
Profil générique sémantique pour l'adaptation de documents multimédias

Cédric Dromzée^{1,2}, Sébastien Laborie¹, Philippe Roose¹

¹LIUPPA/T2I - Université de Pau et des Pays de l'Adour
2, Allée du Parc Montauray
F-64600 Anglet
prenom.nom@iutbayonne.univ-pau.fr

²AEXIUM SAS
29 rue des Pins
F-40230 Saubion
dromzee@aexium.com

RÉSUMÉ. Actuellement, les documents multimédias peuvent être consultés à tout moment et n'importe où sur une grande variété de dispositifs mobiles. L'hétérogénéité de ces plateformes, les préférences utilisateurs mais également le contexte de consultation impose des adaptations de documents à certaines contraintes, comme par exemple, ne pas jouer de contenus audio lorsque l'utilisateur participe à une réunion. Actuellement, les langages de modélisation de contexte ne traitent pas ce type de contraintes. Ils listent généralement de multiples valeurs d'informations qui doivent être interprétées par les processus d'adaptation afin d'en déduire implicitement des contraintes. Notre contribution propose une nouvelle approche de modélisation du contexte où les informations sont liées par des contraintes explicites riches. Afin de valider cette proposition, nous avons utilisé des technologies du Web sémantique en spécifiant des profils en RDF et en expérimentant leur utilisation sur plusieurs plateformes.

ABSTRACT. Currently, multimedia documents can be accessed at anytime and anywhere with a wide variety of mobile devices. 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 constraint. 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.

MOTS-CLÉS : Modélisation du contexte, profil, documents multimédias, contraintes de l'utilisateur et des dispositifs.

KEYWORDS : Context modeling, profile, multimedia documents, devices and users constraints.

1 Introduction

De nos jours, de nombreux documents multimédias peuvent être créés et consultés par les utilisateurs. Ces documents peuvent être composés de différents types de contenus, tels que des vidéos, des contenus sonores, des textes et des images. Des exemples typiques de documents multimédias sont les pages Web ou les présentations SMIL (Bulterman et al., 2008). Dans ces documents, les contenus multimédias sont généralement organisés dans un espace de présentation et synchronisés dans le temps. L'utilisateur a en outre la possibilité d'interagir avec la présentation, par exemple, lorsqu'il clique sur une image, une vidéo est présentée.

Dans le même temps, de nombreux dispositifs mobiles, comme les ordinateurs portables, les tablettes ou les smartphones, sont capables d'afficher ces documents multimédias. Ces accès multiples permettent aux utilisateurs de consulter et d'interagir sur des documents à tout moment et de n'importe où. Toutefois, ces appareils disposent de capacités hétérogènes : des caractéristiques matérielles variées (*taille écran, batterie...*) mais également des capacités logicielles différentes (*codecs lus...*). De plus, les préférences de l'utilisateur ou leurs handicaps peuvent contraindre l'exécution de certains contenus. Par exemple, un utilisateur souhaite lire des textes exclusivement en français et/ou éviter la lecture de contenus sonores lorsqu'il participe à une réunion. Toutes ces restrictions induisent des contraintes, parfois très riches, qui peuvent être spécifiées dans un profil.

Au sein d'un profil, différentes catégories d'informations doivent être gérées : (1) les caractéristiques du terminal (*matériel et logiciel*) et son état courant (*i.e batterie et/ou bande passante faible...*), (2) des informations de contexte liées, par exemple, à des interactions entre l'utilisateur et l'appareil, telles que les préférences de langues ou le niveau sonore ambiant, et (3) les informations liées aux constituants des documents, comme les types de contenu qui peuvent être joués ou bien les possibles organisations ou restrictions de présentation (*mise en page, synchronisation des contenus multimédias...*). Par conséquent, si un document multimédia n'est pas conforme à certaines contraintes spécifiées dans un profil, ce document ne sera pas exécuté correctement sur le terminal cible. Ainsi, pour que les documents multimédias soient accessibles sur de multiples plateformes hétérogènes, ils doivent être adaptés. C'est-à-dire qu'ils doivent être transformés pour satisfaire toutes les contraintes des plateformes cibles.

Depuis plusieurs années, de nombreux travaux ont été proposés pour adapter les documents multimédias, comme dans (Lemlouma et al., 2002) (Kimiaei-Asadi et al., 2005) et (Laborie et al., 2008). En partant d'un profil cible, ces approches combinent des opérateurs de modification du document : **transformation** (*e.g., résumé ou traduction d'un contenu*), **transcodage** (*e.g., modification d'une vidéo encodée en AVI en MPEG*) et **transmodage** (*e.g., modification d'un contenu textuel en contenu sonore*). La richesse du profil cible est exploitée pour déterminer un enchaînement de ces opérateurs (Lemlouma et al., 2001), voir leur déploiement (Laplace et al.,

2008) sur différentes plateformes pour économiser des ressources, telles que la batterie.

Néanmoins, chacune de ces approches exploite son propre format de profil qui se compose habituellement de listes de valeurs, telles que la résolution de l'écran ou la puissance de la batterie. En conséquence, un processus d'adaptation doit interpréter ces profils et en déduire implicitement certaines contraintes de plus haut niveau. Par exemple, *si la batterie est inférieure à 10%, éviter de jouer des vidéos en haute qualité*. Evidemment, dans cette situation, chaque processus d'adaptation peut en déduire ses propres contraintes, sachant qu'elles peuvent être différentes d'un processus à un autre, voire être en contradiction avec les souhaits de l'utilisateur. De plus, seules les contraintes prévues par un processus d'adaptation seront prises en compte. Nous avons également constaté dans la littérature que les langages actuels de modélisation de contexte, comme par exemple, tous ceux présentés dans (Bolchini et al., 2007) et (Forough et al., 2012), ne tiennent pas compte de ce type de contraintes de haut niveau, alors qu'elles pourraient être très utiles pour guider les processus d'adaptation. Par conséquent, toutes ces approches n'ont donc pas de vision générique du profil et surtout de la description des contraintes à retenir pour l'adaptation. Il sera donc difficile, voire impossible, qu'un processus d'adaptation puisse exploiter différents types de profils.

Dans cet article, nous proposons de dépasser cette limitation en définissant un nouveau modèle de description de profils où (1) les informations du profil sont organisées en facettes (c'est-à-dire, *caractéristiques de la plateforme, informations sur le contexte et sur les documents*) et composées de services qui fournissent des données ou bien requièrent des modifications, et (2) les informations du profil sont liées et enrichies par des contraintes explicites de haut niveau. Grâce à cette proposition, les profils peuvent migrer d'une plateforme à une autre tout en préservant les contraintes spécifiées, assurant ainsi l'interopérabilité. Afin de valider notre proposition, nous encodons nos profils en RDF/XML (Klyne et al., 2004) et nous évaluons plusieurs exécutions de requêtes sur différentes plateformes. Les résultats expérimentaux confirment que les processus d'adaptation peuvent accéder à nos descriptions de profil de manière efficace.

Cet article est organisé comme suit. Tout d'abord, la section état de l'art donne un aperçu des approches existantes de modélisation des profils et notamment des contraintes. Ensuite, nous détaillons les spécifications de notre profil basé sur la notion de services et nous les illustrons à l'aide d'exemples concrets. Ensuite, nous proposons notre modèle de profil générique, nommé Semantic Generic Profile (SGP). Certaines requêtes qui peuvent être utilisées par des processus d'adaptation sont proposées afin d'illustrer les possibilités d'exploitation de nos descriptions de profils. Des expérimentations ont également été menées pour mesurer l'efficacité de ces requêtes sur différentes plateformes. Enfin, dans la dernière partie de cet article, nous présentons quelques travaux futurs.

2 Etat de l'art

Depuis plusieurs années, de nombreux travaux ont été proposés afin de modéliser au sein de profils les caractéristiques des terminaux et les contextes liés aux utilisateurs (Bolchini et al., 2007) (Forough et al., 2012). Nous avons constaté que de nombreuses approches offrent une vue exclusivement descriptive des informations contenues dans les profils, tandis que d'autres proposent des améliorations avec quelques expressions limitées de contraintes. Dans cette section, nous présentons un aperçu de ces approches.

2.1.1 CC/PP (*Composite Capability / Preference Profiles*)

CC/PP (Klyne et al., 2004) est une recommandation du W3C pour spécifier les capacités des terminaux ainsi que les préférences des utilisateurs selon le formalisme RDF. La structure de tel profil est très descriptive car elle liste des ensembles de valeurs, tels que la taille de l'écran, la version du navigateur, etc. Toutefois, CC/PP manque de structuration, en effet ce langage limite la description de profils complexes notamment en contraignant une hiérarchie stricte à deux niveaux. De plus, ce langage ne permet pas la description de relations et de contraintes complexes entre les informations de contexte. Enfin, pour intégrer de nouveaux éléments, il est nécessaire d'étendre le vocabulaire de CC/PP.

2.1.2 UAProf (*User Agent Profile*)

UAProf (WAP Forum, 2001) constitue une extension du langage CC/PP pour téléphones mobiles. Les éléments de son vocabulaire sont utilisés dans la même structure de base que celle employée pour CC/PP, néanmoins UAProf permet une description précise des capacités des dispositifs sans fil. Entre autres, il décrit des éléments tels que la taille de l'écran, les capacités multimédias, de caractères Unicode... UAProf est un standard exploité par des milliers de téléphones mobiles mais ce langage se limite exclusivement à la description de caractéristiques matérielles de téléphones sans fils.

2.1.3 CSCP (*Comprehensive Structured Context Profiles*)

CSCP (Buchholz et al., 2004) se base également sur le formalisme RDF. Ce langage permet de décrire la localisation, les caractéristiques du réseau et les dépendances des applications. Il présente également une structure multi-niveaux qui est extensible. Contrairement à CC/PP, il permet de spécifier une description du contexte non limitée à deux niveaux hiérarchiques. Toutefois, il ne s'agit pas d'un standard international et cette proposition ne permet pas de décrire des relations ou des dépendances entre les informations du profil. Néanmoins, CSCP modélise des contraintes mais de très bas niveau, telles que certaines informations du profil ont des valeurs différentes selon certaines conditions. Malgré cette extension, ce langage reste non intuitif et difficile à utiliser pour décrire des informations complexes (Indulska et al., 2003).

2.2 *Modèle générique de profils pour la personnalisation de l'accès à l'information*

Les travaux présentés dans (Chevalier, 2006) proposent un modèle générique de profil spécifié en UML qui permet de décrire la structure et la sémantique générale de tout type de profil d'information ou d'utilisateur. Cette contribution définit entre autres des liens sémantiques entre éléments et intègre la pondération des éléments. Ce graphe sémantique est décrit avec une approche orientée logique de description (Jouanot et al., 2003) via les formalismes RDF/RDFS/OWL. Toutefois, ce modèle ne permet pas d'exprimer des actions (couper le son).

2.3 *WURFL (Wireless Universal Resource File)*

WURFL (Passani, 2007) est un fichier XML de description de ressources de terminaux mobiles. Ce langage contient des informations sur les capacités et fonctionnalités des dispositifs mobiles avec plus de 500 « capabilities » pour chaque périphérique (répartis en 30 catégories). Ce projet se destine à l'adaptation de pages web sur terminaux mobiles (Veaceslav, 2011).

2.4 *Autres approches de modélisation de profils*

D'autres approches de modélisation existent, telles que PDDL (Pervasive Profile Description Language) (Fersha, 2006) ou CCML (Centaurus Capability Markup Language). Néanmoins, elles sont la plupart de temps spécifiques à un domaine et limitées à un ensemble d'aspects du contexte (localisation, environnement, etc.) (Kagal et al., 1995) (Kagal et al., 2001).

3 Facettes et contraintes

Pour qu'un document puisse être exécuté sur de multiples plateformes, différentes informations et contraintes exprimées dans les profils doivent pouvoir être exploitées. Comme nous allons le voir dans cette section, nous proposons de structurer un profil à l'aide de facettes ainsi que de contraintes qui vont guider les processus d'adaptation pour fournir des documents multimédias adaptés répondant à toutes les restrictions spécifiées. Dans ce qui suit, nous présentons par construction comment nous avons spécifié notre modèle en l'illustrant de quelques exemples.

3.1 *Facette*

Un profil doit fournir des informations sur les capacités des plateformes, le contexte de l'utilisateur et les caractéristiques des documents que le terminal est en mesure de prendre en compte. Actuellement, les descriptions d'un profil se composent de valeurs, telles que la taille de l'écran, la langue préférée, le modèle du terminal, etc. Toutefois, si ce profil migre sur différentes plateformes, de nombreuses caractéristiques des profils devront être reconfigurées, comme par

exemple, la résolution de l'écran ou les codecs disponibles. Ainsi, pour assurer sa portabilité, notre structure de profil se compose de descriptions de services.

Un profil est la plupart du temps conçu comme une hiérarchie de descriptions et de valeurs de sortie. La figure 1 illustre des exemples de hiérarchies qui se composent de descriptions de services. Dans cette figure, chaque service si de la hiérarchie est identifié par un nom de ressource, comme par exemple, la langue, l'âge, etc. Chaque valeur v_j correspond à une valeur potentielle renvoyée par ce service.

La hiérarchie des descriptions de service et ses valeurs est appelée une facette. Le nœud racine, soit le nom de service au sommet de la hiérarchie, est le nom de la facette. Dans la figure 1 première colonne, le nom de la facette est «Contexte».

Un profil doit pouvoir décrire différentes catégories d'informations, telles que (1) des caractéristiques de la plateforme (matérielles et logicielles), (2) l'environnement de l'utilisateur et (3) les aspects documentaires. Ainsi, un profil doit être composé de plusieurs facettes.

La figure 1 illustre des descriptions de trois facettes qui correspondent aux trois catégories mentionnées précédemment. Dans cette figure, la facette contextuelle décrit quelques informations sur l'utilisateur, telles que son voisinage, sa position et ses préférences de langues. La facette document spécifie les types de contenus multimédias que le dispositif est capable d'exécuter, et en particulier, le décodage vidéo. La facette liée au matériel recueille les caractéristiques physiques et techniques de l'appareil (mémoire vive disponible, niveau de batterie, etc.)

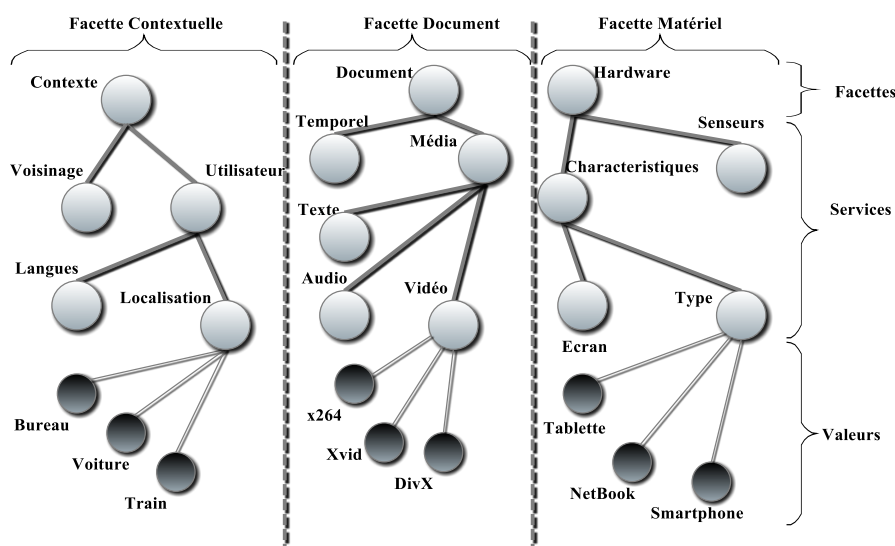


Figure 1. Exemple de facettes

Un service peut fournir des données de sortie (i.e., fonction « get »), c'est-à-dire qu'il sera en charge de procurer des informations sur la situation en cours. Un service peut également mettre à jour son statut en prenant des valeurs en paramètre (i.e., fonction « set »).

Dans la figure 2, d'une part, le service « batterie » fournit en donnée de sortie la valeur 15, ce qui signifie que le niveau de batterie restante est à 15%. D'autre part, le service « luminosité » prend en valeur de paramètre 70, ce qui signifie que l'intensité lumineuse doit être réglée à 70%.

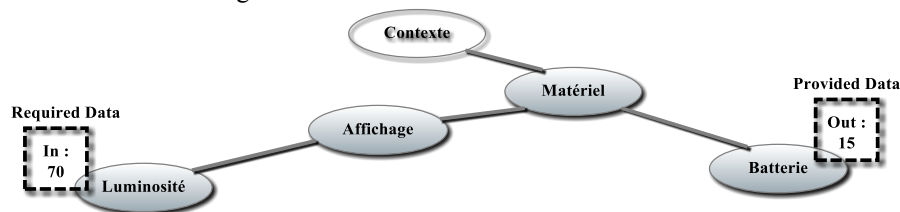


Figure 2. Un exemple de profil composé de services

Cependant, il peut être intéressant de considérer d'autres aspects que nous allons détailler dans les sections suivantes.

3.2 Contraintes

À ce stade, même si les informations de profil sont structurées hiérarchiquement, elles ne décrivent que des informations brutes devant être analysées par un processus d'adaptation qui devra déduire implicitement des restrictions. Par exemple, supposons qu'un profil spécifie la taille d'écran d'un Smartphone, les processus d'adaptation doivent déterminer si ils transforment le document ou non (par exemple, modification de la taille des caractères du texte). Aussi, à partir de la langue de l'utilisateur, les processus doivent également en déduire si ils doivent traduire tous les médias ou seulement certains, tels que le texte, l'audio, etc.

Toutefois, déduire implicitement des contraintes à partir d'informations de profil peut amener à des documents adaptés qui ne sont pas conformes aux besoins de l'utilisateur. Par exemple, il est possible qu'un utilisateur souhaite que les images soient affichées dans leurs résolutions natives et non qu'elles soient adaptées à la résolution de l'écran comme pourrait le décider implicitement un processus d'adaptation. De plus, il est parfois impossible de déduire des contraintes implicites. Par exemple, si l'utilisateur est dans une situation spécifique, telle que participer à une réunion, les processus d'adaptation ne peuvent pas déduire implicitement ce qu'il faut faire lorsque l'utilisateur lance une vidéo : faut-il l'exécuter ou non ?

Pour résoudre de telles situations, nous proposons dans ce qui suit de modéliser des contraintes explicites. À partir des descriptions de facettes, ces contraintes vont associer différentes catégories d'informations du profil.

3.3 Contraintes explicites

Afin de définir des contraintes explicites entre les facettes, nous avons besoin de spécifier des conditions et des actions. Par exemple, si un utilisateur se trouve dans sa voiture (la condition), il peut ne pas vouloir voir les vidéos (l'action). Ainsi, nous proposons de concevoir des contraintes explicites en associant les services de plusieurs facettes à certaines conditions qui doivent être satisfaites. Si toutes les conditions sont remplies, une action sera déclenchée sur un service.

Dans le schéma de la figure 3, nous proposons une modélisation des contraintes explicites en utilisant les termes définis ci-dessus. Dans cette figure, une contrainte explicite peut se composer d'une conjonction ou disjonction de conditions. Chaque valeur v_i sera vérifiée avec un comparateur c_i . Ce comparateur est une relation d'ordre (ou relation binaire) telle que, plus petit ou égal, plus grand ou égal, égal, etc. Il compare la valeur potentielle avec la valeur de la situation en cours, elle-même fournie par le service correspondant. Si toutes (ET) ou une (OU) conditions sont remplies, une action associée à un service sera déclenchée. Bien évidemment, plusieurs types d'actions peuvent être spécifiés en fonction du service ciblé.

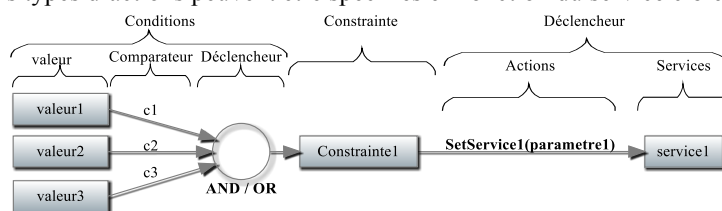


Figure 3. Illustration générale d'une contrainte

L'exemple de la figure 4 présente un profil multi-facettes composé d'une contrainte explicite. La facette contextuelle est composée d'informations relatives à l'emplacement de l'utilisateur. Supposons que l'utilisateur se trouve dans une voiture, cette information sera fournie par la méthode « getLocation » qui est associée au service « UserLocation ». La facette matérielle de l'appareil cible définit certains niveaux de puissance de la batterie, par exemple, le service « BatteryLevel » peut indiquer que le niveau de batterie restante est à 50%. La facette document précise que différents niveaux sonores peuvent être paramétrés. La contrainte explicite « contrainte1 » précise que si l'utilisateur se trouve dans une voiture et que le niveau de la batterie est supérieur à 50%, le niveau sonore de tous les médias audio d'un document multimédia doit être fixé à 70%.

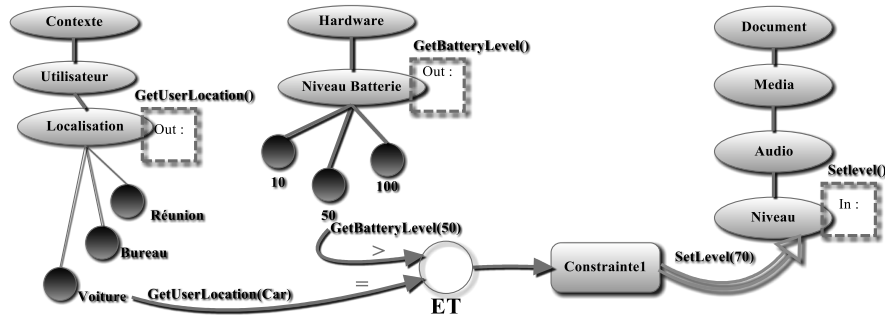


Figure 4. Exemple de contrainte explicite

Naturellement, un profil peut ne pas contenir de contraintes explicites. Ainsi, les profils ne décrivent que des informations sur les trois facettes : caractéristiques de l'appareil, des informations sur le contexte et les structures de documents. De ce fait, chaque service peut être invoqué indépendamment afin de collecter des données. Si aucune contrainte n'est exprimée ou si toutes les conditions ne sont pas vérifiées, aucune action ne sera précisée au processus d'adaptation. Dans ce cas, c'est à lui de décider implicitement des adaptations à réaliser.

Formellement, la règle décrivant une contrainte explicite se décrit de la manière suivante : $Ce = \langle Sc, Sa \rangle$ avec Sc un ensemble de conditions et Sa un ensemble d'actions. L'ensemble des conditions $Sc = \{C1, \dots, Cn\}$ est composé de certaines conditions $Ci = \langle vi, ci \rangle$ avec vi une valeur potentielle d'un service et un comparateur ci . Un Ci condition est rempli si la valeur de vi est conforme à la valeur fournie par le service correspondant et au comparateur ci . Une contrainte explicite déclenche l'ensemble des actions Sa si tous les Ci sont valables.

3.4 Contraintes explicites avancées

Un ensemble de conditions peut déclencher différentes actions sur certains services. Par exemple, la figure 5 montre que les actions peuvent être liées à plusieurs services. Aussi, différentes actions peuvent être liées à un même service (par exemple, lire un texte et traduire un texte en français). Des priorités entre les actions peuvent être définies dans des contraintes explicites en spécifiant une valeur de poids. Ceci est utile lorsqu'une contrainte déclenche des actions alternatives. En effet, il permet d'indiquer une préférence entre des actions concurrentes dans une même contrainte explicite. Chaque poids est un nombre réel compris entre 0 et 1. Plus la valeur du poids est proche de 1 plus l'action de la contrainte explicite correspondante est importante.

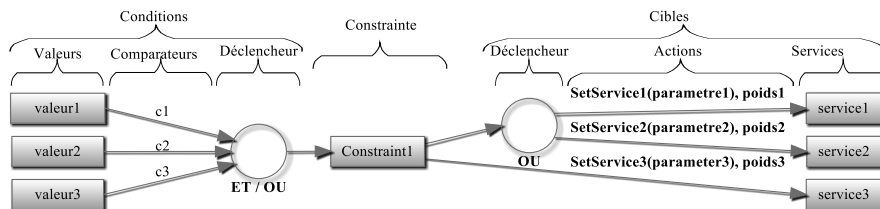


Figure 5. Contrainte explicite déclenchant plusieurs actions.

La figure 6a illustre un exemple d'une contrainte explicite déclenchant plusieurs actions. Dans cet exemple, si la puissance CPU disponible est inférieure à 50%, l'audio est joué et le contenu du texte est lu, alors que les vidéos sont à interdire. Comme vous pouvez le constater, pour une condition donnée, il est possible de déclencher plusieurs actions liées aux services de facettes différentes.

La figure 6b montre un exemple qui illustre la pondération sur des actions. Dans cet exemple, il permet d'exprimer le niveau de compréhension d'une langue en fonction du type de médias. Ici, John préfère le français à l'anglais pour la lecture de texte. Le paramétrage de ces poids n'est pas directement saisi par l'utilisateur. John peut spécifier à l'avance qu'il préfère des textes écrits en français qu'en anglais, et ces poids sont ensuite calculés automatiquement lors de la création du profil.

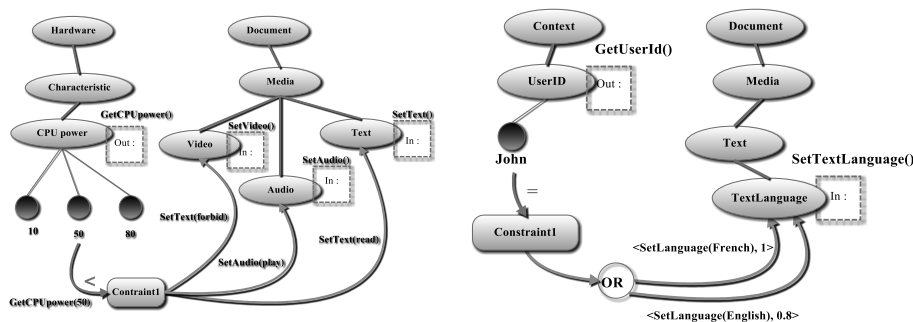


Figure 6a et 6b. D'autres exemples de contraintes explicites

Nous avons précisé les facettes et les contraintes. Maintenant, nous proposons de les intégrer dans un profil.

3.5 Profil

Un profil est un ensemble de facettes pouvant être enrichis par des contraintes.

Dans cet article, nous proposons l'utilisation de trois facettes: caractéristiques de la plateforme, information de contexte et structure de document. Comme nous l'avons montré précédemment, des contraintes de haut niveau peuvent être spécifiées. Par exemple, nous pouvons composer plusieurs conditions avec des opérateurs de

conjonction ou de disjonction et nous pouvons déclencher plusieurs actions avec des priorités. En outre, les paramètres des actions peuvent être des valeurs fixes ou peuvent se référer à des valeurs fournies par les services de profil.

Dans la section suivante, nous modélisons notre proposition intitulée SGP en UML et nous encodons un exemple de profil en RDF/XML. Ensuite, nous proposons des exemples de requêtes pouvant exploiter ce profil. Nous terminons par l'évaluation des performances de ces requêtes sur différentes plates-formes mobiles.

4 Profil générique sémantique

4.1 Modèle UML du Profil générique sémantique (SGP)

La figure 8 présente le méta-modèle de notre profil générique sémantique. Ce méta-modèle est suffisamment général pour convenir à différents processus d'adaptation.

Dans le méta-modèle, nous identifions deux catégories d'informations : les facettes, qui se composent d'une hiérarchie de services réutilisables et des contraintes qui représentent les associations entre les facettes et les services. Il est possible de spécifier de nouvelles facettes.

Ce modèle est générique car il permet d'exprimer les contraintes et les descriptions de services sans considérer l'application qui l'utilise. Ainsi, il n'est pas limité à des fonctions ou des applications spécifiques.

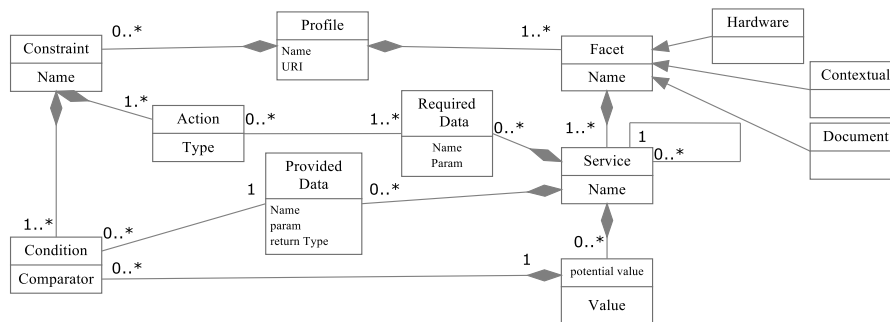


Figure 8. Le modèle du profil sémantique générique (SGP)

Dans leurs travaux, Strang et al. (Strang et al., 2004) analysent six approches de la modélisation de contexte et concluent que celle à base d'ontologies est la plus adaptée. Par conséquent, dans la partie suivante, nous proposons d'encoder nos profils avec un langage supportant les ontologies.

4.2 Profil SGP encodé en RDF/XML

RDF (Resource Description Framework) est un modèle standard pour l'échange de données sur le Web (Klyne et al., 2004). Plusieurs raisons nous incitent à utiliser ce formalisme : Tout d'abord, RDF nous permet d'effectuer des agrégations de

descriptions, ce qui peut être utile si plusieurs fichiers décrivent un profil. Deuxièmement, RDF peut manipuler des concepts sémantiques définis dans des ontologies, décrivant ainsi la sémantique du vocabulaire SGP. Par exemple, la sémantique nous permet d'affirmer que "Senseur" est équivalent à "Capteur". Ainsi, une requête sémantique sur «Capteur» s'appliquera également à "Senseur". Troisièmement, RDF ne nous force pas à exprimer des hiérarchies de données telles que défini dans d'autres langages (c'est une structure basée sur des graphes composés de triplets). Enfin, d'autres langages et propositions dont les objectifs sont de décrire les profils sont basés sur ce formalisme (CC/PP ou CSCP).

```
<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>
```

Figure 9. Extrait de l'exemple de profil en RDF/XML

Dans la figure 9, nous présentons un extrait d'un profil SGP encodée en RDF/XML qui est disponible sur SGPWG¹. Il est composé de la description du service (ScreenResolution). Ce service ScreenResolution est composé d'une fonction d'entrée (SetScreenResolution) et d'une fonction de sortie (GetScreenResolution) qui prend en paramètre une chaîne de caractères.

Dans la prochaine section, nous présentons brièvement quelques exemples de requêtes qui peuvent être utilisées pour récupérer certaines informations contenues dans notre structure de profil SGP.

4.3 Requêtes SPARQL

SPARQL est un langage de requêtes sur des descriptions RDF.

Sur le portail du SGPWG¹, nous présentons plusieurs exemples de requêtes SPARQL qui peuvent être spécifiées dans le but de récupérer des informations importantes contenues dans un profil SGP.

Notez que ces requêtes peuvent être utilisées par les processus d'adaptation afin d'en extraire des contraintes explicites.

¹ <http://www.sgpwg.eu>

```

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 .
}

```

Figure 10. Un exemple de requête SPARQL sur un profil SGP (R03).

D'autres requêtes peuvent être spécifiées : Une requête peut renvoyer une liste contenant tous les services du SGP (R02). Une requête peut retourner une liste contenant toutes les données de sortie des services du profil SGP (R03). Une requête peut retourner la liste des actions et des paramètres requis (R04). Par exemple : SetImageResolution (400x600), SetScreenLuminosityLevel (0,3). Une requête peut renvoyer la hiérarchie complète des services (R05).

Dans la section suivante, nous évaluons s'il est possible d'exécuter ces requêtes sur des terminaux mobiles.

4.4 Expérimentations

En utilisant la bibliothèque JENA³, nous avons testé les performances de plusieurs requêtes SPARQL sur des profils SGP encodés en RDF/XML. Nous avons effectué ces expériences sur deux configurations hétérogènes mobiles. La configuration 1 est un ordinateur portable fonctionnant sous Windows 7 (x64) avec 6 Go de RAM et un processeur quadruple cœur i7-2630QM (2 GHz). La configuration 2 est une tablette Samsung Galaxy Tab sous Android 3.2 avec 1 Go de RAM et un double cœur Tegra 2 (1 GHz).

Dans la figure 11a, nous comparons le temps d'exécution de requêtes sur les deux plateformes. Nous avons également réalisé des boucles de répétitions (10, 50, 100, 500, 1000 fois) en utilisant la requête R01, la requête R02 et une séquence de cinq requêtes différentes (R01, R02, R03, R04, R05). R01 est la requête SPARQL la plus simple : elle retourne tous les triplets (SELECT * WHERE { ?x ?y ?z .}). R02 à R05 étant la série de requêtes précédemment mentionnées.

Nous constatons que l'exécution des requêtes sur un profil SGP est en moyenne 14 fois plus rapide sur la première plate-forme (Windows) que sur la seconde (Android). Cependant, le temps d'exécution des requêtes sur Android reste en dessous de 8 ms contre 1,6 ms sous Windows. La différence de temps d'exécution entre deux requêtes sur un même SGP peut être du simple au double, nous devons donc définir une stratégie de requêtage pour optimiser les performances.

³ <http://incubator.apache.org/jena/>

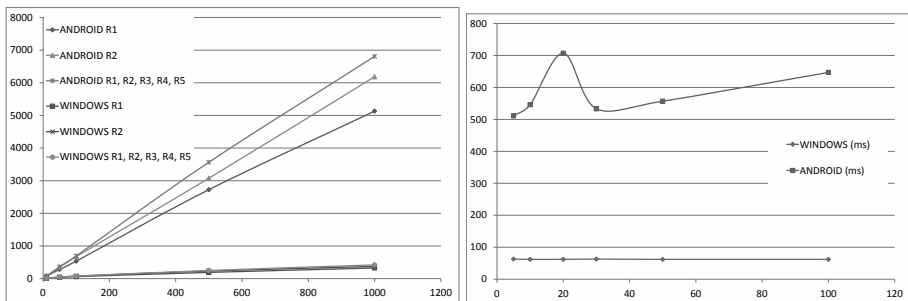


Figure 11a et 11b. Temps d'exécution des requêtes en fonction de la plateforme

Dans la figure 11b, on compare le temps d'exécution de 100 requêtes R1 sur des profils composés par 5, 10, 20, 30, 50 et 100 triplets. Notre objectif est d'identifier l'impact de la taille du profil sur le temps d'exécution des requêtes. Ainsi, nous notons que l'exécution de la requête est neuf fois plus rapide sous Windows que sous Android.

Ces expériences nous permettent de valider la possibilité d'exécuter des requêtes sur des configurations mobiles avec des performances suffisantes. Nous constatons que la taille des profils a un impact faible sur la performance d'exécution des requêtes. Nous souhaitons poursuivre l'évaluation de requêtes sur les profils, et plus particulièrement avec des requêtes sémantiques, et de définir une stratégie d'interrogation optimisée.

5 Futurs travaux

Nous continuons notre travail sur l'amélioration du modèle et nous souhaitons concevoir un Framework SGP. En effet, pour exploiter le SGP, il est nécessaire de disposer d'une couche logicielle capable de fournir des informations exploitables par les processus d'adaptation. La figure 12 présente le positionnement du SGP dans un environnement opérationnel.

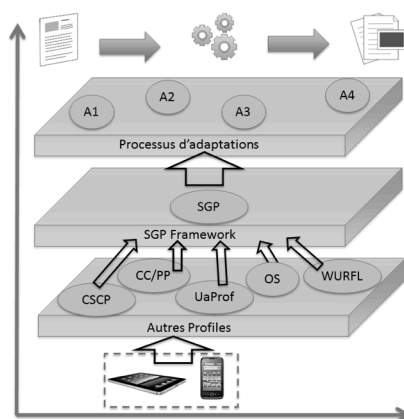


Figure 12. Framework SGP

Notre étude déterminera les besoins fonctionnels du Framework. Ainsi, il pourrait se composer de services assurant des fonctions telles que (1) la création et la mise à jour du SGP à partir de différentes sources (transformations de modèles, par exemple, intégrer des données de CC/PP), l'accès aux ressources du système, formulaire utilisateur, et analyse statistique du comportement de l'utilisateur. (2) L'analyse du contenu des documents pour générer des requêtes SPARQL nécessaires pour déterminer les contraintes à considérer. (3) Le traitement des contraintes afin de fournir des instructions aux processus d'adaptation. (4) D'autres pourraient gérer les interactions avec des services externes tels que les processus d'adaptation distants, l'accès à des capteurs externes (par exemple, GPS...).

6 Conclusion

Dans cet article, nous proposons un profil sémantique générique (SGP) qui permet d'exprimer des profils détaillés. Il se compose d'informations descriptives et d'expressions de contraintes explicites de haut niveau. Ce niveau de description permettra d'améliorer l'adaptation de documents multimédias en prenant en compte, entre autres, les préférences utilisateur dans des situations complexes. Notre modèle SGP considère plusieurs facettes et il est extensible. De plus, ce modèle est portable et générique. Nous proposons un encodage en RDF/XML qui permet d'utiliser toutes les technologies du Web sémantique (RDF Schema, ontologie, SPARQL...).

Nous sommes en cours d'expérimentation et de validation de notre proposition dans une application réelle (mise en œuvre sur des terminaux Windows/Android). Nous envisageons également d'étudier l'utilisation des ontologies pour proposer une architecture et des outils afin de générer semi-automatiquement des contraintes⁴.

7 Bibliographie

- Bolchini C., Curino C.A., Quintarelli E., Schreiber F. A. et Tanca L., "A Data-oriented Survey of Context Models," ACM SIGMOD Record, vol. 36, no.4, pp. 19-26, 2007.
- Buchholz S., Hamann T., Hübsch G., "Comprehensive Structured Context Profiles (CSCP): Design and Experiences," in Proc. of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, Orlando, Florida, 2004.
- Bulterman D., Jansen J., Cesar P., Mullender S., Hyché E., DeMeglio M., Quint J., Kawamura H., Weck D., Garcia Paneda X., Melendi D., Cruz-Lara S., Hanclick M., Zucker D., Michel. T. et W3C, "Synchronized Multimedia Integration Language (SMIL 3.0), W3C Recommendation," Décembre 2008. <http://www.w3.org/TR/SMIL/>.
- Chevalier M., Soulé-Dupuy C., Tchiéhom P., "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.

4. Ces travaux sont en rapport avec le projet ANR MOANO

- Ferscha A., Hechinger M., Riener A., Schmitzberger H., Franz M., Rocha M. et Zeidler A., "Context-Aware Profiles," *Autonomic and Autonomous Systems, ICAS '06*, 2006, USA.
- Forough S., et Reza J., "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.
- Indulska J., Robinson R., Rakotonirainy A., Henricksen K., "Experiences in Using CC/PP in Context-Aware Systems," *Mobile Data Management 4th International Conference (MDM '03)*, Melbourne, 2003.
- Jouanot F., Cullot N., and Kokou Y., "Context Comparison for Object Fusion," *Lecture Notes in Computer Science*, vol. 2681, p. 1031, 2003.
- Kagal L., Korolev V., Avancha S., Joshi A., Finin T., Yesha Y., "Centaurus: An Infrastructure for Service Management in Ubiquitous Computing Environments," *Wireless Networks*, vol. 1, 1995.
- Kagal L., Korolev V., Chen H., Anupam J., Finin T., "Centaurus : a framework for intelligent services in a mobile environment," *The 21st International conference on distributed computing system workshop*, Mesa, 2001.
- Kimiaei-Asadi M., Dufourd J.-C., "Context-aware Semantic Adaptation of Multimedia Presentations," *Multimedia and Expo, ICME 2005. IEEE International Conference* , Amsterdam, 2005.
- Klyne G., Carroll J., "Resource Description Framework (RDF): Concepts and Abstract Syntax," Février 2004. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.
- Klyne G., Reynolds F., Woodrow C., Hidetaka O., Hjelm J., Butler M. H., Tran L., "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0," 2004. <http://www.w3.org/Mobile/CCPP/>.
- Laborie S., Euzenat J., Layaïda N., "Semantic Adaptation of Multimedia Documents," *Multimedia Tools and Applications*, vol. 55, no. 3, pp. 379-398, Décembre 2011.
- Laplace S., Dalmau M., Roose P., "Kalinahia: Considering quality of service to design and execute distributed multimedia applications," *IEEE/IFIP Int. Conference on Network Management and Management Symposium, Salvador de Bahia, Brazil* , 2008.
- Lemlouma T., Layaïda N., "NAC: A Basic Core for the Adaptation and Negotiation of Multimedia Services," *Opera Project, INRIA*, September 2001.
- Lemlouma T., Layaïda N., "Content Adaptation and Generation Principles for Heterogeneous Clients," *W3C Workshop on Device Independent Authoring Techniques*, Germany, 2002.
- Passani L., "WURFL (Wireless Universal Resource File)," 2007. <http://wurfl.sourceforge.net>.
- Strang T., Linnhoff-Popien C., "A Context Modeling Survey," *Proceedings of the First International Workshop on Advanced, 6th Int. Conf. on UbiComp2004*, England, 2004.
- Veaceslav C., "Applying Next Generation Web Technologies in the Configuration of Customer Designed Products," *Royal Institute of Technology*, Stockholm, 2011.
- WAP Forum, User Agent Profile : "WAG UAProf, Wireless Application Protocol," 2001. <http://www.wapforum.org>.