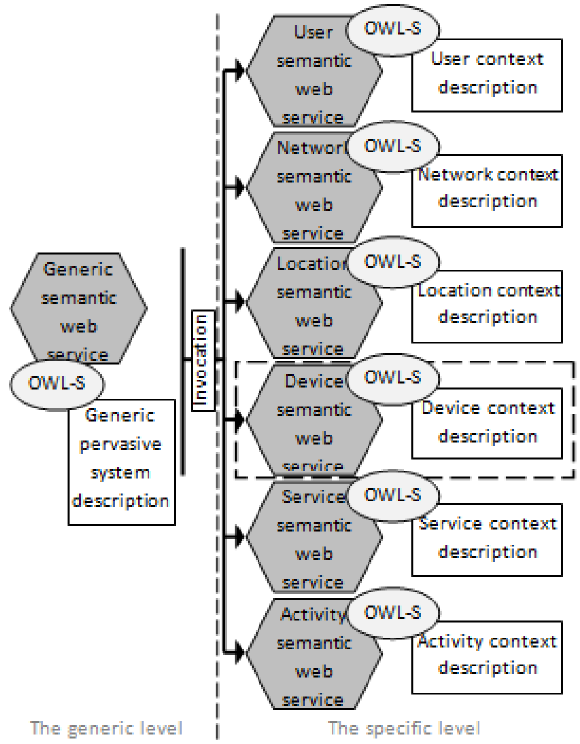

Figure 4. The purposed architecture levels.

The goal of the information separation in the specific level is to facilitate the conceptual adaptation by the use of the most appropriate contextual information and to reduce the search time by the specific semantic web service invocation used in the adaptation phase. In this paper, the device context is of a growing interest.

A. *The proposed description framework: the generic and device contexts*

The pervasive information system entails six information categories (device, network, activity, service, location and user) [10]. In our proposal, we distinguish two levels of description. The first level presents the generic level and the second level presents the specific level (see Figure 4). In the generic level, we create a generic semantic web service. In the OWL-S description structure of the generic web service, we integrate the generic OWL description. In the specific level, we create a semantic web service for each pervasive information system categories. Also, for each category, we create a semantic web service. In the specific semantic web service description, we integrate the information system category OWL description in the OWL-S structure.

1) The proposed description of the generic level: The generic semantic web service generally describes the pervasive environment. It has all the necessary information about each web service context shown in the second level. This information is described in an extended OWL-S ontology (see Figure 5). The extended OWL-S ontology includes the basic information examining a pervasive system.

The pervasive context is presented by the "PervasiveContext" OWL class. The activities in a pervasive system are presented by the "A-Context" OWL class. They exist in a device. The latter is symbolized by the "D-Context" OWL class. Each device ("D-Context") offers services in a pervasive system. The services are presented by the "S-Context" OWL class. The latter regroups the characteristics of the services provided by the pervasive system. All the devices existing in the pervasive works are modeled through the "N-Context" OWL class.

The two classes "D-Context" and "U-Context" represent the entire agent that exists in a pervasive system. For this reason, we position the two classes as a sub-class of the» Agent" OWL class.

Each of the classes presented in the proposed OWL-S structure will be transformed into ontology. The latter regroups the classes and the attributes shown by the OWL semantic relation "owl:onProperty". The ontological structures are used to detail the contexts defined in the pervasive system.

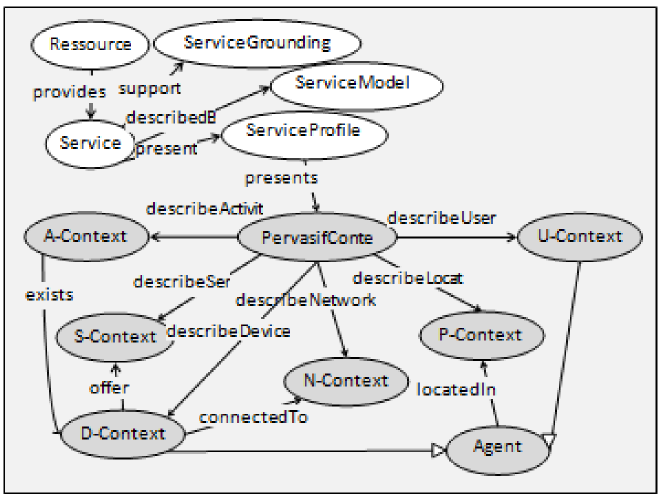

Figure 5. The proposed structure for the first level of description.

2) The proposed description of the device context: The device side is very significant in the pervasive information system since the pervasive system is accessible anywhere, anyhow and to anything. Indeed, such a system can be executed according to the existing hardware.

The OWL ontology is created to describe the device context and the activity profiles in the pervasive information system. Each device has a configuration, rules and preferences. In the mobile system, a new communication method has emerged to satisfy the user's needs. Such a method paves the way for the propagation of intelligent systems through the invention of smart phones, such as "blackberrys", "iPhone", and touch pads, such as the "iPad". In order to ensure adaptation, the pervasive system must capture a material characteristic to ensure the answer to the query in accordance with the hardware configuration. To ensure the generality of the proposed device context model we define a "key" and "values" properties for each class. These properties can catch any value depending on the user's contextual situation.

This ontology is inserted in the OWL-S structure. The original purpose of OWL-S is to implement the semantic web services. OWL-S is based on OWL to define the abstract categories of entities and events in terms of classes and properties. OWL-S uses this ontology language description to define a particular ontology for the web services. This ontology is used to describe the web service properties as well as its services available to the public. The OWL-S structure regroups a set of ontology. Each one provides a functionality to describe the web service semantically. The ontology main classes described by OWL-S are defined by the following figure (see Figure 6).

The necessity to use the OWL-S ontology is justified by the creation of a semantic web service that has a dynamic description. This dynamic is provided by the addition of contextual descriptions to the OWL-S structure. The description depends on the use of the context of a pervasive system. To integrate the device context ontology, we use the two classes "service" and "service profile".

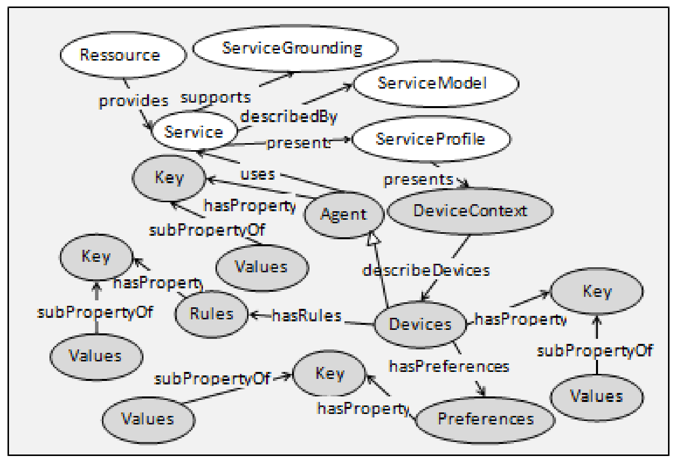

Figure 6. The proposed structure for the device context.

3) The data properties used in the device context: In this section, we present a list of properties and sub-properties (see. Table 1) used to describe the Android device context in the pervasive system. These properties are used as an instance of the presented "key" and "value" properties (see Figure 6). This list is detected automatically using Android programs and submitted to the device semantic web service using SOAP. This list is used to create an RDF instance to the device context in conformity with the presented OWL-S structure (see Figure 6). This service is proposed by the device semantic web service. Since we propose two generic properties in the created model in our proposed framework, we present a Java Server Page to add other properties to the created RDF device instance and to the created OWL-S extension. This Java Server Page is designed for the developer in the pervasive system representing the target audience of our proposed framework.

B. The proposed adaptation framework

In the proposed adaptation framework, we present two different works. The first work is concerned with the proposal of the adaptation phase and communication between the generic and the device context. The second work is about applying the rules to the created models which offer a complete conceptual adaptation system.

TABLE I. THE DEVICE CONTEXT PROPERTIES.

Classes	Sub-properties	Properties
Agent	AgentCharact	AgentType AgentName
Devices	BatteryCharact	Voltage Temperature Technology Status Scale Presence Plugged Level Health
	DBluetoothChar	DBluetoothAdress DBluetoothName
	DeviceCharact	DeviceType ScreenBrightness ScreenDimension DeviceName CPUSpeed
	LBluetoothChar	LBluetoothAdress LBluetoothName
	Volume	MaxVolume ActualVolume VolumeMode
	MemoryCharact	TotInteMem AvailInteMem AvailExteMem TotExteMem TotalRAMMem AvailRAMMem
	CameraCharact	CameraNumber CameraSize CameraEncodFmt
Preferences	Language	LanguageCountry UsedLanguage
Rules	SupScreenMode	ScreenModeValue SupResolution
		ResolutionValue
	SupPictureFormat	PictFormatVal
	SupAudioFormat	AudioFormatVal
	SupVideoFormat	VideoFormatVal

1) The generic and the device semantic web services communication: The proposed architecture is made up of two fundamental parts (see Figure 7). The first one consists in

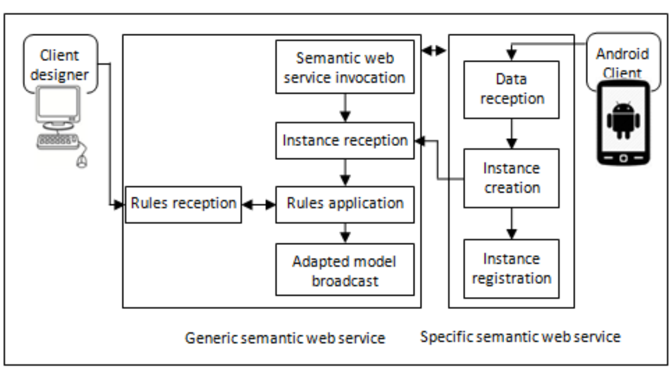

Figure 7. The purposed Framework Architecture.

developing a generic semantic web service and the second one regroups a set of specific web services corresponding to the pervasive information system categories [10]. We develop a set of semantic web services using an OWL-S structure. In the generic semantic web service case, we integrate the semantic relation between the different specific semantic web services. For each one of the pervasive information system categories (person, device, network, service, application and location), we create a specific semantic web service. Also, we integrate their classes into the OWL-S description which corresponds to their semantic web service.

The value of the generic web service creation resides in ensuring the specific web services communication by regrouping their created instances and applying the user's rules to the latter. The specific web services are created in an attempt to receive the Android device characteristics and to create an RDF instance in accordance with the OWL-S description.

2) The semantic web service invocation: The classical web service architecture is composed of three elements. The first element represents the user. The second element stands for the provider. The last element is the registry. The interaction between the three elements is ensured by SOAP. The purpose of the semantic web service is to integrate the ontology description in the OWL-S web service description. In this section, we present the interaction between the Android client, the generic semantic web service client and the specific semantic web service (see Figure 8).

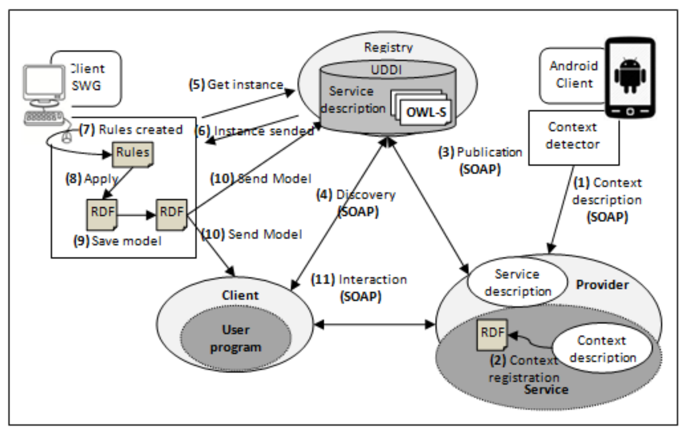

Figure 8. The proposed semantic web service architecture.

(1) To participate in the adaptation framework, the Android devices must send their characteristics to the specific semantic web service.

(2) Based on the received information from the Android device, the provider in the specific semantic web service creates an RDF instance conforming to the OWL description inserted in OWL-S description.

(3) The provider publishes the service and the created instance in the registry using OWL-S description.

(4) The client discovers the service and the created instance.

(5) The generic web service client requests the created instance from the registry.

(6) The registry sends the requested instance to the generic web service client.

(7) The generic web service client creates his rules using a jsp page.

(8) The generic web service client applies the created rules to the created instance.

(9) The generic web service client saves the adapted model.

(10) The adapted model is sent to the registry and to the specific web service client.

(11) The interaction between the client and the provider is started again.

In the next section, we will present the pervasive information system adaptation framework to the device context.

3) The Jena rules: To validate the proposed instance, we propose to execute two Jena rules in the created RDF instance (see Figure 9). The result of the rules execution is a model used by the developers in the pervasive system domain to ensure the adaptation.

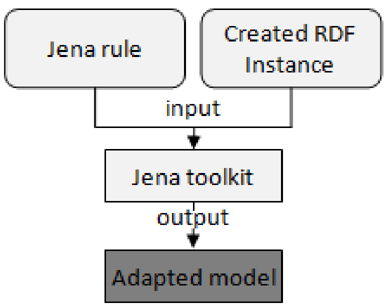

Figure 9. The Jena input/output.

We decide to use Jena [11] because it defines a java package that can manipulate the files in any of the standard RDF storage formats. Additionally, Jena can store and read RDF data in a relational database. Jena offers a statement-centric (based on the subject-predicate-object structure) support to manipulate the RDF and OWL data, and comes with a built-in RDF query language, SPARQL. Jena provides a programmatic environment for RDF, RDFS, OWL and SPARQL and includes a rule-based inference engine. In the next section, we will present an example of two Jena rules.

Rule 1: If the battery level is lower than 50% and the screen brighteness equal to 100%. The value of this latter is changed to 30%.

The Rule description:

$$((Battery.Level(50) \wedge DeviceCharacteristics.ScreenBrightness(100)) \Longrightarrow (DeviceCharacteristics.ScreenBrightness(50) \wedge Volume.ActualVolume(5.0)))$$

The Jena code:

```
@prefix rdf:
http://www.w3.org/1999/02/22-rdf-syntax-ns#
@prefix perSys:
http://example.org/PervasiveSystem#
@prefix xs:
http://www.w3.org/2001/XMLSchema#
[PreferredVolume:
(?d ?rdf:type ?t1),
(?c ?rdf:type ?t2),
(?c perSys:Level ?a)
lessThan(?a, 50)
(?c perSys:ScreenBrightness "100%")
(?c perSys:ActualVolume "15.0")
->
(?c perSys:ScreenBrightness "30%")
(?c perSys:ActualVolume "5.0")
```

The execution result of the second rule presents the RDF model as described below. The RDF model regroups the device characteristics after the execution rule where the screen brightness equals to 100% and the battery level equals to 50%. Also, the execution result of this rule is to change the screen brightness value into 30%.

```
<perSys:Device> <rdf:Description
rdf:about=
"http://PervasiveSystem
#DeviceCharacteristics">
<perSys:CPUSpeed
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#float"
>1.0</perSys:CPUSpeed>
<perSys:ScreenBrightness rdf:datatype=
"http://www.w3.org/2001/XMLSchema#float"
>30</perSys:ScreenBrightness>
...
<perSys:DeviceName
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string"
>Unknown sdk</perSys:DeviceName>
</rdf:Description>
</perSys:Device>
...
<perSys:Device>
<rdf:Description
rdf:about=
"http://PervasiveSystem
#BatteryCharacteristics"> <perSys:Voltage
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#int"
>0</perSys:Voltage>
...
<perSys:Presence rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string"
>true</perSys:Presence>
<perSys:Plugged
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#int"
>1</perSys:Plugged>
<perSys:Level
rdf:datatype=
```

```
"http://www.w3.org/2001/XMLSchema#int"
>50</perSys:Level>
<perSys:Health
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#int"
>2</perSys:Health>
</rdf:Description>
</perSys:Device>
<rdf:Description
rdf:about="http://PervasiveSystem#Volume">
<perSys:MaxVolume
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string"
>15.0</perSys:MaxVolume>
<perSys:ActualVolume rdf:datatype="http:
//www.w3.org/2001/XMLSchema#string"
>5.0</perSys:ActualVolume>
<perSys:VolumeMode
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string"
>Normal mode</perSys:VolumeMode>
</rdf:Description>
</perSys:Device>
```

The device context does not present the totality of information included in the pervasive system. In fact, the pervasive system presented a collection of six categories of contextual information. We presented in previous work our design of each category in this paper we present a rule applied to two contextual information categories: the device context and the location context.

Rule 2: The devices located in "Tunisia, ISIM Gabes, Amphi 1", the preferred Volume Mode must be changed from "Normal Mode" to "Silent Mode".

The Rule description:

$$((Adress.CountryName("Tunisia") \wedge Adress.Region("Gabes") \wedge Adress.FeatureName("ISIM") \wedge Adress.SpecificLocation("Amphi1") \wedge Volume.VolumeMode("Normalmode")) \Longrightarrow Volume.PreferedMode("Silentmode"))$$

Jena code:

```
@prefix rdf:
http://www.w3.org/1999/02/22-rdf-syntaxns#
@prefix perSys:
http://PervasiveSystem#
@prefix xs:
http://www.w3.org/2001/XMLSchema#
[PreferredLocation:
(?d ?rdf:type ?t1),
(?c ?rdf:type ?t2),
(?c perSys:CountryName "Tunisia"),
(?c perSys:Region "Gabes"),
(?c perSys:FeatureName "ISIM"),
(?c perSys: SpecificLocation "Amphi1")
(?h ?rdf:type ?t3),
(?h perSys:DeviceName ?r)
->
(perSys:DesignedDevice ?rdf:type ?t3)]

@prefix rdf:
http://www.w3.org/1999/02/22-rdf-syntaxns#
@prefix perSys:
http://PervasiveSystem#
@prefix xs:
http://www.w3.org/2001/XMLSchema#
[PreferredVolumeMode:
(?v ?rdf:type ?t3),
(?w ?rdf:type ?t4),
(?w perSys:DeviceName ?r)
(?y ?rdf:type ?t5),
(?y perSys:VolumeMode "Normal mode")
->
(?y perSys:PreferedMode "Silent mode") ]
```

The execution of the second rule aims to define two Jena rules: "PreferredLocation" and PreferredVolumeMode". The first one permits to know the devices name located in "Amphi1" and the second one permits to define the desired volume mode ("Silent Mode").

```
<perSys:Location>
<rdf:Description
rdf:about="http://PervasiveSystem#Adress">
<perSys:FeatureName>ISIM
</perSys:FeatureName>
<perSys:Region>Gabes
</perSys:Region>
<perSys:CountryName>Tunisia
</perSys:CountryName>
<perSys:SpecificLocation>Amphi1
</perSys: SpecificLocation>
<perSys:CountryCode>TN
</perSys:CountryCode>
</rdf:Description>

...
<rdf:Description
rdf:about="http://PervasiveSystem
#DesignedDevice"> Samsung GT-S5360
</rdf:Description>
</perSys:Location>
<perSys:Device>
<rdf:Description
rdf:about=
"http://PervasiveSystem
#DeviceCharacteristics">
<perSys:CPUSpeed
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#float"
>1.0</perSys:CPUSpeed>
<perSys:ScreenBrightness rdf:datatype=
"http://www.w3.org/2001/XMLSchema#float"
```

```
>30</perSys:ScreenBrightness>
...
<perSys:DeviceName
rdf:datatype=
"http://www.w3.org/2001/XMLSchema#string"
> Samsung GT-S5360</perSys:DeviceName>
</rdf:Description>
<rdf:Description
rdf:about=
"http:PervasiveSystem#Volume">
<perSys:PreferedMode> Silent mode
</perSys:PreferedMode>
<perSys:MaxVolume>15.0
</perSys:MaxVolume>
<perSys:ActualVolume>3.0
</perSys:ActualVolume>
<perSys:VolumeMode>Normal mode
</perSys:VolumeMode>
</rdf:Description>

...
</perSys:Device>
```


IV. Conclusion

The most detailed description or modeling provides an accurate level of adaptation. In this paper, we present a pervasive information system adaptation using an Android device. In order to do this, we created a model including all the contextual information necessary to adapt the application to the device context. In fact, we proposed an extensible model to describe the device context through the OWL-S extension. Also, we defined an important number of contextual information.

Moreover, we tried to define a JSP interface to add more data to the suggested extensible model. In this paper, we propose a complete approach to provide a framework to adapt applications to the device context. In the future work, we seek to complete the proposal of the pervasive information system adaptation framework gathering all contextual information categories (user, device, network, application, location and service).

References

[1] T. Charri, “Adaptating pervasive applications in multi-contextual environments,” Computer Science, National Institute of Applied Sciences, LIRIS, September 2007.

[2] D. E. Dedefa, “Context modeling and collaborative context-aware services for pervasive computing,” Computer Science, National Institute of Applied Sciences, INSA de Lyon, December 2007.

[3] T. Chaari, F. Laforest, and A. Celentano, “Adaptation in Context-Aware Pervasive Information Systems: The SECAS Project,” Int. Journal on Pervasive Computing and Communications(IJPCC), vol. 3, no. 4, Dec. 2007, pp. 400–425, Available: http://liris.cnrs.fr/publis/?id=2996 Retrieved: June, 2014.

[4] T. H. Sven Buchholz and G. Hübsch, “Comprehensive structured context profiles (cscp): Design and experiences,” in Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, ser. PERCOMW ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 43–, Available: http://dl.acm.org/citation.cfm?id=977405.978623 Retrieved: June, 2014.

[5] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, “Owl-s: Semantic markup for web services.” Internet [http://www.w3.org/Submission/2004/SUBM- OWL-S-20041122/], 2004, Available: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/ Retrieved: June, 2014.

[6] M. D, M. Burstein, M. D, McIlraith, M. Paolucci, K. Sycara, M. D, M. Burstein, M. D, M. S, M. Paolucci, K. Sycara, M. D, E. S, and N. Srinivasan, “Bringing semantics to web services with owl-s,” World Wide Web Journal, vol 10, no. 3, pp. 243-277, no. CMU-RI-TR-, August 2007.

[7] L. Qiu, Z. Shi, and F. Lin, “Context optimization of ai planning for services composition,” in ICEBE. IEEE Computer Society, 2006, pp. 610–617.

[8] L. Qiu, L. Chang, F. Lin, and Z. Shi, “Context optimization of ai planning for semantic web services composition,” Service Oriented Computing and Applications, vol. 1, no. 2, 2007, pp. 117–128.

[9] S. B. Mokhtar, D. Fournier, N. Georgantas, V. Issarny, S. B. Mokhtar, D. Fournier, and N. Georgantas, “Context-aware service composition in pervasive computing environments,” in RISE, vol. 3943. Springer, 2005, pp. 129–144.

[10] F. Achour, A. Jedidi, and F. Gargouri, “A two-leveled semantic web service description of the pervasive information system.” in MobiWIS, ser. Lecture Notes in Computer Science, F. Daniel, G. A. Papadopoulos, and P. Thiran, Eds., vol. 8093. Springer, 2013, pp. 35–48.

[11] B. McBride, “Jena: a semantic web toolkit,” IEEE Internet Computing, vol. 6, no. 6, 2002, pp. 55–59.