Open access • Proceedings Article • DOI:10.1145/2187980.2188231

# Towards an interoperable device profile containing rich user constraints

— Source link

Cédric Dromzée, Sébastien Laborie, Philippe Roose

**Institutions:** University of Pau and Pays de l'Adour

**Published on:** 16 Apr 2012 - The Web Conference

**Topics:** Context model, Adaptation (computer science), Semantic Web, Context (language use) and Mobile device

# Towards an interoperable device profile containing rich user constraints

Cédric Dromzee, Sébastien Laborie, Philippe Roose

# Towards an Interoperable Device Profile Containing Rich User Constraints

Cédric Dromzée
LIUPPA / AEXIUM SAS
29 rue des Pins
F-40230 Saubion
dromzee@aexium.com

Sébastien Laborie
LIUPPA/T2I – Univ. of Pau
2, Allée du Parc Montaury
F-64600 Anglet
Firstname.Lastname@univ-pau.fr

Philippe Roose
LIUPPA/T2I - University of Pau
2, Allée du Parc Montaury
F-64600 Anglet
Firstname.Lastname@univ-pau.fr

## ABSTRACT
Currently, multimedia documents can be accessed at anytime and anywhere with a wide variety of mobile devices, e.g., laptops, smartphones, tablets. Obviously, platforms heterogeneity, user's preferences and context variations require documents adaptation according to execution constraints, e.g., audio contents may not be played while a user is participating at a meeting. Current context modeling languages do not handle such a real life user constraints. They generally list multiple information values that are interpreted by adaptation processes in order to deduce implicitly such high-level constraints. This paper overcomes this limitation by proposing a novel context modeling approach based on services where context information are linked according to explicit high-level constraints. In order to validate our proposal, we have used Semantic Web technologies by specifying RDF profiles and experiment their usage on several platforms.

## Categories and Subject Descriptors
H.1.1 Systems and Information Theory

## General Terms
Theory.

## Keywords
Context Modeling, Profile, Multimedia Documents, Devices and Users constraints.

## 1 INTRODUCTION
Nowadays, a huge amount of multimedia documents can be created and accessed by users. These documents may be composed of different types of contents, such as videos, audios, texts and images. Good examples of multimedia documents are, for instance, Web pages or SMIL presentations [1]. In those documents, multimedia contents are synchronized and organized according to the graphical layout of the presentations. Moreover, users may be able to interact with presentations by selecting particular of its elements (e.g., a click on a picture plays a video).

Besides, many mobile devices (e.g., laptops, smartphones and tablets) are able to display multimedia documents. This universal access allows users to consult documents anytime and anywhere. However, such devices have heterogeneous capabilities and characteristics in terms of hardware (e.g., screen size, battery) and software (e.g., players, codecs) characteristics. Moreover, user's preferences or handicaps may prevent from playing specific multimedia contents. For instance, a user may avoid reading texts written in French and/or avoid playing audio contents while he is participating at a meeting. All these restrictions introduce constraints that have to be specified with a profile.

In a profile, various categories of information have to be managed: (1) **device characteristics** (hardware and software), (2) **context information** related to interactions between the user and the device, such as the preferred languages or the surrounding devices and (3) **document structure**, like the types of contents which are played or the presentation organization (e.g., layout, multimedia contents synchronization, hypermedia links). Consequently, if a multimedia document does not comply with some constraints that are specified inside a target profile, the document may not be correctly executed on the target device. Thus, in order to display documents on any devices, these ones have to be adapted, i.e., transformed in order to comply with the target profiles.

Since the last decade, a fair amount of research has been conducted on multimedia document adaptation, e.g., [2], [3], [4]. Considering some target profiles, these approaches combined multiple operators: transcoding (e.g., AVI to MPEG), transmoding (e.g., text to speech) and transformation (e.g., text summarization). Of course, each profile expressiveness is exploited by these approaches in order to determine a combination of these operators, e.g., [5], or their deployments (e.g., for saving battery energy), e.g., [6] and [7].

However, each proposition exploits specific profile format, which usually contains a list of multiple information values, such as the screen size, the user language and the battery power. Consequently, an adaptation process has to interpret such profiles and to deduce implicitly some constraints. For instance, if the battery power is lower than 10%, avoid playing hi-quality videos. Obviously, each adaptation mechanism may deduce different constraints that in many situations might be wrong, thus providing incorrect adapted documents. Furthermore, current context modeling languages do not consider expressing such high-level constraints, e.g., [8] and [9], while they might be very useful to guide the adaptation process.

In this paper, we propose to overcome this limitation by defining a new profile description model where (1) profile information are organized into facets (e.g., device characteristics, context information and document structure) and composed of services that either provide data or require modifications, and (2) some profile information are linked by explicit high-level constraints. Thanks to this proposal, our profiles may migrate from one platform to another one while preserving the specified constraints, thus ensuring interoperability. In order to validate our proposal, we encode profiles in RDF/XML [10] and evaluate several query executions on different platforms. Experimental results confirm that adaptation processes can access our profile descriptions efficiently.


*WWW 2012 Companion*, April 16-20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

This paper is organized as follows. Firstly, the related work section gives an overview of the current existing profiles and context modeling approaches. Then, we detail our service-based profile specification and we illustrate it through real-life examples. Thereafter, we propose a corresponding model, named Semantic Generic Profile (SGP), and some instantiations encoded in RDF/XML. Some queries that may be used by adaptation processes are presented in order to show how they may exploit our profile descriptions. Experiment results have been also conducted on several platforms to measure query efficiency. Finally, in the last section, we conclude and present some future work.

# 2 RELATED WORK

Since the last decade, a fair amount of research has been proposed in order to model devices characteristics and users contexts [11] [8] that are further exploited by multimedia document adaptation processes. We have noticed that some of these approaches provide exclusively a descriptive view of context information (e.g., CC/PP, UAProf, WURFL), while others propose enhancements with some constraints expressions (e.g., CSCP, Context-ADDICT). In this section, we present an overview of these approaches.

## 2.1 CC/PP

CC/PP (Composite Capability / Preference Profiles) is a W3C recommendation for specifying device capabilities and user preferences. This profile language is based on RDF and is maintained by the W3C Ubiquitous Web Applications Working Group (UWAWG) [12]. The profile structure is very descriptive since it lists sets of values which could correspond to the screen size, the browser version, etc. Indulska et al. [13] have expanded the vocabulary of CC/PP to describe the location, network characteristics and application dependencies. However, the CC/PP structure lacks functionality, e.g., it limits complex structure description by forcing a strict hierarchy with two levels. Furthermore, it does not consider the description of relationships and constraints between some context information. Finally, it is necessary to extend the vocabulary used in CC/PP to include new elements corresponding to hardware profile [14].

## 2.2 UAProf

UAProf [15] is based on RDF and is a specialization of CC/PP for mobile phones. More precisely, its vocabulary elements use the same basic format as the ones used in CC/PP for describing capabilities and preferences for wireless devices. Thus, it describes specific items, such as the screen size, the supported media formats, etc. UAProf is a standard adopted by a wide variety of mobile phones and provides detailed lists of information about the terminal characteristics. However, this standard is limited to the description of wireless telephony equipment characteristics. Hence, it does not allow users to express requirements.

## 2.3 CSCP

CSCP (Comprehensive Structured Context Profiles) uses RDF and is also based on CC/PP. In contrast to CC/PP, CSCP has a multilevel structure and is expandable. Even if CSCP provides a description of the context, which is not limited to two hierarchical levels, this proposal does not describe relationships, constraints and dependencies between context information. CSCP models the constraints but does not define any action that have to be taken into account by adaptation processes [16]. Indulska et al. concluded that this model is not intuitive and difficult to use in order to describe complex information [13]. It is developed as a proprietary model for specific domains [17].

## 2.4 Context-ADDICT

Context-aware Data Integration Customization and Tailoring proposes the Context Dimension Tree. The context can be represented with hierarchical structure composed of a root and some level nodes. The authors propose constraints and relationships among values [18]. In Context-ADDICT, the data sources are generally dynamic, transient and heterogeneous in both their data models (e.g., relational, XML, RDF) and schemas [19]. The Context-ADDICT approach lacks the features not relevant for the data tailoring problem such as Context History, Context Quality Monitoring, Context Reasoning and Ambiguity and Incompleteness Management [8]. This model is not generic since it depends on the application used. Hence, the data are structured according to the application requirements. It isis therefore necessary to know in advance the considered context.

## 2.5 Generic profiles for the personalization of Information Access

Chevalier [20] proposes a generic UML profile for describing the structure and semantics of any type of user profile information. This contribution is used to describe the semantic links between elements and incorporate the weighting of the elements. The semantic graph is described thanks to a logic-oriented approach [21] with RDF, RDFS and OWL. However, this model does not express actions under conditions (e.g., increase audio volume according to a specific situation).

## 2.6 WURFL

WURFL (Wireless Universal Resource File) is an XML description of mobile devices resources. WURFL contains information about the capabilities and functionality of mobile devices with more than 500 "capabilities" for each device (divided into 30 groups). This project is intended to adapt Web pages on mobile devices [22], [23]. But, unfortunately, the user cannot specify explicit constraints (e.g., decrease screen luminosity if battery level is below 10%).

## 2.7 SPICE

The European project SPICE[1] (Service Platform for Innovative Communication Environment) has specified a user profile structure that considers the creation of different sub-profiles for different services and enables the inclusion of contextual constraints which specify specific usage conditions. They have proposed an RDF/XML description of such profiles. Nevertheless, this approach requires a new sub-profile creation each time some conditions have to be considered.

## 2.8 Others approaches

Alternative approaches to design profiles with markup languages exist, such as PPDL (Pervasive Profile Description Language) [24] or CCML (Centaurus Capability Markup Language). However, these frameworks are most of the time domain specific and limited to a set of aspects of the context (e.g., location, environment) [25], [26].

[1] http://www.ist-spice.org

Other projects have been proposed for user and context modeling, such as E2R[2] and MAGNET Beyond[3]. However, they do not consider the specific aspects for multimedia documents, and especially the document structure dimension inside profile.

In the next section, we present step by step how we have built an independent user profile by integrating rich explicit constraints.

# 3 FACETS AND CONSTRAINTS

In order to display a multimedia document on multiple devices, several constraints described in a profile have to be satisfied. A profile is usually composed of some characteristics and some constraints that will be used by an adaptation process in order to compute an adapted document complying with all the specified restrictions.

In this section, we present our Semantic Generic Profile (SGP) and illustrate it with some real life oriented examples.

## 3.1 Facet

A profile should provide information on some devices capabilities, user context and documents characteristics that the target device is able to take into account. Currently, profile descriptions contain description values, such as screen size, preferred language, device model, etc. However, if this profile migrates on different platforms, many profiles characteristics have to be reconfigured. An easy evolving characteristic is for instance a screen resolution or some available codecs. Therefore, to ensure profile portability our profile structure is composed of service descriptions.

A profile is most of the time designed as a hierarchical descriptions and data outputs. Figure 1 illustrates an example of a hierarchy which is composed of services descriptions concerning the context of a specific user.

In Figure 1, each service $s_i$ of the hierarchy is identified by a resource name, e.g., Language, Age, etc. Each potential value $v_j$ corresponds to a potential parameter value of the related service. For instance, the ***Language*** service has two potential values: *French* and *English*. Of course, this means that the user may understand both languages, i.e., French and English.

The whole hierarchy of service descriptions and data values is called a facet. The root node, i.e., the service name on top of the hierarchy, will be the name of the facet. In Figure 1, the name of the facet is “Context”.

A service may provide some data, i.e., it will be in charge of giving information about a current situation. A service may also require some parameters in order to update its status.

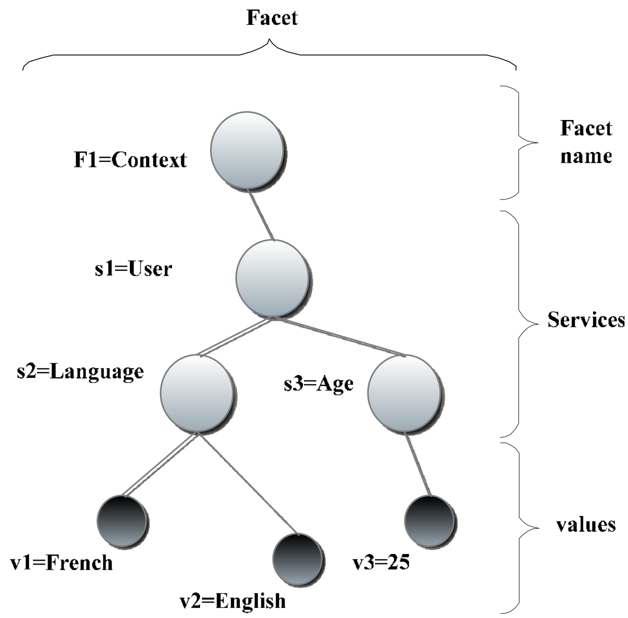


**Figure 1. A facet example describing a user context**

In Figure 2, on one hand, the Battery service provides some data, here 15, meaning 15% of the remaining battery level. On the other hand, the Luminosity service may require a parameter value in order to update the screen luminosity, here 70 means that the luminosity intensity has to be set to 70%.

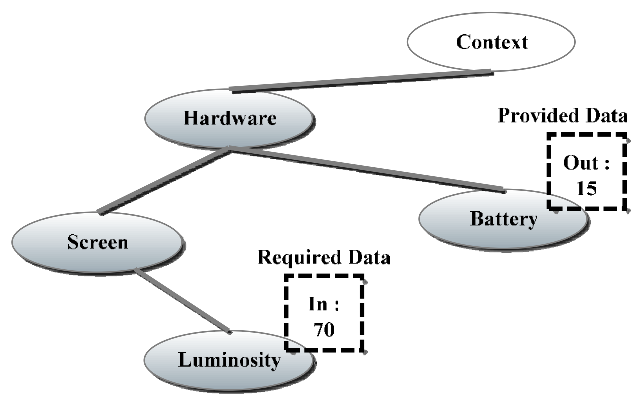


**Figure 2. A service-based profile**

Naturally, as shown in the next section, it may be interesting to consider other facets.

## 3.2 Multi-facets

A profile has to describe different categories of information: (1) Device characteristics (hardware and software), (2) Context information related to the user and the device and (3) Document structures that can be executed by the target device. Consequently, profiles may be composed of several facets.

Figure 3 illustrates the descriptions of three facets that correspond to the three previously mentioned categories. In this figure, the contextual facet describes some user information, such as its neighbors, its location and its preferred languages. The document facet specifies the types of multimedia contents that the device is

[2] https://ict-e3.eu
[3] http://www.neclab.eu/Projects/Magnet.htm

able to execute and particularly video decoding. The hardware facet collects physical and technical characteristics of the device (e.g., RAM available, Battery level, etc.).

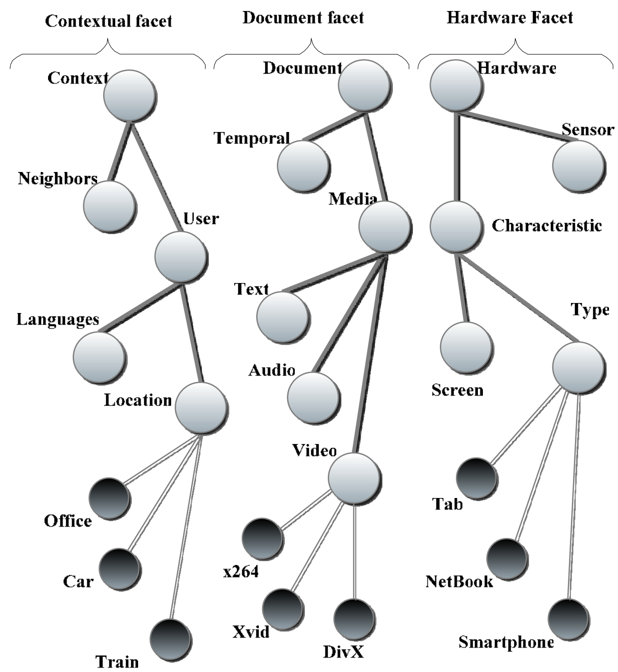


**Figure 3. A multi-facet example**

## 3.3 Constraints

At this point, even if profile information is hierarchically structured, it provides only raw data that have to be analyzed further by adaptation processes in order to deduce implicit constraints. For example, considering that a profile specifies a specific smartphone screen size, the adaptation processes have to deduce if they have to transform the document or not. Moreover, based on the user language information, adaptation processes also have to deduce if they have to translate other media, like texts, sounds, etc.

However, deducing implicit constraints from profiles may lead to adapted multimedia documents that do not comply with the real user needs. For example, it is possible that a user would like a picture in a native resolution display and not the screen resolution display. Furthermore, it is sometimes impossible to deduce some implicit constraints. For instance, adaptation processes cannot deduce what to do if the user is in a specific situation, like in a train. In such a situation, does the user want to allow the execution of videos or not?

To solve such situations, we propose to design explicit constraints. Using facet descriptions, these constraints will associate different categories of profile information.

## 3.4 Explicit constraints

In order to define explicit constraints between facets, we need to specify conditions and actions. For example, if a user is located in his car (*i.e., the condition*), he may not want to see videos (*i.e., the action*).

Hence, we propose to design explicit constraints by associating several facet services to some conditions that have to be satisfied. Obviously, if all conditions are satisfied, an action will be triggered on a facet service.

Figure 4 illustrates the specification of an explicit constraint by using the terms that have been defined above. In this figure, an explicit constraint may involve several potential values of a service. Each value $v_i$ will be checked with a comparator $c_i$. A comparator is a binary relation, such as equal, less than, greater than, etc. The comparator compares a potential value with the current situation value provided by the related service. If all (i.e., AND) or one (i.e., OR) conditions are satisfied, it triggers an action which is associated to a service. Of course, several types of actions may be specified depending on the targeted service.

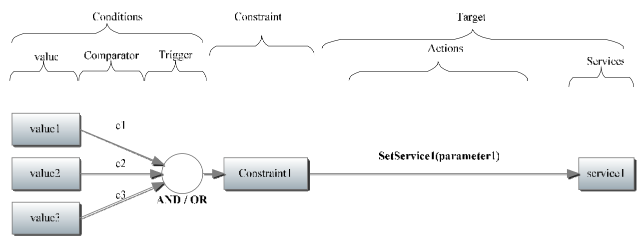


**Figure 4. General scheme of an explicit constraint**

Figure 5 presents a multi-facets profile, which contains an explicit constraint. The contextual facet is composed of information related to the user's location. Let suppose that the user is located in a car, this information will be provided by the method *getUserLocation* associated to the *UserLocation* service. Moreover, the hardware facet of the target device defines some battery power levels, e.g., in the figure *the BatteryLevel* service may provide a remaining battery level of 50%. Finally, the document facet specifies that different sound levels may be set.

As shown in Figure 5, an explicit constraint has been defined (see the arrows in the figure). It specifies that if the user is located in a car and if the battery power is greater than 50%, we should set the audio level of all audio media in a multimedia document to 70 %.

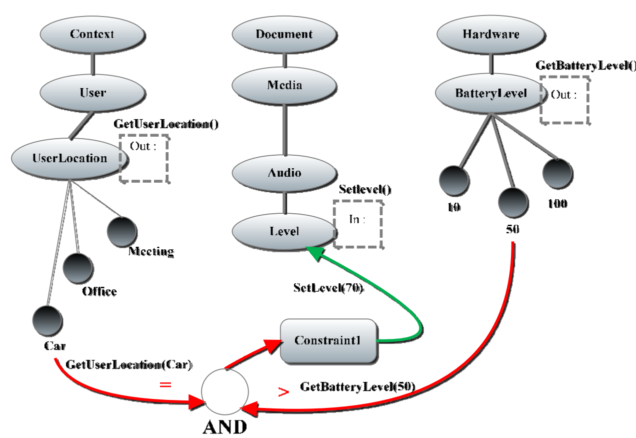


**Figure 5. An explicit *constraint* example**

Naturally, a profile may not contain an explicit constraint. Hence, the profile will only describe information about the three facets: Device characteristics, Context information and Document structures. Each service may be invoked independently in order to collect some data. If no constraints are specified or if none of the conditions are satisfied, any action will be specified for some adaptation processes. In this case, it is up to each process to make implicitly the adaptation.

Formally, an explicit constraint is: Ce=<Sc, St> with Sc a set of conditions and St a set of targets. The set of conditions Sc={C1,...,Cn} is composed of some conditions Ci=<vi,ci> with vi a potential value of a service and ci a comparator. A condition Ci is satisfied if the value of vi complies with ci and the value provided by the related service. An explicit constraint triggers an action on a service if all Ci are valid.

## 3.5 Advanced explicit constraints

A set of conditions may trigger different actions on some services. For instance, Figure 6 shows that actions may be related to several services. Moreover, different actions may be related to the same service (e.g., read text and translate text to French).

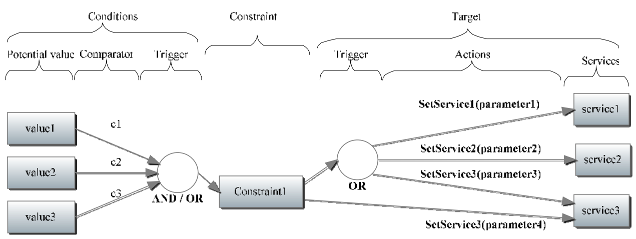


**Figure 6. An explicit *constraint* with multiple actions**

Figure 7 illustrates such an example of an explicit constraint triggering several actions. In this example, if the available CPU power is less than 50%, audio are played and the text content is read, while videos are forbidden/removed. As you may see, for a given condition, we may trigger several actions related to different facet services.

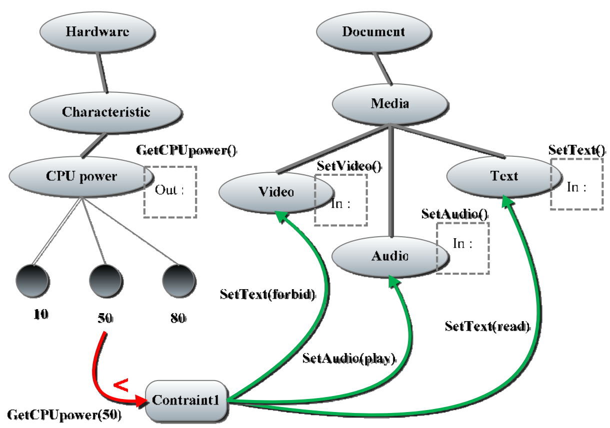


**Figure 7. Explicit constraint triggering several actions**

## 3.6 Priorities inside explicit constraints

Priorities between actions may be defined in explicit constraints by specifying a weight value. This is useful when a constraint triggers alternative actions. Indeed, it indicates a preference between concurrent actions of explicit constraints (Figure 8). Each weight is an integer between 0 and 1. The more the weight value is close to 1 the more the action of the corresponding explicit constraint is important.

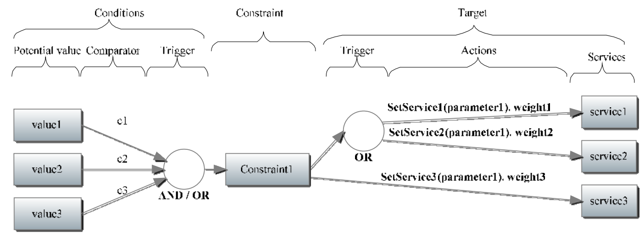


**Figure 8. Priorities *between* actions of explicit constraints**

Figure 9 shows an example that illustrates potential weights on actions. Especially, it specifies the understanding level of a language according to the type of a media.

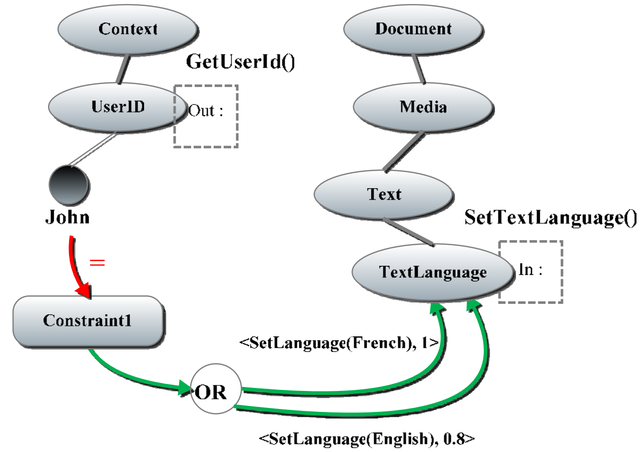


**Figure 9. Priorities *between* actions**

For example, in Figure 9, John is better in French than in English for reading texts. Of course, these weights cannot be set directly by the user. John may specify in advance that he prefers texts written in French than in English, and these weights are then computed automatically during the profile creation.

We have specified facets and constraints. In the next section, we propose to integrate these notions into a global profile.

## 3.7 Profile

A profile is a set of facets that can be enriched by constraints. In this paper, we promote the use of three facets: device characteristics, context information and document structures. As we have shown previously, complex and rich high-level constraints may be specified. For instance, we may compose several conditions with conjunction and disjunction operators and we may trigger several actions with priorities. Moreover, actions parameters may be fixed values or may refer to values provided by the profile services. For example, Figure 10 specifies a constraint that set the image resolution of documents to the same resolution of terminal screen if its resolution is between 800x600 and 1480x1200.

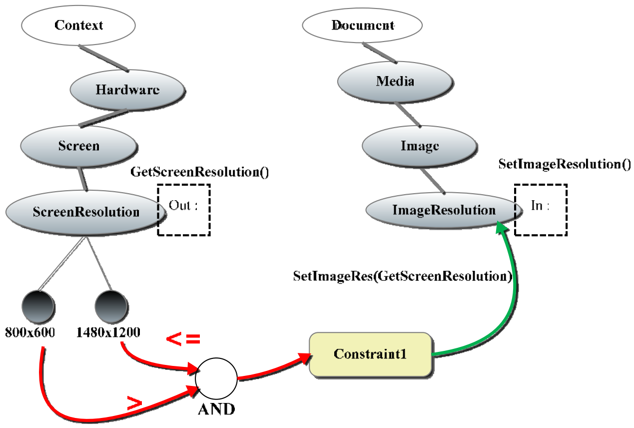


**Figure 10. Multiple conditions constraint example**

In the next section, we model our proposal, entitled SGP, with a UML diagram and we propose an encoding of such profiles in RDF / XML. Then, we propose some query examples in order to show how to exploit our profile structure. Finally, we evaluate query performance on different mobile platforms.

# 4 THE SEMANTIC GENERIC PROFILE

## 4.1 UML modeling of our Semantic Generic Profile (SGP)

Figure 11 presents the meta-model of our Semantic Generic Profile. This meta-model is general enough to suit to various adaptation processes that transform multimedia documents.

In such a meta-model, we identify two categories of information: facets, which consist of a hierarchy of reusable services, and constraints that represent associations between the facets and services. Of course, one may specify new kind of facets.

This model is generic because it allows expressing constraints and information profile regardless of the application using it. Thus, it is not limited to specific functions or applications.

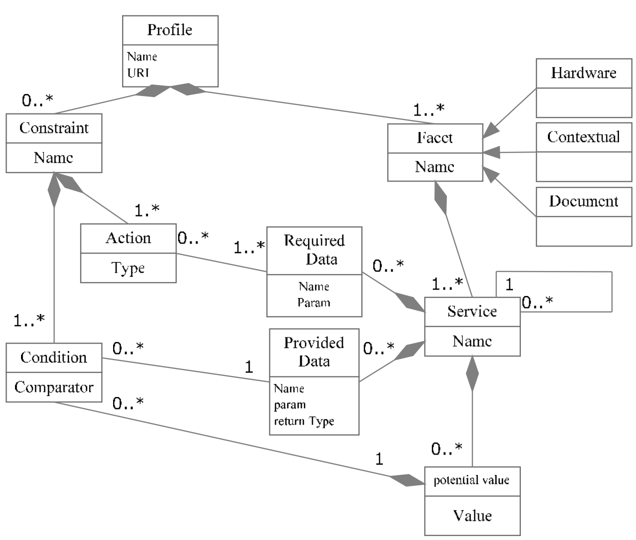


**Figure 11. Our *Semantic Generic Profile* (SGP) model**

In the next section, we propose to encode SGP profiles using RDF/XML. Indeed, it will allow us to query profile descriptions with semantic queries.

Strang et al. [27] analyze six context modeling approaches (i.e., Key-value modeling, markup scheme modeling, object oriented modeling, graphical modeling, logic based modeling and ontology based modeling approaches). Authors have shown that the ontology based context modeling was the most appropriate. In the next section, we encode our SGP profile with an ontology adapted language.

## 4.2 SGP profiles encoded in RDF/XML

RDF (*Resource Description Framework*) is a standard model for data interchange on the Web [10]. Several reasons motivate us to use this formalism: Firstly, RDF allows us to perform aggregations of descriptions, which can be useful if several services describe a profile. Secondly, RDF can handle semantic concepts described in ontologies, thus enhancing the SGP semantics. For instance, semantics allow us to state that "Sensor" is equivalent to "Captor". Hence, a semantic query on "Captor" will also refer to "Sensor". Thirdly, RDF does not force us to express hierarchies of data as defined in other languages (it has a graph-based structure thanks to triples). Finally, other languages and proposals whose objectives are to describe profiles are based on this formalism (e.g., CC/PP or CSCP).

Figure 12 is an example of a SGP profile encoded in RDF/XML. It is composed of a RDF header, some services descriptions (*i.e., Screen, ScreenResolution, ScreenLuminosity, and Battery*) of the hardware facet and an explicit constraint. The *ScreenResolution* service is composed by input (i.e., *SetScreenResolution*) and output (*GetScreenResolution*) functions with parameters (*e.g., string*).

The explicit constraints is composed of the following condition: if the battery level is less than 0.1 (for 10%); and the following action : set the screen luminosity level to the value 0.3 for 30 % .

```
<?xml version="1.0"?>
<rdf:RDF                                    xmlns:sgp="http://SGP#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <sgp:Profile rdf:about="http://SGP#Profil_1">
    <sgp:name>John's profile</sgp:name>
    <sgp:describes>
      <sgp:Facet rdf:about="http://SGP#Hardware">
        <sgp:contains>
          <sgp:Service rdf:about="http://SGP#Screen">
            <sgp:contains>
              <sgp:Service
rdf:about="http://SGP#ScreenResolution">
                <sgp:in>
                  <sgp:InputFunction
rdf:about="http://SGP#SetScreenResolution">
                    <sgp:param
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"/>
                  </sgp:InputFunction>
                </sgp:in>
                <sgp:out>
                  <sgp:OutputFunction
rdf:about="http://SGP#GetScreenResolution">
                    <sgp:return
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"/>
                  </sgp:OutputFunction>
```

```
                         </sgp:out>
                      </sgp:Service>
                    </sgp:contains>
                      <sgp:contains>
<sgp:Service rdf:about="http://SGP#ScreenLuminosity">
                       <sgp:in>
                         <sgp:InputFunction
rdf:about="http://SGP#SetScreenLuminosity">
                            <sgp:param
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"/>
                           </sgp:InputFunction>
                         </sgp:in>
                      </sgp:Service>
                    </sgp:contains>
                  </sgp:Service>
               </sgp:contains>
               <sgp:contains>
                 <sgp:Service rdf:about="http://SGP#Battery">
                    <sgp:out>
                      <sgp:OutputFunction
 rdf:about="http://SGP#GetBatteryLevel">
                         <sgp:return
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"/>
                       </sgp:OutputFunction>
                    </sgp:out>
                 </sgp:Service>
              </sgp:contains>
           </sgp:Facet>
        </sgp:describes>
        <sgp:handles>
           <sgp:Constraint rdf:about="http://SGP#C1">
              <sgp:contains>
                 <sgp:Condition>
                   <sgp:on  rdf:resource=http://SGP#GetBatteryLevel
 />
                   <sgp:comparator>&lt;</sgp:comparator>
                   <sgp:value>0,1</sgp:value>
                 </sgp:Condition>
              </sgp:contains>
              <sgp:trigger>
                <sgp:Action>
                  <sgp:over
rdf:resource="http://SGP#SetScreenLuminosity" />
                  <sgp:param>0,3</sgp:param>
                </sgp:Action>
              </sgp:trigger>
           </sgp:Constraint>
        </sgp:handles>
</sgp:Profile>
```

**Figure 12. An SGP profile encoded in RDF/XML**

In the next section, we briefly introduce some query examples that may be used to retrieve some information contained in our profile SGP structure.

## 4.3 SPARQL Queries

SPARQL is a RDF query oriented language able to retrieve and manipulate data stored in RDF descriptions. It was made a standard by the RDF Data Access Working Group (DAWG) of the W3C, and considered as one of the key technologies of the Semantic Web.

In the following, we present several SPARQL queries as examples that may be specified in order to retrieve important information contained in a SGP profile. Note that these queries may be used by adaptation processes in order to extract explicit constraints.

R1 is the most simple query: it return triples

```
PREFIX sgp: <http://SGP#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT * WHERE {?x ?y ?z .}).
```

*Query R1*. List all triples

The query R2 returns a list containing all SGP services.

```
PREFIX sgp: <http://SGP#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT *our SGP.
WHERE {
        ?S rdf:type sgp:Service .
}
```

*Query R2*. SGP services list query

The query R3 returns a list containing all services that provide data.

```
PREFIX sgp: <http://SGP#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT *
WHERE {
        ?S rdf:type sgp:Service .
        ?S sgp:out ?F .
        ?F rdf:type sgp:OutputFunction .
}
```

*Query R3*. List of services that could provide some data

The query R4 returns actions list with required data. For example: *SetImageResolution(400x600), SetScreenLuminosityLevel(0.3)*

```
PREFIX sgp: <http://SGP#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT *
WHERE {
        ?S rdf:type sgp:Service .
        ?S sgp:in ?F .
        ?F rdf:type sgp:InputFunction .
}
```

*Query R4*. Actions list query

The query R5 returns the complete hierarchy of services.

```
PREFIX sgp: <http://SGP#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT * WHERE {
        ?S1 rdf:type sgp:Service .
        OPTIONAL {
                ?S2 rdf:type sgp:Service .
                ?S1 sgp:contains ?S2 .
        }
}
```

*Query R5*. Services hierarchy.

We evaluate whether it is possible to execute those queries on different mobile devices. In the xnext section, we detailthedetail the tests that we have performed on two different platforms.

### 4.4 Experimentation

Using the JENA library, we have tested the performance of several SPARQL queries on some SGP profiles encoded in RDF/XML. We have performed these experiments on two heterogeneous mobile configurations.

Configuration 1: A Laptop running Windows 7 (x64) with 6GB of RAM and i7-2630QM quadruple core processor (2 GHz).

Configuration 2 : A Samsung Galaxy Tab running Android 3.2 with 1 GB of RAM and a double core Tegra 2 processor (1 GHz).

In Figure 13, we compare the execution time of queries on both platforms. We have also performed repeated loops (10, 50, 100, 500, 1000) of query 1 (R1), query 2 (R2) and a five different query sequences (R1, R2, R3, R4, R5).

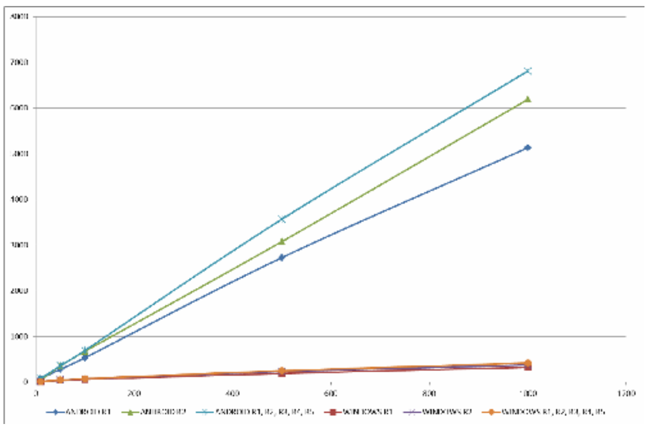

**Figure 13. Query execution time comparison**

We find that query execution on SGP profiles is on average 14 times slower on the first platform to the second. However, query execution time on Android remains below 8 ms against under 1.6 ms on Windows. The difference between two requests is up to double in time, thus, we must create a strategy to optimize performance querying.

In Figure 14, we compare the execution time of 100 R1 queries on different profiles composed by 5, 10, 20, 30, 50 and 100 triples.

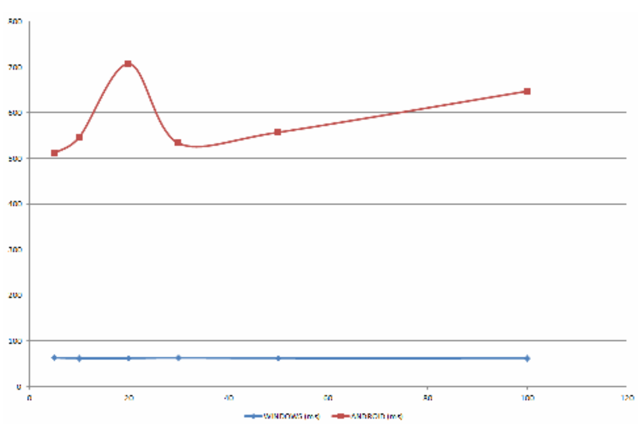

**Figure 14. Query execution time based on the size of a profile**

Our objective in figure 14 is to identify performance differences between profiles with different sizes.

We note that the query execution is nine times faster in conf1. However, the profile size seems to have a low impact on processing time.

Those experiments allow us to consider the processing of queries on configurations 1 and 2 in order to check performances. We find that the profile size (triplet) has low impacts on the query execution performance. We want to continue the evaluation of queries on profiles, and especially semantic queries, in order to design an optimized strategy.

## 5 FUTURE RESEARCH DIRECTIONS

In order to concretely exploit our SGP profile descriptions, it is necessary to use a software layer which is able to (1) integrate and complete the profile and verify the constraints, (2) provide information for different adaptation processes.

Figure 15 illustrates how our SGP proposal may be included into an adaptation use case. More precisely, from an initial multimedia document and our profile, the software layer will indicate to an adaptation process a set of transformations that have to be executed in order to provide an adapted document.

Furthermore, we will enhance our SGP structure with semantic information by specifying an SGP ontology. Obviously, our RDF/XML descriptions will use the vocabulary defined in such an ontology. Moreover, thanks to this ontology, it will enable the integration of other profile descriptions, e.g., CC/PP, UaProf.

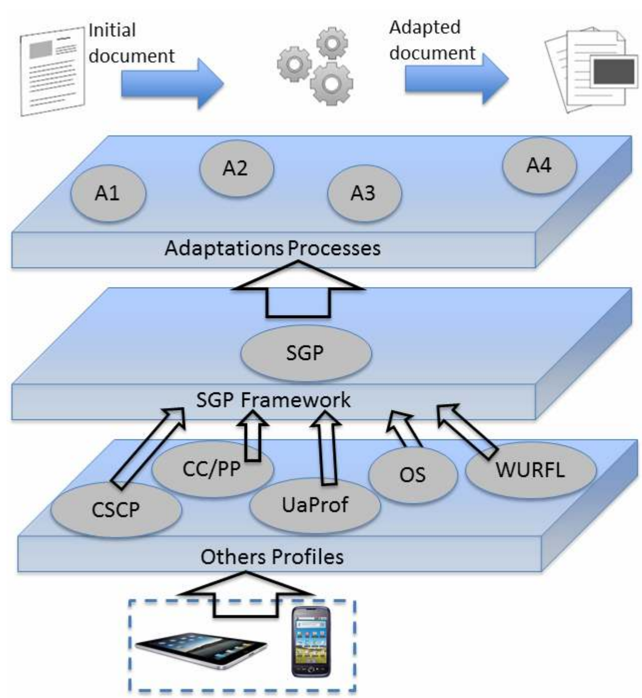


**Figure 15. SGP Framework positioning**

## 6 CONCLUSION

We have defined a semantic generic profile (SGP) which organizes profile information into facets. In this paper, we have considered three kind of facets which are related to the device characteristics, the context information and the document structure. Moreover, we have proposed to link these profile information with the specification of high-level explicit constraints. These constraints enable to model different types of actions under rich conditions. Naturally, the main objective of our profile structure is to better guide the adaptation process in order to compute valid adapted multimedia documents.

In the future, we plan to develop a global framework that will use our SGP profile structure. Firstly, this framework will exploit the profile semantics in order to exploit other information which are contained into other types of profile, such as CC/PP, CSCP, etc. Furthermore, we will develop some efficient methods that will compare some initial multimedia documents with our profile structures. Finally, we will experiment in a real adaptation architecture the benefits of using our profile descriptions. For instance, we will evaluate the user feedback on the adaptation of multimedia documents which have been made on different platforms with and without our profile structure.

## ACKNOWLEDGMENT

This work has been partially supported by the ANR MOANO Project "Models and tools for mobile applications discovery of territory" [28].

## REFERENCES

[1] D. Bulterman, J. Jack, C. Pablo, M. Sjoerd, H. Eric, D. Marisa, Q. Julien, K. Hiroshi, W. Daniel, G. P. Xabiel, M. David, C.-L. Samuel, H. Marcin, Z. Daniel F., M. Thierry and W3C, "Synchronized Multimedia Integration Language (SMIL 3.0), W3C Recommendation," december 2008. [Online]. Available: http://www.w3.org/TR/SMIL/.

[2] S. Laborie, J. Euzenat and N. Layaïda, "Semantic Adaptation of Multimedia Documents," *Multimedia Tools and Applications,* vol. 55, no. 3, pp. 379-398, December 2011.

[3] T. Lemlouma and N. Layaďda, "Content Adaptation and Generation Principles for Heterogeneous Clients," in *In W3C Workshop on Device Independent Authoring Techniques*, St. Leon-Rot, Germany, September 2002.

[4] M. Asadi and J.-C. Dufourd, "Context-aware Semantic Adaptation of Multimedia Presentations," in *Multimedia and Expo, ICME 2005. IEEE International Conference on*, Amsterdam, , The Netherlands, 2005.

[5] T. Lemlouma and N. Layaïda, "NAC: A Basic Core for the Adaptation and Negotiation of Multimedia Services," Opera Project, INRIA, September 2001.

[6] S. Laplace, M. Dalmau and P. Roose, "Kalinahia: Considering quality of service to design and execute distributed multimedia applications," in *IEEE/IFIP Int'l Conference on Network Management and Management Symposium*, Salvador de Bahia, Brazil , 2008.

[7] B. Girma, L. Brunie and J.-M. Pierson, "Planning-Based Multimedia Adaptation Services Composition for Pervasive Computing," in *2nd International Conference On Signal-Image Technology and Internet-based Systems (ACM/IEEE SITIS'2006)*, Hammamet, Tunisia, 17-21 December 2006.

[8] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber and L. Tanca, "A Dataoriented Survey of Context Models," Milano.

[9] S. Forough and J. Reza, "A Comparative Study of Context Modeling Approches and Applying in an Infrustructure," *Canadian Journal on Data Information and Knowledge Engineering,* vol. 3, no. 1, 2012.

[10] G. Klyne and J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax," february 2004. [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[11] M. Viviani, N. Bennani and E. Egyed-Zsigmond, "A Survey on User Modeling in Multi-Application Environments," in *The Third International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services CENTRIC'10*, Nice, France, 2010.

[12] G. Klyne, F. Reynolds, C. Woodrow, O. Hidetaka, J. Hjelm, M. H. Butler and L. Tran, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0," january 2004. [Online]. Available: http://www.w3.org/Mobile/CCPP/.

[13] J. Indulska, R. Robinson, A. Rakotonirainy and K. Henricksen, "Experiences in Using CC/PP in Context-Aware Systems," in *Mobile Data Management 4th International Conference (MDM '03)*, Melbourne, 2003.

[14] B. Girma, B. Lionel and P. Jean-Marc, "Content adaptation in distributed," *Journal of Digital Information Management,* vol. 3, no. 2, pp. 95-100, 2005.

[15] W. Forum, "WAG UAProf, Wireless Application Protocol," 2001. [Online]. Available: http://www.wapforum.org.

[16] S. Buchholz, T. Hamann and G. Hübsch, "Comprehensive Structured Context Profiles (CSCP): Design and Experiences," in *PERCOMW '04 Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Orlando, Florida, 2004.

[17] C. Tarak, E. Dejene, L. Frédérique and S. Vasile-Marian, "A comprehensive approach to model and use context for adapting applications in pervasive environments," *Journal of Systems and Software,* vol. 80, no. 12, p. 1973–1992, 2007.

[18] C. Bolchini, C. Curino, E. Quintarelli, F. Schreiber and L. Tanca, "Context-ADDICT. Technical Report," in *Politecnico di Milano*, Milano, 2006.

[19] C. Bolchini, G. Orsi, E. Quintarelli, A. Schreiber and L. Tanca, "Context Modeling and Context Awareness: steps forward in the Context-ADDICT project," *IEEE Data Eng. Bull.,* vol. 34, no. 2, pp. 47-54, 2011.

[20] M. Chevalier, C. Soulé-Dupuy and P. Tchienehom, "Semantics-based Profiles Modeling and Matching for Resources Access," *Journal des Sciences pour l'Ingénieur, African Journals Online (AJOL),* vol. 1, no. 7, pp. 54-63, 2006.

[21] F. Jouanot, N. Cullot and K. Yétongnon, "Context Comparison for Object Fusion," *Lecture Notes in Computer Science,* vol. 2681, p. 1031, 2003.

[22] L. Passani, "Wurfl (Wireless Universal Resource File),"

2007. [Online]. Available: http://wurfl.sourceforge.net.

[23] C. Veaceslav, "Applying Next Generation Web Technologies in the Configuration of Customer Designed Products," Royal Institute of Technology, Stockholm, 2011.

[24] A. Ferscha, M. Hechinger, A. Riener, H. Schmitzberger, M. Franz, M. Rocha and A. Zeidler, "Context-Aware Profiles," in *Autonomic and Autonomous Systems, ICAS '06*, 2006.

[25] L. Kagal, V. Korolev, H. Chen, J. Anupam and T. Finin, "Centaurus : a framework for intelligent services in a mobile environment," in *The 21st International conference on distributed computing system workshop*, Mesa, 2001.

[26] L. Kagal, V. Korolev, S. Avancha, A. Joshi, T. Finin and Y. Yesha, "Centaurus: An Infrastructure for Service Management in Ubiquitous Computing Environments," *Wireless Networks,* vol. 1, 1995.

[27] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," in *Proceedings of the First International Workshop on Advanced, Sixth International Conference on UbiComp2004*, Nottingham, England, 2004.

[28] T2i, "Models & Tools for Pervasive Applications focusing on Territory Discovery (MOANO)," [Online]. Available: http://moano.liuppa.univ-pau.fr.