

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Faculté de sciences exactes et sciences de nature et de la vie
Département : Sciences Exactes et Informatique



MEMOIRE DE MASTER

Domaine :MI
Filière :Informatique
Option :Système et multimédia

IDD:
Interface Dynamique Distribuée

Préparé Par : Abdesslam HATTAB
Karim AKKAL

Devant le jury :

Mr.Louardi BRADJI	MCB	Université de Tébessa	Président
Mr. Makhlouf DERDOUR	MCA	Université de Tébessa	Rapporteur
Mr. Issam BENDIB	MAA	Université de Tébessa	Examinateur

Date de soutenance :
Note : 15,50 Mention :...

Session 2015/2016

Université Larbi Tébessi -Tébessa

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Larbi Tébessi - Tébessa
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Déclaration sur l'honneur de non-plagiat

(à joindre obligatoirement au mémoire, remplie et signée)

Je soussigné(e),
Nom, Prénom : hattab Abdessalem et AKKAL Karim
Régulièrement inscrit(e) en **Master** au département : Sciences Exactes & Informatique
N° de carte d'étudiant : 2010/5030756/10 — 2010/40.6219.7/10
Année universitaire : 2015-2016
Domaine: Mathématique et Informatique
Filière: Informatique
Spécialité: SYM
Intitulé du mémoire : IDD : Interface Dynamique Distribuée

Atteste que mon mémoire est un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité. Je certifie également que je n'ai ni recopié ni utilisé des idées ou des formulations tirées d'un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales sont signalées entre guillemets.

Sanctions en cas de plagiat prouvé :

L'étudiant sera convoqué devant le conseil de discipline, les sanctions prévues selon la gravité du plagiat sont :

- L'annulation du mémoire avec possibilité de le refaire sur un sujet différent ;
- L'exclusion d'une année du master ;
- L'exclusion définitive.

Fait à Tébessa, le : 25-05-2016

Signature de l'étudiant(e) :

ملخص

مع التطور الهائل في المجال التكنلوجي خاصة بالنسبة لـ : تطور مصادر التفاعل (حواسيب شخصية, هواتف ذكية, لوحات رقمية ..), انتشار الشبكات اللاسلكية, واحتواء اغلب مصادر التفاعل على عدد مهم من المستشعرات....اصبح استخدام واجهات المستعمل الديناميكية أمرا ضروريا وحتميا .

عدة طرق تم اقتراحها في مجال تكييف واجهات المستخدم وانشاء تطبيقات ديناميكية ,حيث كان تصميم واجهات ديناميكية هو الحل ولكن اغلب هذه الاقتراحات واجتها جملة من النقائص فعلى سبيل المثال لا الحصر :اغلب الطرق لا تهتم بخصائص الجهاز ,الاتصال وتوزيع الخدمات , غياب الجمع بين تكييف واجهة المستخدم مع السياق و طريقة التفاعل , من جهة و الجمع بين تكـييف و تكـيف واجهة المستخدم من جهة اخرى ,و كذلك أغلب الطرق المقترحة لا تهتم بتكييف الواجهة بشكل ديناميكي ,

في عملنا هذا : قمنا باقتراح طريقة لتصميم واجهة مستخدم رسومية ديناميكية ,تحتوي على مجموعة من القطع (قائمة قابلة للتحريك, ازرار, حيز نص, حيز صورة, حيز فيديو ....) مع الخصائص المتعلقة بتكييف واجهات المستخدم في الوقت الحقيقي (إضافة, حذف, تحريك.....), مع محاولتنا استعمال مختلف طرق التفاعل (اللمس, الصوت...) , ولعل اهم شيء هو قدرة الطريقة التي قمنا باقتراحها على الجمع بين تكـــيف وتكييف الواجهة مع المحتوى وكذلك تكيف المحتوى مع السياق.

كلمات مفتاحية : واجهة ديناميكية ,التكيف ,السياق

abstract

With the great technologies like ; advances in interactive resources (PC, tablet, smartphone, etc.), wireless networks, and integration of different types of sensors in most of the interaction resources, etc ... the use of interfaces dynamic become an imperative necessity.
Several methods have been proposed for adapting interfaces and create dynamic applications ; the design of dynamic interfaces is a solution. However, most of the proposed work meets some limits such as : they does not take into account the characteristics of the device, connection and distribution services, the absence of the combination of the interface adaptation to the context and modality (touch, voice, etc ...) on at one hand and the combination of manual and automatic adaptation on the other. However, they neglect to adapt the interface dynamically.
In this work ; we will propose a method for dynamic interface design that contains widgets (drop-down lists, buttons, textview, imageview, VideoView ...) with properties related to adaptation in real time interfaces (adding, deleting, moving, ...) and we will use different modality (touch, voice, ect.). This method allows to adapt the content of the interfaces to the context and to adapt the interface to content using the manual and automatic adjustments.

Keywords : Dynamic interface, adaptation, context

Résumé

Avec les grandes technologies tells-que ; les progrès dans les ressources d'interaction (PC, Tablet, Smartphone, etc.), les réseaux sans fil, et l'intégration des diffèrents types des capteurs dans la plus part des ressources d'interaction, etc... l'utilisation des interfaces dynamiques deviennent une nécessité impérieuse.
Plusieurs méthodes ont été proposées pour l'adaptation des interfaces et la création des applications dynamiques dont la conception des interfaces dynamique est une solution. Toutefois, la plupart des travaux proposés répondent à quelques limites tels que : ne prend pas en compte les caractéristiques de l'appareil, la connexion et la distribution des services, l'absence de la combinaison entre l'adaptation des interfaces par rapport au contexte et la modalité (tactile, vocale, ect...) d'une part et la combinaison entre adaptation manuelle et automatique d'autre part. Cependant, ils négligent l'adaptation de l'interface dynamiquement.
Dans ce travail ; nous avons proposé une méthode de conception d'interface dynamique qui propose des widgets (listes déroulantes, boutons, textview, imageview, videoview...) avec des propriétés liées à l'adaptation des interfaces en temps réel (ajout, suppression, déplacement...), ainsi nous avons utilisé différent mode de modalité (tactile, vocale, ect.), cette méthode permet d'adapter le contenu des interfaces aux différents besoins des utilisateurs.

Mots clés : interface dynamique, adaptation, contexte ....

Remerciements

Tout d'abord, nous remercions DIEU de nous avoir donné la force et le courage qui nous a permis d'accomplir ce modeste travail.
Nous tenons à remercier notre encadreur Monsieur Derdour M., Docteur a l'université de Tébessa. pour leur suivi de ce travail,ses conseils précieux,ses juste critiques.
Nous tenons à remercier Monsieur le Docteur Bradji L d'avoir accepter d'honorer cette soutenance comme président de jury. et nous remercions Mensieur Bendib I qui a bien voulu examiner ce travail et d'etre membre de jury.
Nous tenons à remercier infiniment nos enseignants, exemple de sagesse et de patience, ainsi que pour leurs précieux conseils et leurs hautes qualités de respect pour leurs étudiants.
On n'adresse aussi nos remerciements à nos parents, et toutes les personnes qui ont participé de près ou de loin pour l'accomplissement de notre projet.

Table des matières

Introduction Générale **1**

1 Les Concepts de base **3**

1.1 Introduction 3

1.2 adaptation : 3

1.2.1 Classification de l'adaptation 4

1.3 les types d'adaptations 5

1.3.1 Adaptation des contenus 6

1.3.2 Adaptation des service 7

1.3.3 Adaptation des Interfaces 8

1.4 Contexte 8

1.4.1 Les systèmes sensibles au contexte 8

1.4.2 Les applications sensibles au contexte 9

1.5 Interface dynamique 10

1.6 Interface dynamique distribuée 11

1.7 Conclusion 11

2 Synthèse Des Traveaux Connexes **12**

2.1 Introduction 12

2.2 l'état de l'art : l'adaptation des interfaces 12

2.2.1 ArtStudio 12

2.2.2 TERESA 13

2.2.3 Adaptation des applications au contexte en utilisant les services WEB 14
2.2.4 EMMA 15
2.2.5 Approche de Hachani 15
2.2.6 Approche de Bouchelligua 16
2.2.7 CARACTH 17
2.2.8 La comparaison entre les travaux Synthétisés 18
2.3 Conclusion 19

3 MCIDD : Méthode De Conception Pour les Interfaces Dynamiques Distribuées 20
3.1 introduction 20
3.2 Contribution 21
3.2.1 Démarche 21
3.2.2 Processus Et Méthodes 22
3.3 Conclusion 31

4 Etude de Cas 32
4.1 introduction 32
4.2 Outils de développement 32
4.3 Scéario et implémentation 33
4.4 Mise en oeuvre : application de notre méthode sur le scénario 2 34
4.5 Conclusion 43

Bibliographie 45

Table des figures

3.1 la démarche de notre méthode. 22
3.2 diagramme de classe représenté la capture des informations du contexte et les besoin d'utilisateur 23
3.3 Extraction du contexte de l'usage device et serviceweb et la personnalisation avec le patron Observer 25
3.4 diagramme de classe illustre l'adaptation de l'interface 26
3.5 diagramme de class illustré l'adaptation du contenu. 27
3.6 diagramme de class présenté adaptation de media. 28
3.7 representation d'adaptation des services. 29
3.8 diagramme de class représenter Les trois premières étapes de la méthode. 30

4.1 Scénario 1 33
4.2 diagramme de class suivé durant l'utilisation de notre example . 35
4.3 interface avec la langue française. 38
4.4 interface avec la langue Anglaise. 38
4.5 interface d'inscription avec la langue française. 39
4.6 interface d'inscription avec la langue Anglaise. 39
4.7 interface adapter à écran de petite taille 40
4.8 interface adapter à écran de grande taille 40
4.9 adaptation de contenu au type de connexion 3G en écran petit taille. 41
4.10 adaptation de contenu au type de connexion 2G en écran petit taille. 41

4.11 adaptation de contenu au type de connexion 3G en écran de grande taille. 42
4.12 adaptation de contenu au type de connexion 2G en écran de grande taille. 42

Liste des tableaux

1.1 comparaison entre adaptation manuelle et automatique 11

2.1 synthèse de ArtStudio 13
2.2 synthèse de TERESA 14
2.3 synthèse de l'approche de (Tarak Chaari ,2007) 14
2.4 synthèse de EMMA (Vincent Ganneau1 et al ,2008) 15
2.5 synthèse de l'approche de Hachani 16
2.6 synthèse de l'approche de (Bouchelligua et al., 2010) 16
2.7 synthèse de CARACTH 17

Introduction Générale

De nos jours, on observe un développement considérable dans le domaine de la création d'applications mobiles, où un grand nombre d'applications ont été établis dans plusieurs domaines.
Malgré cette évolution et la grande quantité de ces applications sur les téléphones mobiles, télévision interactive, tablette ...etc, nous constatons que les utilisateurs de ces applications abandonnent leur utilisation. Ceci, et selon une étude réalisée par Grenier " Dans une période de 12 mois : Après un mois d'utilisation, seulement 38% des utilisateurs d'appareil lOS ou Android continueront d'utiliser l'application. Après 6 mois, seulement 14% l 'utiliseront encore et finalement 2% ".
Le problème d'abandon, est que les applications utilisées ne répondent pas aux besoins des utilisateurs, Pour résoudre ce problème nous vous proposons de créer des applications capables de s'adapter aux besoins des utilisateurs, les éléments qui peuvent mener à une adaptation consiste à prendre en considération de nouvelles données relatives à l'environnement d'utilisation et aux plates-formes d'interaction. Le triplet <utilisateur, plate-forme, environnement> constitue alors la cible de l'adaptation. On parle également de contexte d'utilisation.
Les interfaces dynamiques viennent pour répondre à l'une des pistes des applications sensibles aux contextes.
Plusieurs méthodes ont étés proposées dans la conception des interfaces dynamiques, mais la plupart des méthodes présentent quelques limites notamment : ne prend pas en compte les caractéristiques de l'appareil, la connexion et la distribution des services, l'absence de la combinaison entre l'adaptation des interfaces par rapport au contexte et la modalité d'une part (tactile, vocale, ect...) et la combinaison entre adaptation manuelle et automatique d'autre part.
Dans ce mémoire, nous proposons une méthode de conception d'interface graphique dynamique qui contient des widgets (listes déroulantes, boutons, textview, imageview, videoview...) avec des propriétés liées à l'adaptation en temps réel (ajoute, suppression, déplacement), ainsi nous utilisons différents

types de modalité (tactile, vocale, ect.), cette méthode permet d'adapter le contenu des interfaces au contexte et adapter l'interface au contenu.

Plans de travail :

- Dans le chapitre 1, nous passons en revue les concepts de base qui sont en relation avec le domaine des interfaces dynamiques distribuées tel que : l'adaptation, les types d'adaptations, classification des adaptations , interface dynamique , interface dynamique distribuée.
- Dans le chapitre 2, nous analysons quelques travaux en relation avec le domaine des interfaces dynamiques distribuées.
- Dans le chapitre 3, nous proposons une méthode de conception des interfaces dynamiques distribuées pour résoudre les problèmes trouvés après l'analyse des travaux connexes.
- le chapitre 4 est un chapitre d'étude de cas et implémentation, il permet de présenter quelques scénarios, ainsi vérifier la performance de la méthode proposée.

Chapitre 1

Les Concepts de base

1.1 Introduction

Ce chapitre définie les vocabulaires qui nous utiliserons dans la suite de cette étude, nous commençons par une définition concernant la notion d'adaptation avec ses déférents types (service, contenu, interface), et nous basons beaucoup plus sur la adaptation d'interface qui est le plus important dans notre sujet , ensuite la contexte et aussi les systèmes et les applications sensible au contexte et leur caractéristique, et finalement on définit l'interface dynamique et l'interface dynamique distribuée.

1.2 adaptation :

Selon Thierry.Vieville " Le dictionnaire le définit comme : " adapter " c'est un verbe qui signifie changer un "comportement ou une attitude " pour mieux correspondre aux "circonstances". 'Plus précisément il y a deux acteurs : (1) l'objet qui est adapté (c'est souvent un mécanisme, un instrument, un dispositif, un réflexe ou un comportement) et (2) le sujet : c'est le mécanisme qui adapte "[1].
L'adaptation est, selon le grand dictionnaire " l'opération qui consiste à apporter des modifications à un logiciel ou à un système informatique dans le but d'assurer ses fonctions et, si possible, d'améliorer ses performances, dans un environnement d'utilisation " [3].

Et selon Layaïda, la définition est : " Par adaptabilité, on entend les moyens automatiques ou semi-automatiques qui permettent aux contenus

[...] d'être utilisables sur des terminaux ayant des caractéristiques et des ressources très variées. "[LAY99]. Sur l'intérêt de l'adaptation, M. Satyanarayanan déclare que"L'adaptation est nécessaire quand il y a une disparité significative entre l'offre et la demande d'une ressource" [SAT01].

1.2.1 Classification de l'adaptation

Après la présentation des différentes raisons d'utilisation de l'adaptation dans une application, nous citons dans cette section les différentes classifications d'adaptation utilisées. en peut classé l'adaptation selon plusieurs critère parmi lequel on trouve deux qui sont : le premier qui est selon la méthode d'adaptation et la deuxième selon la façon d'adaptation.

Selon la méthode d'adaptation :

on distingue deux type principale dans cette catégorie :adaptation statique, adaptation dynamique.

- **Adaptation statique**
  L'adaptation statique vise la préparation de plusieurs versions d'une même ressource avant son exploitation. Cette ressource peut être une application, une image, une vidéo qu'un document. Ce dernier peut réutiliser dans un nouveau contexte après son adaptation manuellement. au début de l'apparition des terminaux mobiles, Cette technique a été adoptée pour plusieurs développements afin de pouvoir exploiter des ressources initialement prévues pour des PC standards. Avec une adaptation statique, le résultat est dédié au nouveau contexte d'utilisation de la ressource ce qui garantit la sûreté de son exploitation. Elle présente aussi une solution simple et efficace car chaque fournisseur s'occupe de l'adaptation de la ressource pour ses contextes. Cependant, elle présente l'inconvénient de la difficulté d'évolution pour la prise en charge de nouveaux contextes d'utilisation[CHA07].
- **Adaptation dynamique**
  On effectue l'adaptation dynamique a travers les transformations sur la ressource au cours de son utilisation. Il arrive que ce système d'adaptation dynamique après qu'il' intercepte des requêtes des utilisateurs, il exécute encore à la volée des transformations suivant les caractéristiques du contexte d'utilisation actuel. OPERA [LEM01] et MUSA [MEN02] sont des applications où les données sont adaptées dynamiquement aux caractéristiques du terminal utilisé. Elles présentent une

adaptation des pages WEB pour différents terminaux en agissant sur leur composition logique, spatiale, temporelle et hypertextuelle. Avec une adaptation dynamique, le temps de développement acquise est si important puisque la transformation des ressources est automatiquement à la demande de l'utilisateur. Cependant, des problèmes d'adaptation peuvent surgir dans un nouveau contexte d'utilisation inconnu, en fournissant un résultat d'adaptation qui parait inadéquat à ce contexte. En outre, le temps d'attente nécessaire à l'adaptation de la ressource peut gêner l'utilisateur. En effet, l'adaptation dynamique augmente automatiquement la latence de renvoi de l'information demandée par l'utilisateur[CHA07].

selon la façon d'adaptation :

On distingue deux type principale dans cette catégorie : adaptation automatique, adaptation manuelle.

- **Adaptation Automatique**
  Une application auto-adaptative est apte a modifier son comportement en répond à des changements dans son environnement d'opération. L'environnement d'opération fait référence peut être à des choses observées par le système logiciel, depuis les entrées d'un utilisateur jusqu'à l'information atteinte à partir de capteurs externes[ORE99]. Lorsque l'auto-adaptation est réalisée à travers la reconfiguration dynamique, il résulte dans la modification de l'architecture d'une application par l'application même [CER04].
- **Adaptation manuelle**
  Une application d'une adaptation manuelle est capable de modifier le comportement en réponse à des changements faits à partir d'un utilisateur manuellement.

1.3 les types d'adaptations

On remarque trois types d'adaptations

1. **l'adaptation de contenu (données) :** les données sont modifiées pour présenter à l'utilisateur qu'ils sont important à sa situation.
2. **l'adaptation de comportement(service) :**qui se traduit par l'adaptation des fonctionnalités ,est fournie par un composant ou un service

.

3. **l'adaptation d'interface (IHM) :** : l'interface de l'application présente les informations et propose des fonctionnalités qui sont pertinentes à la situation de l'utilisateur.

1.3.1 Adaptation des contenus

Selon LEMLOUMA.T " L'adaptation de contenu est définie généralement comme le processus qui transforme un contenu de son état initial vers un état final afin de satisfaire un ensemble de contraintes. Le contenu final peut provenir du même contenu source ou faire usage d'autres ressources de contenu. Satisfaire une contrainte donnée, revient à changer l'état du contenu d'origine afin que l'usage du contenu, dans son état final, ne pose pas de problème en la présence de la contrainte. Si un contenu donné ne peut pas satisfaire une contrainte de l'environnement, cela peut causer un problème d'utilisation du contenu dans cet environnement. "[LAM04].
L'adaptation de contenu se résume en trois opérations : le transcodage, le transmodage et la transformation. Les opérations d'adaptation du contenu sont aussi dépendantes de type (image, vidéo, audio ou texte) de contenu à adapter, et de l'évolution de consommation d'énergie dans le temps au niveau client.
plusieurs conversions sont possible, par exemple : une vidéo vers une suite d'image, une vidéo vers une bonde son, un texte vers une image etc. cette conversion s'appelle transcodage le transcodage est aussi nécessaire quand le client donne privilège à une modalité aux autres.

- **Principe de l'adaptation de contenu**
  Les services du modèle fonctionnel de l'application accordent un ensemble de données multimédia (images, vidéos, son, texte...). Ces données peuvent être inexploitables dans un contexte d'utilisation particulier. En citant l'exemple ou le terminal de l'utilisateur final de l'application n'accepte que des images de format PNG et que cet utilisateur invoque un service qui a son tour fournit des images sous un autre format, dans ce cas un transcodage est nécessaire sur les données. Le transcodage de format d'images ne considérait pas comme le seul opérateur d'adaptation requis. En effet, un nombre considérable de types d'adaptation peuvent être essentiels pour fournir des données exploitées d'une manière directe par le client.
  La réussite et l'efficacité de l'adaptation de contenu dépendent considérablement tantôt de la qualité et tantôt de la quantité de connais-

sances recueillies sur : le contexte d'utilisation de l'application, les outils d'adaptation de données disponibles et le contenu [CHA07].

Selon M.Derdour "nous parvenons de classer les traitements d'adaptation de médias en trois catégories :

- En premier lieu : Le Transmodage qui consiste à changer la modalité des médias. Par exemple, la transformation d'un texte en image si le terminal visé ne possède pas la police exigée.
- Ensuite, Le Transcodage consiste à modifier le format de codage des médias. Par exemple, la vidéo peut être transcodée du format MOV au format AVI.
- Enfin, La Transformation des médias qui rejettent le changement des modalités et des formats. Ce processus transforme le contenu, par exemple en réduisant la taille d'une image "[DER12].

1.3.2 Adaptation des service

Différentes approches manuelles et automatiques ont été proposées pour accomplir l'adaptation. Généralement, la tâche d'adaptation de service consiste à lancer un service ou un changement de sa forme si quelques événements se produisent dans l'environnement de l'équipement informatique. Les méthodes classiques d'adaptation de services emploient essentiellement les structures conditionnelles. Une condition est composée de plusieurs variables représentant les entrées de système intégrés avec les opérateurs logiques. Une fois qu'une condition devient vraie, le système devrait se comporter d'une façon bien définie [MIR09].

Un service ou une forme de service est fourni lorsqu' une configuration de contexte se produit. L'adaptation selon le contexte consiste à associer une configuration du contexte à un service ou à une forme d'un service d'une manière automatique [MIR09].

- **Configuration du service :**
  Représente un ensemble de paramètres prédéfinis fonctionnels ou non-fonctionnels à appliquer à un service afin qu'il délivre les fonctionnalités attendues.
- **Reconfiguration du service :**
  Reconfigurer un service permet de l'adapter c'est-à-dire de modifier sa structure et/ou son comportement fonctionnel pour répondre à la variation de son contexte opérationnel.

1.3.3 Adaptation des Interfaces

L'adaptation des interfaces homme-machine vise une production des systèmes interactifs tenant compte à la forme de menu convenable à la situation dans le but que l'utilisateur atteigne ses objectifs.

- **Principe de l'adaptation des interfaces**
  Les interfaces d'interaction assurent les échanges de données entre l'utilisateur final de l'application et les différents services de l'application. Ces interfaces doivent fonctionner correctement dans leur contexte d'utilisation. La bonne exécution de ces interfaces est conditionnée par leur capacité à s'adapter au terminal utilisé et aux préférences de l'utilisateur.
  L'adaptation d'interface dépend principalement des caractéristiques du terminal utilisé (facette terminal du contexte) et des préférences de l'utilisateur (facette utilisateur du contexte). Elle dépend aussi de la description des données adaptées de l'application. [CHA07].

1.4 Contexte

La notion de" contexte" a occupeune place se primordial dans le domaine de la littérature. Elle a attiré l'attention d'un grand nombre de chercheurs dans plusieurs domaines tels que la philosophie, la psychologie, l'informatique, etc.
Parmi les définitions dans le domaine d'informatique :
Selon DEY.A" le contexte couvre toutes les informations pouvant être utilisées pour caractériser la situation d'une entité. Une entité est une personne, un lieu, ou un objet qui peut être pertinent pour l 'interaction entre l'utilisateur et l 'application, y compris l 'utilisateur et l 'application eux-mêmes " [DEY01].
Et selon CHE .G "Le contexte est un ensemble d'état et de paramètres qui soit détermine le comportement d'une application ou bien dans lequel un évènement d 'application se produit et est intéressant pour l 'utilisateur "[CHE00].

1.4.1 Les systèmes sensibles au contexte

Un système est dit sensible au contexte s'il est capable de s'adapter aux changements contextuels en temps réel. D'où l'apparition de la définition la

plus utilisée par la plupart des chercheurs dans leurs contributions, celle de Dey" Un système est sensible au contexte s'il utilise le contexte pour fournir des informations et des services pertinents pour l 'utilisateur, ou la pertinence dépend de la tâche demandée par l 'utilisateur"[DEY00].

Cette définition a mis en évidence trois étapes indispensables pour qu'un système soit qualifié de "sensible au contexte " :

- Capture et stockage du contexte d'une façon transparente pour l'usager.
- Interprétation du contexte capté afin de changer l'état de représentation des services. Ces derniers seront capables d'être exploités par les utilisateurs finaux de l'application à travers des actions à exécuter. Ces tâches forment implicitement les préférences des usagers.
- Passage des informations interprétées à l'application afin de garantir l'exécution des services selon le changement contextuel sans l' intervention explicite des utilisateurs [CHE 14].

1.4.2 Les applications sensibles au contexte

Les applications sensibles au contexte sont des applications dont le comportement peut varier en fonction du contexte[DEY01]. Une application sensible au contexte qu'elle doit automatiquement extraire de l'information ou effectuer des actions en fonctions du contexte utilisateur détecté par les capteurs [BRO 98].
Une application est dit sensible au contexte s'elle est capable de s'adapter aux changements contextuels en temps réel.
dans le domaine de l'adaptation des systèmes interactifs identifie le contexte par le triplet " Utilisateur, Plate-forme, Environnement " [CAL03] où :

- **l'utilisateur :** est typiquement décrit par ses capacités (perceptuelles, motrices et cognitives), mais aussi par ses préférences et ses caractéristiques culturelles (e.g. langue naturelle).
- **la plate-forme (user device) :** qui se décline en deux sous catégories :
  - la plate-forme physique (matérielle).
  - la plate-forme logique (logicielle).
- **l'environnement :** se réfère à l'environnement physique accueillant l'interaction.
- **Caractéristiques des applications sensibles aux contextes :**
  - Les informations et les services peuvent être présentés à l'utilisateur dans le contexte courant. Ceci inclut la sélection d' informations de proximités ("où est la banque la plus proche ?") et de services ainsi que les commandes contextuelles (interface modifiable).

- ? Exécution automatique d'un service dans un certain contexte. Ceci inclut les actions déclenchées par des capteurs de contexte ("context-triggered actions"), si la lampe s'allume quand une personne rentre dans une pièce et l'adaptation automatique au contexte, avec par exemple un téléphone qui change son volume d'écoute en fonction du bruit ambiant.
- étiquetages du contexte à l'information pour une restitution ultérieure.[DEY00]

1.5 Interface dynamique

Nous allons commencer par la définition des interfaces statiques, a partir de cette dernière on arrive a déduire une définition de l'interface dynamique. Donc l'interface statique est une interface dans laquelle la modification par l'utilisateur ou par l'application elle-même est impossible, autrement dit il est inchangeable, il garde le même style au temps d'exécution, car il n'existe pas une relation entre ce type d'interfaces et l'adaptation des interfaces.
Par contre on trouve que l'interface dynamique donne la possibilité aux utilisateurs ou bien aux applications qui gèrent l'interface de modifier le contenu de l'interface pour répondre aux changements des besoins de l'utilisateur, généralement ce type d'interface est utilisé dans les systèmes ou bien les applications sensibles au contexte.

Il existe deux types d'interfaces dynamiques :

1. **Interfaces adaptables : adaptation manuelle :** Un système est dit adaptable, s'il fournit à l'utilisateur les outils qui lui permettent de modifier les caractéristiques du système. L'interface peut être ajustée à l'initiative de l'utilisateur. Il peut alors décider d'adapter l'interface selon ses besoins afin de pouvoir réaliser l'action désirée plus facilement. Le système n'est basé sur aucun algorithme intelligent, il est passif au sens où c'est l'utilisateur qui prend les décisions. Ce type d'interface est également désigné par le terme d'interface paramétrable ou configurable, puisque l'utilisateur peut ajuster à n'importe quel instant des paramètres de l'interaction[SIM07].
2. **Une interface adaptative : adaptation automatique :** Une interface peut être non seulement adaptable mais également "adaptative" .Il s'agit d'interfaces qui s'adaptent d'elles-mêmes en fonction des évolutions du contexte sans que celui-ci n'ait à intervenir pour les paramétrer. pour qui un système est dit adaptatif, s'il est capable de rectifier ses caractéristiques personnel au détriment des besoins de l'utilisateur

un système adaptatif est un système interactif qui adapte son comportement à chaque individu sur la base d'inférences non triviales, élaborées à partir des informations venant de l'utilisateur, Elles correspondent aux interfaces "intelligentes" ou "virtuelles", et aux interfaces de demain ...[SIM07]

Adaptation manuelle	Adaptation automatique
- L'utilisateur choisit les propriétés - facile à développer. - Le problème c'est que l'utilisateur ne sait pas toujours quelle est la meilleur propriété qui lui convient.	- Les propriétés choisissent automatiquement - Difficile à développer

TABLE 1.1 – comparaison entre adaptation manuelle et automatique

1.6 Interface dynamique distribuée

C'est une interface qui contient des composants modifiable ou on peut ajouter ou supprimer des composants, l'exécution des composants de l'interface est assurée complètement ou partiellement par l'appareil ou par d'autres plateforme (Cloud, appareils voisin, ordinateur,...).

1.7 Conclusion

Dans ce chapitre nous avons identifié les concepts relatifs à notre thème, et nous présentons la notion d'adaptation avec ses types(Contenu, Service, et Interface). Nous évoquerons aussi l'utilité d'adaptation d'interface qui m'intéressons, dans le chapitre suivant nous synthétisons les travaux d'adaptation d'interface existants et nous présentons les objectifs de ce travail de thèse.

Chapitre 2

Synthèse Des Traveaux Connexes

2.1 Introduction

Dans ce chapitre, on met l'accent sur les travaux existants des interfaces graphiques dynamiques. Nous analysons quelques méthodes de conception qui aident au développement de ce genre d'interface. Enfin, nous nous présentons les défis existes dans les travaux analysé, et nous nous essayerons de proposer des solutions.

2.2 l'état de l'art : l'adaptation des interfaces

Dans cette section, nous allons étudier quelques travaux ayant une relation avec l'adaptation des interfaces. Chaque travail sera étudié et suivie par un tableau qui représente le travail étudié brevement ,et la suite des ces travaux en conclut par un tableau qui résume les méthodes proposées et indique la différence entre elles.

2.2.1 ArtStudio

ArtStudio (Adaptation par Réification et Traduction Studio) est un environnement logiciel pour le développement des interfaces plastiques multi-cible [THE01].
ArtStudio fait partie des premiers environnements respectant le cadre de référence CAMELEON.
Le point fort d'ArtStudio réside dans la génération d'IHM multi-cibles étant

donné que cette approche embarque un ensemble de règles pour choisir l'interface la plus appropriée à la plate-forme cible [SOT08].
Cette approche ne s'intéresse qu'à l'adaptation de l'interface graphique par rapport à la plateforme. Cependant, l'approche ArtStudio est limitée à des plates-formes graphiques Java et des pages Web.
ArtStudio ne fait qu'une génération d'interface graphique qui supporte les dispositifs d'interaction usuels (clavier, écran, souris ou eq) Cette adaptation peut être réalisée pendant la conception de l'IHM (Design time).

Contexte d'usage	Plateforme -software-
Moment d'adaptation	Design time
Type d'adaptation	Adaptation manuelle
Multimodale	non
Support Adaptation de modalité	-
Type d'adaptation	-
Moment d'adaptation	-

TABLE 2.1 – synthèse de ArtStudio

2.2.2 TERESA

TERESA(Transformation Environnent for inteRactivE Systems representAtions)

Est un environnement qui a pour but la génération semi-automatique d'interfaces multiplateforme et multimodale (graphique et vocale).
Le principe suivi dans cette approche était ? One Model, Many Interfaces ?,les auteurs ont adopté le cadre de référence CAMELEON basé sur l'utilisation d'un seul modèle de départ pour en dériver plusieurs interfaces cibles.
Pendant toutes les phases de conception, TERESA s'appuie sur la notation XML
TERESA régit une génération d'interface graphique ou bien vocal , mais ne supporte pas l'adaptation de modalité , uniquement l'adaptation de l'interface par rapport à la plateforme, cette adaptation réalisée pendant la conception de l'IHM (Design time)[BER04].

Contexte d'usage	Plateforme -software-
Moment d'adaptation	Design time
Type d'adaptation	Adaptation manuelle
Multimodale	Oui (graphique et vocale)
Support Adaptation de modalité	non
Type d'adaptation	-
Moment d'adaptation	-

TABLE 2.2 – synthèse de TERESA

2.2.3 Adaptation des applications au contexte en utilisant les services WEB

Ils proposent une approche basée sur les composants permettant d'adapter les systèmes interactifs par transformation dynamique d'assemblages de composants, qui sont prédéveloppés pour des besoins spécifiques. L'ajout ou le retrait de composants, ainsi que la création ou la suppression de liaisons entre composants, permettent alors de construire divers systèmes interactifs. Cette approche résout le problème d'interface graphique dynamique (soit adaptive ou adaptable) , mais elle présente l'inconvénient d'utiliser un ensemble de variantes fixes et prédéfinies pour diverses utilisations . Par exemple dans le projet SICOM présenté dans l'article, la personnalisation des interfaces on a seulement trois chois pour personnaliser interface et pour l'adaptation dynamique on a seulement adaptation par rapport à la taille de écran[aa].

Contexte d'usage	Plateforme -hardware- taille de écran utilisateur
Moment d'adaptation	Real time
Type d'adaptation	Adaptation automatique pour -Plateforme- Adaptation manuelle pour-utilisateur-
Multimodale	Oui
Support Adaptation de modalité	non
Type d'adaptation	-
Moment d'adaptation	-

TABLE 2.3 – synthèse de l'approche de (Tarak Chaari ,2007)

2.2.4 EMMA

EMMA (Embedded Manager for Mobile Adaptation) propose un système embarqué capitalisant les actions de l'utilisateur en mobilité pour identifier ses contextes d'usage clés et mettre en oeuvre les adaptations appropriées à chacun d'eux. EMMA collecte les données d'usage (application utiliser, personnalisation effectué) situées dans le temps (jour de la semaine, moment de la journée), localisation géographique. Par exemple :
Chaque dimanche à 9 h l'utilisateur en réunion dans son bureau (localisation géographique). L'application propose l'adaptation en mode réunion, ce mode extrait par rapport aux personnalisations effectuées chaque semaine à 9 h (par exemple basculer le téléphone en mode vibreur ...etc.) .
Mais l'adaptation en mode réunion par exemple ne se fait pas automatiquement, il sera proposée à l'utilisateur qui décide d'y basculer. Le problème de cette conception basée sur l'utilisateur c'est qu'elle ne prend pas en considération l'existence des personnes qui n'utilisent pas un emploi de temps hebdomadaire mais annuel ou mensuel. Autre problème c'est le dérangement de l'application quand chaque fois demande l'autorisation de basculer. Pire, le contexte d'adaptativité identifié dans la conception (travail , maison) n'est jamais modifié pour l'ajout ou la suppression d'un contexte [SOT08]. En résumé on dit qu'EMMA est un projet réalisé par une méthode de conception qui n'utilise que le contexte utilisateur et ce contexte est adapté seulement en deux mode (travail ,maison).

Contexte d'usage	utilisateur environnement : (localisation géographique)
Moment d'adaptation	Real time
Type d'adaptation	hybride-adaptation
Multimodale	non
Support Adaptation de modalité	non
Type d'adaptation	-
Moment d'adaptation	-

TABLE 2.4 – synthèse de EMMA (Vincent Ganneau1 et al ,2008)

2.2.5 Approche de Hachani

Les auteurs de cette approche (Hachani et al. 2009) proposent une méthode générique pour l'adaptation des IHM sensibles au contexte. Ils déve-

loppent des règles de transformation génériques pour l'adaptation du modèle des tâches au contexte. les auteurs effectuent une adaptation de l'interface à la langue de l'utilisateur et à la taille de l'écran[HAC09].

Contexte d'usage	Plateforme -hardware- taille de écran utilisateur : langue
Moment d'adaptation	Real time
Type d'adaptation	auto-adaptation
Multimodale	oui
Support Adaptation de modalité	non
Type d'adaptation	-
Moment d'adaptation	-

TABLE 2.5 – synthèse de l'approche de Hachani

2.2.6 Approche de Bouchelligua

Les auteurs publient un article (An MDE Approche for User Interface Adaptation to the Context of Use) dans le journal springer, cet article propose une approche qui permet la génération des IHM plastiques. L'adaptation de l'interface générée se fait en respectant les différents éléments du contexte " Utilisateur - Plateforme - Environnement "[BOU10]. Heureusement on trouve une approche qui utilise les trois éléments du contexte " Utilisateur - Plateforme - Environnement " pour faire l'adaptation d'interface. Mais malheureusement L'adaptation se fait pendant le design time

Contexte d'usage	Utilisateur-Plateforme-Environnement
Moment d'adaptation	Design time
Type d'adaptation	auto-adaptation
Multimodale	Oui (tactile ,vocal)
Support Adaptation de modalité	-
Type d'adaptation	-
Moment d'adaptation	-

TABLE 2.6 – synthèse de l'approche de (Bouchelligua et al., 2010)

2.2.7 CARACTH

Les développeurs disent que plusieurs utilisateurs cessent d'utiliser un produit ou bien une application, le facteur principal est l'ignorance de ces utilisateurs et leur facultés physique dans la processus de conception de cette application et précisément de l'interface graphique de cette application . Vestizou , thomann et françois ont proposé une méthodologie de conception permettant l'enchaînement de la caractérisation des spécifique motrice des usages vers la conception d'interface personnalisée .
Les auteurs proposent un processus de conception contient quatre étapes :

1. Analyser les spécificités physiques : L'analyse permet une collection des performances motrices de l'utilisateur par le concepteur.
2. Créer un modèle utilisateur : Le modèle utilisateur représente les facultés physiques sous forme de données statistiques .
3. Proposer une solution de conception Concevoir des interfaces pour satisfaire aux exigences de l'utilisateur (interface adaptable a l'utilisateur).
4. Evaluer la solution de conception : Cette étape permet aux concepteurs de vérifier si l'interface adapter répond aux exigences de l'utilisateur ,si l'interface ne répond pas aux exigences utilisateur en retour à l'étape 3[VEY14].

Malheureusement la méthodologie de conception proposée (CARACTH) adapte l'interface à l'utilisateurs par rapport à son contexte et seulement sur les spécificités motrices , et cette adaptation se déroule au moment de la conception de l'interface. En résumé on dit que CARACTH est une méthodologie de conception adapte seulement la modalité d'interface (tactile ,vocal ...).

Contexte d'usage	utilisateur :spécifité motrice
Moment d'adaptation	Design time
Type d'adaptation	auto-adaptation
Multimodale	Oui (tactile ,vocal)
Support Adaptation de modalité	oui
Type d'adaptation	Design time
Moment d'adaptation	auto-adaptation

TABLE 2.7 – synthèse de CARACTH

2.2.8 La comparaison entre les travaux Synthétisés

le tableau suivant présente une comparaison entre les différents traveaux syntétisés ,pour connaitre les avanages et les limites de chaque travail

	Contexte d'usage			Adaptation de la modalité	Moment d'adaptation		Type d'adaptation	
	Plateforme	Utilisateur	Environnement		real time	Design time	manuel	Automa -tique
ArtStudio	✓	X	X	X		✓		✓
TERESA	✓	X	X	X		✓		✓
Tarak Chaari et al	✓	✓	X	X	✓		✓	✓
EMMA	X	✓	✓	X	✓		✓	
Hachani et al	✓	✓	X	X	✓			✓
Bouchelligua et al.	✓	✓	✓	X		✓		✓
CARACTH	X	✓	X	✓		✓		✓

Les méthodes proposées dans la conception des interfaces graphique présentent quelques limites notamment :

- Complexe à implémenter dans des cas ou propose une conception très chère à implémenter dans d'autres cas ; par exemple dans le cas de CARATCH il faut utiliser des capteurs et des logiciel de traitement payant.
- EMMA propose le mélange de l'adaptation entre l'adaptation automatique et l'adaptation manuelle mais la solution proposer dérange l'utilisateur chaque fois l'adaptation manuelle demande l'autorisation de bascule a un autre mode.
- Ne prendre pas en compte l'adaptation de modalité par exemple dans ARtStudio on a seulement un modalité (graphique), et dans TERESA on a deux modalités (graphique , vocal) et en choisi la modalité dans l'étape de la conception et n'existe pas un adaptation entre les modalités .
- CARACTCH propose l'adaptation de modalité mais malheureusement cette adaptabilité fonctionne dans la période de la conception seulement.
- Les méthodes analysées supportent l'adaptation de la présentation au contenu, et ne supportent pas l'adaptation du contenu.

après la synthèse des travaux nous n'avons pas trouvé une méthode de conception combinée entre l'adaptation des interfaces par rapport le contexte et la modalité et elle est intéresse avec l'adaptation du contenu, qui permet de laisser le choix d'utilisateur soit ; l'utilisation de l'interface proposée par l'adaptation automatique ou bien faire une personnalisation a l'interface proposée (adaptation manuelle).

Nous allons essayé de proposer une méthode de conception d'interface graphique dynamique qui contient des widgets (listes dérolantes, boutons, textview, imageview, videoview ...) avec des propriétés liées à l'adaptation en temps réel (ajoute, suppression, déplacement), cette méthode permet d'adapter le contenu des interfaces au contexte.

Cette méthode propose aussi une conception très simple et facile à implémenter.

Après la proposition de la méthode nous allons essayer d'implémenter cette dernière dans l'étude de cas de l'interface dynamique est en distribuant cette dernière en multi-users et multi-devices .

2.3 Conclusion

Après avoir synthétisé les travaux de recherche faites qu'ont en relation avec l'adaptation des interfces , nous n'avons pas trouvé une méthode de conception pour l'adaptation des interfaces automatiquement et manuellement au contexte, modalité, et contenu et elle s'intéresse à l'adaptation de ce dernier. Dans ce cadre, nous allons essayer, dans le chapitre suivant, de proposer une méthode de conception d'interface graphique dynamique, cette méthode permet d'adapter les interfaces au contexte, modalité et au contenu et adapter ce dernier au contexte.

Chapitre 3

MCIDD : Méthode De Conception Pour les Interfaces Dynamiques Distribuées

3.1 introduction

Introduction : Une interface dynamique est utilisée généralement dans les applications sensibles au contexte, l'interface est réalisée par exécution des tâches suivantes :

- capturer les différents changements au niveau du contexte des besoins d'utilisateur.
- établir la meilleure interprétation de ces informations et les présenter sur une interface dynamique.

Mais comme nous avons vu dans le chapitre précédant, les conceptions d'interface dynamique proposées dans les travaux déjà faits contiennent plusieurs problèmes par exemple :
La Complexité d'implémenter les méthodes de conceptions proposées, la majorité des méthodes de conception proposées n'arrivent pas à combiner entre l'adaptation automatique et l'adaptation manuelle... Afin de répondre à cette problématique nous proposons une méthode de conception d'interface adaptable au changement très simple et très facile à implémenter...

3.2 Contribution

Pour réaliser notre méthode, nous allons utiliser des capteurs permettant de récupérer les informations du contexte et les besoins des utilisateurs, et on utilise le patron de conception " observateur " pour détecter les changements dans le contexte et les besoin d'utilisateur, et finalement on propose une interface dynamique adaptable à l'utilisateur, ce dernier peut changer l'interface pour répondre à ses besoins.

3.2.1 Démarche

Notre méthode de conception illustrée par une démarche que l'on peut résume en ceci :

1. recuperation des informations du contexte et besoin de l'utilisateur : Capter les informations du contexte et les nouveaux besoins des utilisateurs.
2. analyse des changement aux besoins et au contexte : détecter les changements aux besoins et aux contextes par le composant observateur.
3. Conception et proposition de l'interface adaptable aux changements.
4. évaluation de l'adaptation de l'interface :Vérifier l'utilité et l'utilisabilité de l'interface .
   selon la satisfaction de l'utilisateur ,nous deciderons de terminer le processus ou retourner vers l'étape 1
   Si l'utilisateur est satisfait :aller à étape 5
   si non le système va recomancer le processus d'adaptation
5. deploiement de l'interface adaptée

Et la figure 3.1 suivante résume les étapes de notre démarche

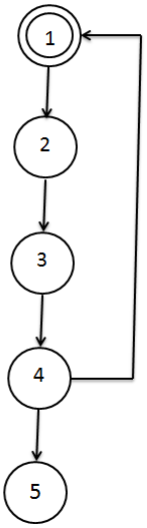


FIGURE 3.1 – la démarche de notre méthode.

3.2.2 Processus Et Méthodes

Etape 1 : recupération des informations du contexte et besoin de l'utilisateur : Capter les informations du contexte et les nouveaux besoins des utilisateurs

1. **Capter les informations du contexte** Nous avons identifié deux sources de contexte :
   - Device
   - service
   - **Capter du contexte du Device** En utilise les capteurs pour extraction des contextes liés à l'environnement de l'utilisateur. Les appareils mobile contient des capteurs, en utilise les capteurs pour mesure et détecter les variations du contexte de l'utilisateur telles que la position géographique, la luminosité, la température, le niveau de bruit, l'humidité, etc.
     Ils détectent également les variations liés à la plateforme telle que la version du système d' exploitation, la taille de l'écran, la langue.
   - **Capture du contexte du service** En utilise service web pour stocker les informations du contexte utilisateur fixé. Par exemple : la température préféré. Le niveau de bruit maximal qui peut supporter. La langue préférée...etc.

2. **Capture des besoins d'utilisateur** Les options qui peuvent ajouter ou bien supprimer par les déférents utilisateurs pour la personnalisation de l'interface. L'utilisateur personnaliser l'interface manuellement a votre besoin. Par exemple : ajouter des buttons. Supprimer des buttons. Changer la modalité . Changer les couleurs de l'interface...ect.

Et la figure 3.2 ci dessus représente la capture des informations du contexte et les besoin d'utilisateur.

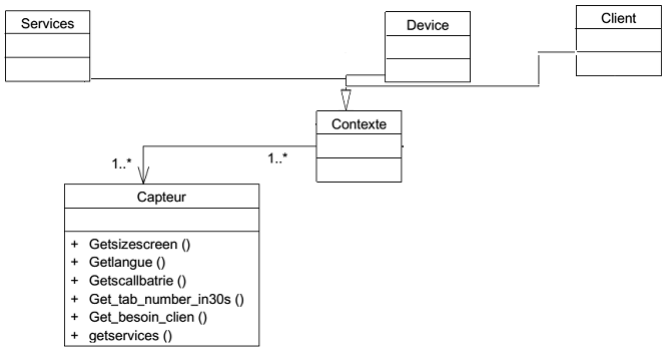


FIGURE 3.2 – diagramme de classe représenté la capture des informations du contexte et les besoin d'utilisateur .

Etape 2 : analyse des changement aux besoins et au contexte : détecter les changements aux besoins et aux contextes par le composant observateur.

Le patron de conception (OBSERVER) offre une meilleure solution pour la détection des changements en temps real.

- **Les patrons de conception** Un patron de conception (Design Pattern) est une solution à un problème récurrent dans un contexte donné, déjà abordé lors de la conception d'autres systèmes tels que, la détection d'un changement sur un objet ...Etc. " Un patron décrit un problème à résoudre, une solution, et le contexte dans lequel cette solution se situe"[JOH97]. On peut considérer un patron de conception comme formalismes orientés objet, ce qui signifie qu'on privilégie les solutions éprouvées. On a 3 catégories de patrons :
  - Création : concerne le processus de création des objets
  - Structure : lié à la composition des classes ou objets
  - Comportement : caractérise les façons dont les classes ou les objets interagissent ou distribuent les responsabilités.
- **Le patron de conception OBSERVER** Est un patron de type Comportement, Comme son nom et sa structure, il offre la possibilité de l'observation en étant muni d'un processus de notification . Par conséquent, le modèle Observateur/Observable est utilisé en programmation pour envoyer un signal par un sujet (observable) à l'observateur si on a un changement dans le sujet (observable). [CHE14]
  Observateur, étant à l'écoute permanente des événements et du changement d'état, il enregistre les observations et effectue la tâche adéquate en tenant compte des informations enregistrées.
  L'observateur agit donc d'une manière transparente selon l'état du sujet.
- **L'utilisation du patron de conception Observer dans notre cas**
  - **Détection du changement dans le composant Device** Donc, pour obtenir les informations reliés aux contextes de l'utilisateur, nous allons utiliser le DEVIC1E (Smartphone utilisateur) comme un observable. Les différentes variations du contexte de l'utilisateur sont détectées par des capteurs intégrés dans le DEVICE, dans chaque détection de changement, une notification sera envoyée à L'observateur.
  - **Détection du changement dans le composant Client** Personnalisation de l'interfaces manuellement par l'utilisateur représente le deuxième sujet d'observation, si l'utilisateur changer les options de l'interface la classe Adapter changer, l'observateur détecter change-

ment puisque la classe Adapter est un observable est comme nous dit observable envoyer un signal a l'observateur s'il existe un changement.

– **Détection du changement dans le composant Service** Le service représente le troisième sujet d'observation alors que l'observateur sera le même déjà appliqué au DEVICE.
La figure 3.3 suivante représente l'utilisation du patron Observateur pour la détection des changements du contexte d'usage (Device, services) et les personnalisations qui fait par l'utilisateur.
Comme nous avons dit précédemment observateur(Observation) reste en écoute pour recevoir le signal envoyer par le observable (Capteur) s'il existe un changement au niveau des donneés capter.

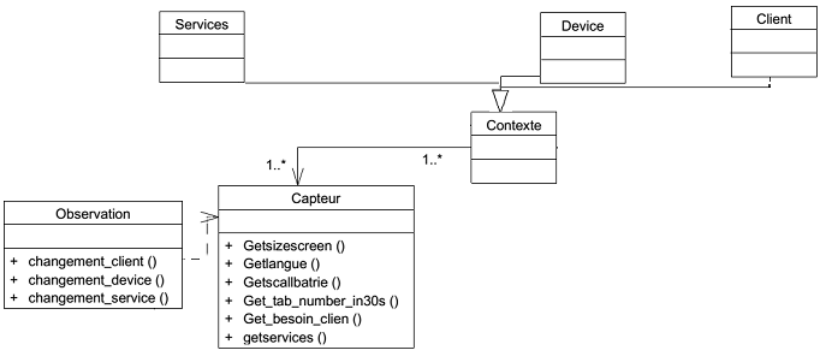


FIGURE 3.3 – Extraction du contexte de l'usage device et serviceweb et la personnalisation avec le patron Observer .

Etape3 :Conception et proposition de l'interface adaptable aux changements

Apres la capture de changement sur le contexte nous proposons une adaptation d'interface qui répond à ce dernier selon les trois types d'adaptation suivant :

1. **Adaptation de l'interface** Concernant les l'adaptation des interface on a deux types, l'un qui est défini l'adaptation statique ou le concepteur prépare plusieurs versions des interfaces et l'utilisateur utilise une de ses interface qui est proposée par l'application, par contre le deuxième type qui se base sur la notion de dynamité ou nous avons utilisé les widgets (liste déroulante, bouton, textview) et en donnent

a l'utilisateur la possibilité d'ajouter, supprimer, déplacer ou bien dupliquer les widgets d'une manière convenable à ce qu'il veut changer sans oublier la possibilité de changer la modalité concernant le tactile, graphique ou bien vocale selon le besoin, et la structure l'adaptaion d'interface est représenté par diagramme de classe de la figure3.4 ci-dessus

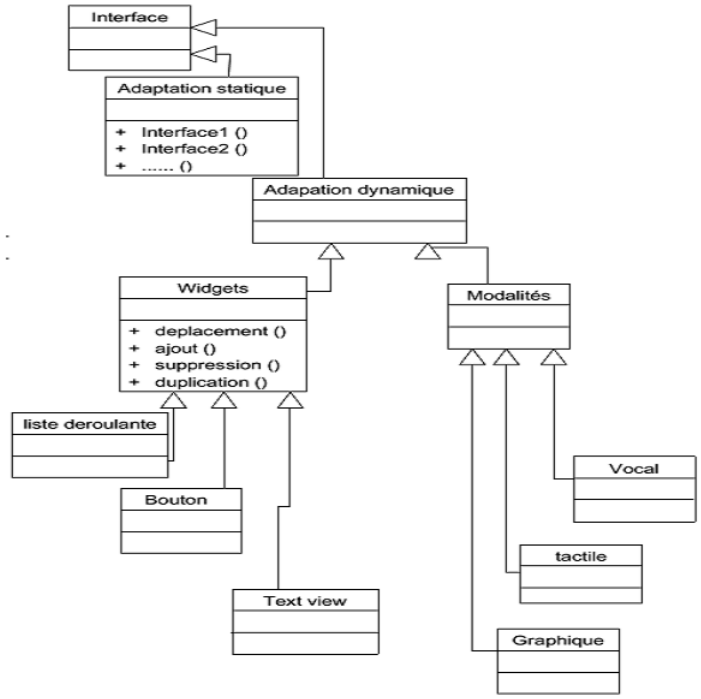


FIGURE 3.4 – diagramme de classe illustre l'adaptation de l'interface .

2. **Adaptation du contenu** Les médias se classent en deux catégories : les médias caractérisés par des dépendances temporelles (continus), tels que la vidéo et le son, et les médias discrets tels que les images et le texte. Chaque type de média a un ensemble de formats d'encodage et possède un certain nombre de propriétés spécifiques comme le titre (dans audio/vidéo par exemple), la police (dans le texte par exemple), la taille (image), etc. On utilise quatre types de liens structurels entre médias : hypermédia (possibilité de navigation dans le document multimédia), logique (permet de décrire l'organisation logique d'un flux sous forme d'une hiérarchie de médias), temporel (synchronisation des objets multimédia dans le temps), et spatial (disposition des objets multimédia dans l'espace d'affichage). La structure des médias est exprimée en UML à l'aide d'un diagramme de classe de la figure 3.5.

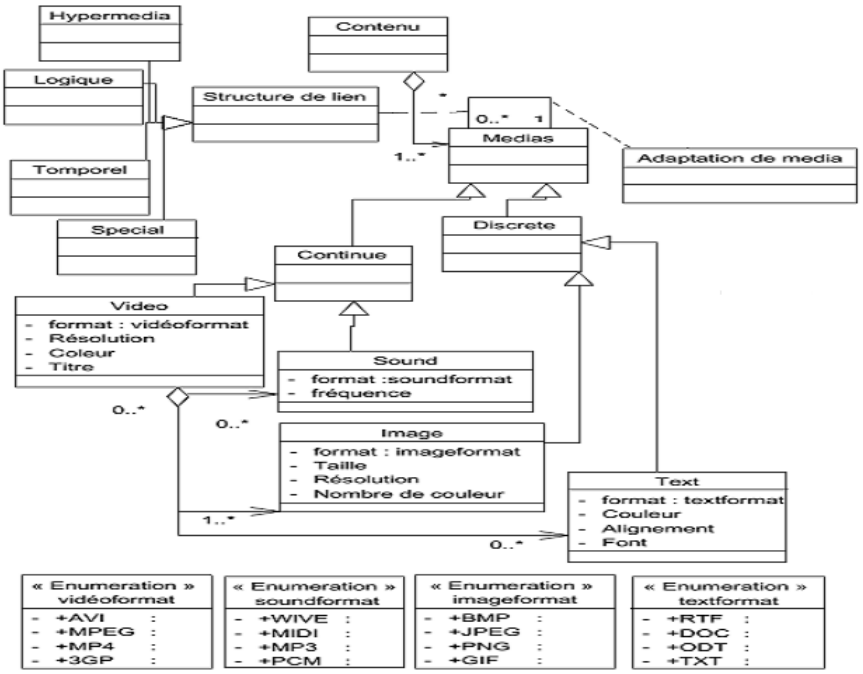


FIGURE 3.5 – diagramme de class illustré l'adaptation du contenu.

- **Adaptation de media** Nous classifions les traitements d'adaptation de médias en trois catégories :
  - **Le Transmodage :** le trasmodage permet de changer la modalité des médias. Par exemple, la transformation d'une vidéo en Sound.
  - **Le Transcodage :** le transcodage permet de changer le format de codage des médias. Par exemple, l'image peut être transcodée du

format PNG au format JPEG.

– **La Transformation :** Ce processus transforme le contenu, par exemple en change la police d'un texte.

Le diagramme de la figure 3.6 présenté l'adaptation de media proposer par DERDOUR.M [DER12]

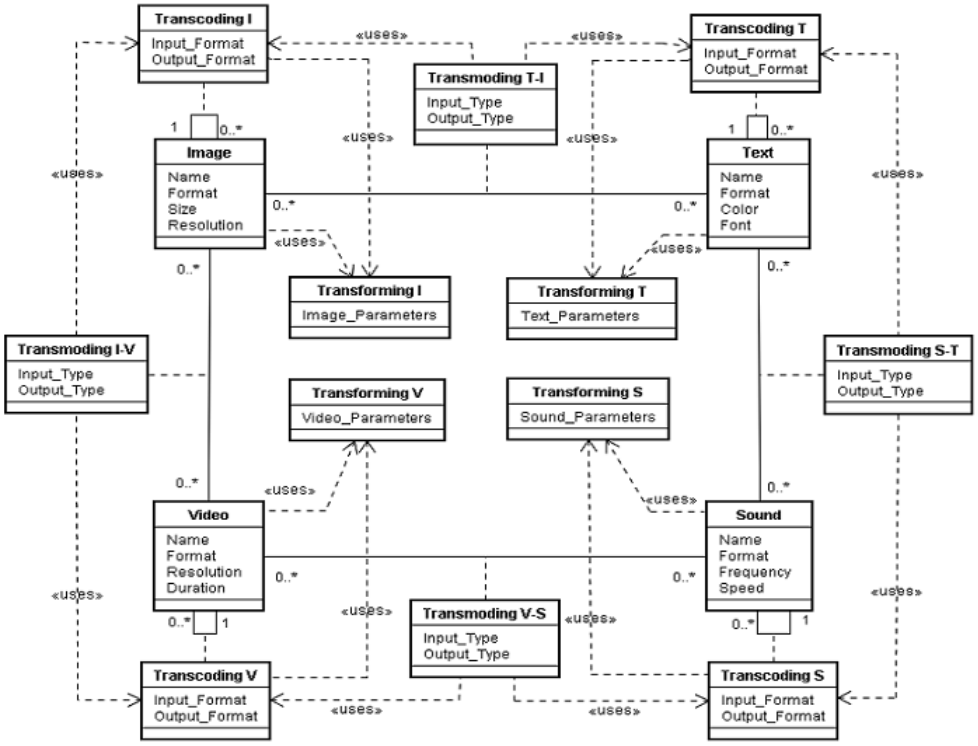


FIGURE 3.6 – diagramme de class présenté adaptation de media.

Ce diagramme montre les différentes classes d'association permettant le passage d'un type de média à un autre, d'un format de média à un autre, ou d'une propriété à une autre. La relation entre la classe d'association Transmodage et la classe d'association Transcodage exprime que la classe Transmodage peut faire appel à la classe Transcodage pour accomplir sa tâche. Il en va de même entre la classe Transcodage et la classe Transformation. La classe média (son, vidéo, texte, image) utilise la classe Transformation pour changer les caractéristiques de média. Cette relation indique que chaque format de média a un ensemble de paramètres pour gérer les différentes qualités. La transformation est un type particulier d'adaptation qui garde le même type et le même format de média avec des changements de caractéristiques (exemple :

mise en noir et blanc d'une image JPEG)[DER12].

3. **adaptation du service** On utiliser pour l'adaptation du service :
   - **Reconfiguration du service :** configurer et reconfigurer les services au besoin d'utilisateur
   - **Réassemblage du service :** réassembler les services au besoin d'utilisateur

   La déférence entre Reconfiguration et réassemblage est :
   - **Reconfiguration :** utiliser pour configurer les service à partir de zéro ,en utiliser généralement la reconfiguration si nous voulons faire un grand changement dans les services.
   - **Réassemblage :** utiliser si on a un petit changement par exemple ajouter, supprimer un service, sans toucher les autres services

   Et l'adaptation de service représent par le diagramme de la figure 3.7

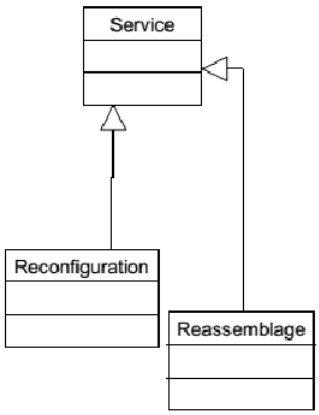


FIGURE 3.7 – representation d'adaptation des services.

La figure 3.8 résume les trois premières étapes de la méthode

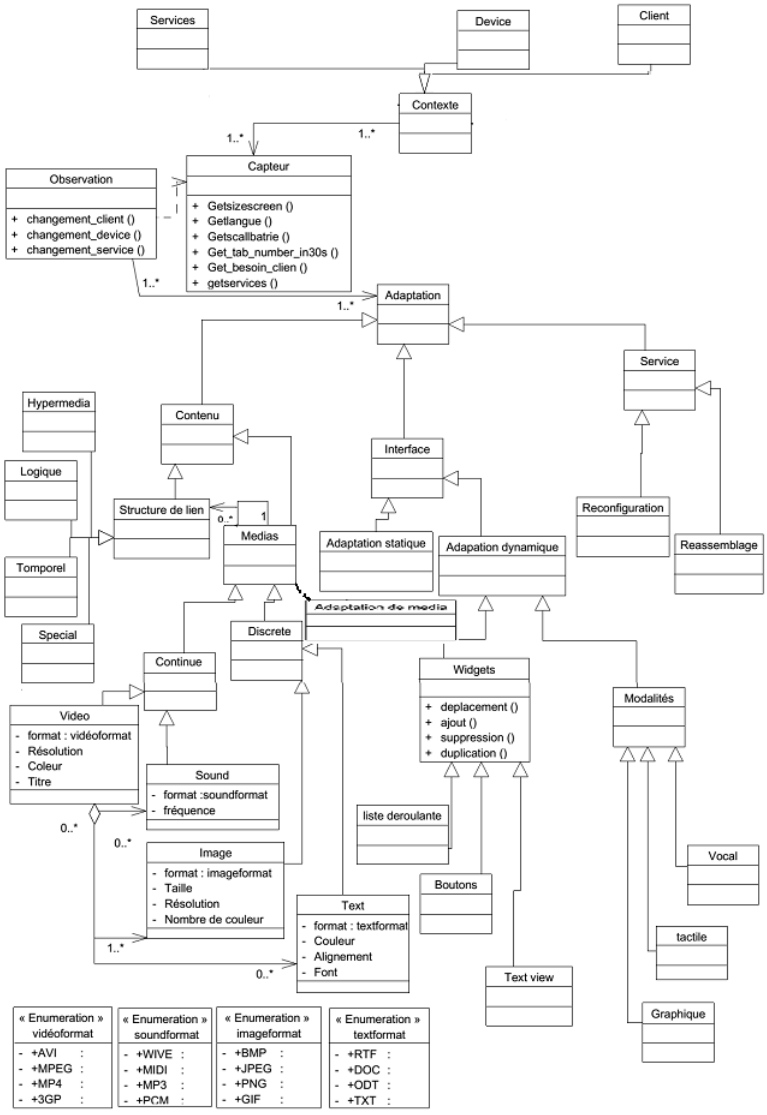


FIGURE 3.8 – diagramme de class représenter Les trois premières étapes de la méthode.

Etape 4 : évaluation de l'adaptation de l'interface :Vérifier l'utilité et l'utilisabilité de l'interface .

selon la satisfaction de l'utilisateur ,nous deciderons de terminer le processus ou retourner vers l'étape 1
Si l'utilisateur est satisfait :aller à étape 5
si non le système va recomancer le processus d'adaptation

Etape 5 :deploiement de l'interface adaptée

cette étape est très importante car elle assure le passage entre la conception de l'adaptation et la mise en service de cette dérnière. elle permet d'établire les liens de comunication entre l'application et les parties ajoutées (widget, service, composant, modalité, ...).

3.3 Conclusion

Dans ce chapitre, nous avons présenté notre méthode de conception d'interface graphique dynamique adaptable au nouveau contexte d'utilisateur. Dans notre proposition le contexte est défini par les paramètres liés avec :

- L'utilisateur
- Devise
- Service

Nous avons illustré la méthode proposée avec une démarche très simple a implémenter, et très efficaces puisque nous avons divisé la partie d'adaptation en trois sections pour garantir la fiabilité de l'interface :

- Adaptation d'interface
- Adaptation contenue
- Adaptation service

Chaque section de la méthode proposée est illustrée par un diagramme de classe et enfin nous avons terminé par un diagramme de classe décrivant la structure générale de la méthode proposé.
nous allons essayer, dans le chapitre suivant, d'utiliser notre méthode de conception pour arriver à développer une application adaptable aux besoins d'utilisateur, et en distribuant cette dernière en multi-users et multi-devices
.

Chapitre 4

Etude de Cas

4.1 introduction

Dans ce chapitre, nous allons citer deux scenarios pour nos cas d'étude, et on présente la liste des outils et les langages de programmation que nous avons réunis pour l'implémentation et la réalisation de notre application du cas d'étude, en suite nous utilisons la méthode proposée dans le chapitre précédant pour la conception de nos cas d'étude, et nous présenterons dans la troisième étape de la méthode proposé " Conception et proposition de l'interface adaptable aux changements " quelques exemples des interfaces représentants les interfaces de l'application qui ont été réalisées.

4.2 Outils de développement

1. **Environnement matériel :** Pour la réalisation de l'application :
   - PC HP CORE 2 DUO avec Windows7 32bit.
   - Smartphone et Tablet pour réaliser les tests
2. **Environnement logiciel Android :** Android est un système d'exploitation mobile pour les Smartphone, les tablettes tactiles, ...etc. C'est un système open source qui facilite de plus en plus la création d'applications mobiles avec un langage de programmation " JAVA " connu par des milliers de développeurs à travers le monde
3. **Environnement de développement Android studio** Android Studio est l'environnement de développement officiel de Google qui remplace l'IDE d'Eclipse (avec donc exactement les mêmes fonctionnalités)

depuis le 8 décembre 2014. Dans notre cas d'étude nous avons utilisé la version 2.2.1

4.3 Scéario et implémentation

Scénario 1 :
Une interface qui contient par exemple un lecteur de vidéo en streming qui entrain de présenter ou d'afficher le vidéo. L'utilisateur n'arrive pas à comprendre la langue, il cherche un composant de sous-titrage en ligne.

Après une recherche chez les fournisseurs de services, Il trouve le composant et il l'insert.

L'utilisateur a besoin des coordonnés GPRS, il cherche un service qui fournit cette information, ce service se trouve chez un appareil voisin et il est libre en service, l'appareil de l'utilisateur récupère les coordonnées à partir de l'appareil voisin et mis à jours l'affichage au directement sur l'appareil utilisateur. Elle récupère ca a partir une service offre par l'appareil voisin.

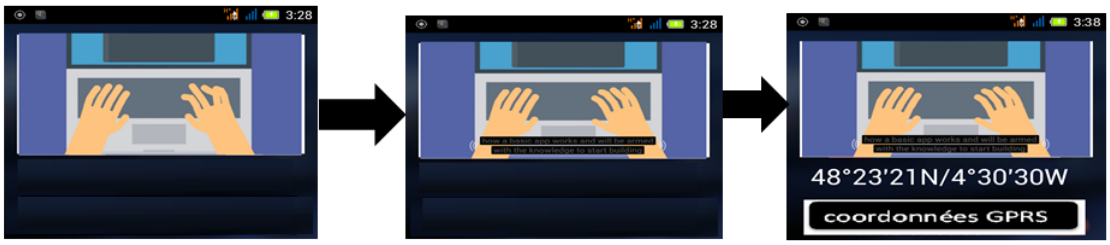


FIGURE 4.1 – Scénario 1

Scénario 2 :

Nous allons considérer l'étude de cas suivante, pour illustrer notre méthode : Un apprenant décide de s'informer sur les réseaux informatiques, et il n'a pas trouvé un moyen efficace, alors, nous avons proposé une application mobile qui explique les réseaux informatique d'une façon très simple en utilisant les MOOC (Massive Open Online Courses). L'application proposée contient des interfaces dynamiques (adaptables et adaptatives), cette dernière, est la chose la plus importante dans cette application.

" Quand un apprenant utilise l'application, l'interface de cette dernière s'adapte automatiquement à la langue du système. Cependant, nous avons proposé deux langues : Français et anglais. " Alors, si la langue de système

est " française ", l'interface adapte à la langue française sinon anglaise. L'interface qu'est affiché sur l'écran, contient deux champs ; le premier est pour le nom d'utilisateur, le deuxième est pour le mot passe et un bouton pour vérifier les informations entrée dans les deux champs précédant, et finalement un bouton pour l'enregistrement dans l'application. Dans le cas où l'apprenant entre un nom d'utilisateur et un mot de passe valide, l'application propose l'interface adaptable par rapport la taille d'écran. " Dans le cas d'une grande interface (tablette par exemple) une liste des cours réseau informatique sera afficher dans la partie gauche de l'écran et le cours sélectionner s'affichera dans la partie droite.

" Si on prend le cas d'un (Smartphone), la liste des cours sera affichée sur l'écran, si nous choisissons un cours de la liste, nous nous dirigerons vers une autre interface qui contient le cours choisie.

Remarque : Comme vous le savez les MOOC se basse beaucoup plus sur les vidéos mais avec la connexion (2G) par exemple il sera impossible d'ouvrir la vidéo des fois, c'est pour cela, nous avons proposé d'adapter la vidéo en format MP3 dans le cas d'une connexion (2G), Cette adaptation mis en place si elle est approuvée par l'apprenant, par la rotation de l'écran du dispositif.

4.4 Mise en oeuvre : application de notre méthode sur le scénario 2

nous considerons le 2 eme scénario et nous essayons d'applique les étape de notre méthode sur le scénario 2 . Notre cas d'étude, malgré sa simplicité, offre diverses propriétés importantes pour examiner notre méthode de conception, surtout les propriétés de l'interface dynamique (adaptabilité, adaptativité).

- **Adaptabilité :** L'utilisateur accepter l'adaptativité de l'interface au type de connexion ou bien n'accepte pas cette adaptation.
- **Adaptativité :** l'application s'adapter au contexte automatiquement :
  - adapter la langue de l'interface avec langue de système.
  - adapter la présentation de l'interface avec longueur de l'écran.

Et la figure 4.2 exprime le diagrame de classe suivé durant l'utilisation de notre example :

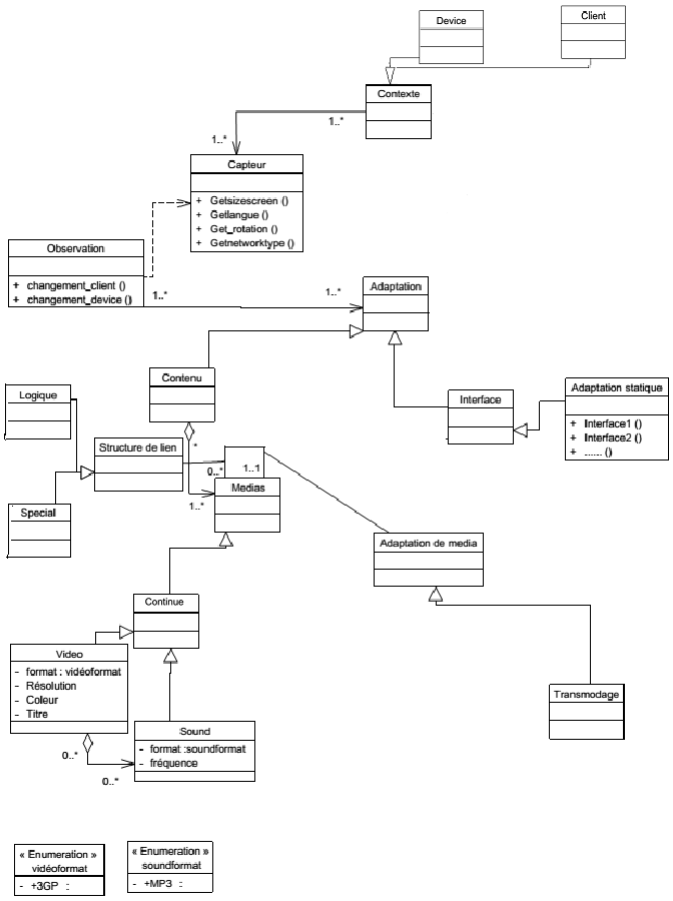


FIGURE 4.2 – diagramme de class suivé durant l'utilisation de notre example .

Etape 1 : recupération des informations du contexte et besoin de l'utilisateur : Capter les informations du contexte et les nouveaux besoins des utilisateurs

Capter les contextes du DEVICE :

Dans notre cas d'étude nous avons besoin de détecter les trois variations dans la plateforme de DEVICE telle que :

- la taille de l'écran.
- la langue de système.
- Le type de la connexion internet.

Capter les besoin de l'utilisateur :

Acquisition à partir de la rotation de l'écran, s'il existe un changement au niveau du type de connexion internet, si l'apprenant favorisé l'adaptation de la contenu au nouveau type de connexion, l'apprenant tourner l'écran.

Etape 2 : analyse des changement aux besoins et au contexte : détecter les changements aux besoins et aux contextes par le composant observateur.

Le patron de conception " Observer " permet de détecter tout changement lié à la plateforme ou à la besoin de l'utilisateur.
Par exemple le changement de la langue de système, rotation de l'écran, changement au niveau de la type de connexion .

Etape3 :Conception et proposition de l'interface adaptable aux changements

Dans notre cas d'étude la conception de l'interface graphique dynamique elle se base sur deux type d'adaptation :

1)Adaptation de l'interface :

Dans notre cas d'étude on a deux Contexte et un besoin lie avec l'adaptation de l'interface :

- Langue de système : pour crée une interface utilisateur compatible avec la langue système.
- la taille de l'écran : pour affiche un écran duplique si ce dernier de grand taille.
- Rotation de l'écran : pour déterminera si l'utilisateur favorise l'adaptation de contenu a la nouvelle type de connexion, ou bien non .

A)Adaptation de l'interface a la langue de système :

Pour accéder à notre application l'apprennent doit s'authentifier.

- Si apprenant utilise le système android avec la langue française en déclenche qui produit une interface s'adapte à l'apprenant qui la langue précédente.

- Si apprenant utilise le système android avec autre langue appart la langue français en déclenche la tâche qui produit une interface s'adapte à l'apprenant qui la langue précédente.

Les deux figures 4.3 et 4.4 donnent les interfaces à travers laquelle l'apprennent s'identifie tel que la première interface s'adapter à l'utilisateur qui utilise le système android avec la langue français et la deuxième s'adapte a l'utilisateur qui utilise autre langue.



FIGURE 4.3 – interface avec la langue française.

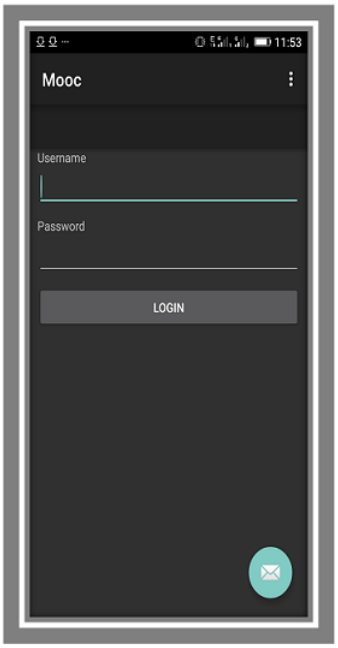


FIGURE 4.4 – interface avec la langue Anglaise.

Les deux figures 4.5 et 4.6 représentent les interfaces à travers laquelle l'apprennent d'enregistrer tel que la première interface s'adapter à l'utilisateur qui utilise le système android avec la langue français et la deuxième s'adapte a l'utilisateur qui utilise autre langue.



FIGURE 4.5 – interface d'inscription avec la langue française.

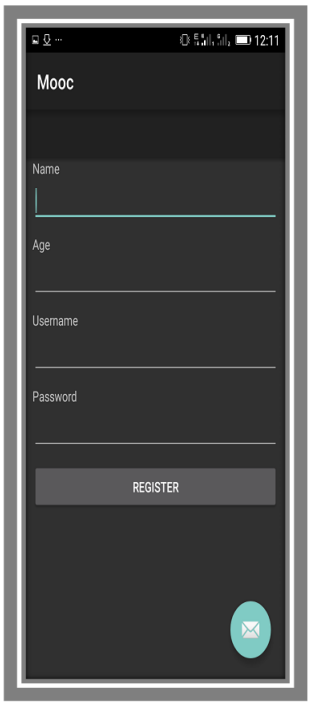


FIGURE 4.6 – interface d'inscription avec la langue Anglaise.

B)Adaptation de l'interface a la taille de l'écran :

Apre l'inscription, le système affiche l'interface d'authentification automatiquement, l'apprenant antre leur " nom d'utilisateur " et leur " mot de passe " est connecter à l'application.

- Si l'utilisateur utilise un terminal avec un écran petit taille en déclenche la tâche qui affiche l'interface de l'écran petit taille.
- Si non en affiche l'interface de l'écran grand taille.

Les deux figures ci-dessus représentent deux interfaces la figure 4.7 pour les écrans de taille petite et la figure 4.8 pour les grands écrans.



FIGURE 4.7 – interface adapter à écran de petite taille .

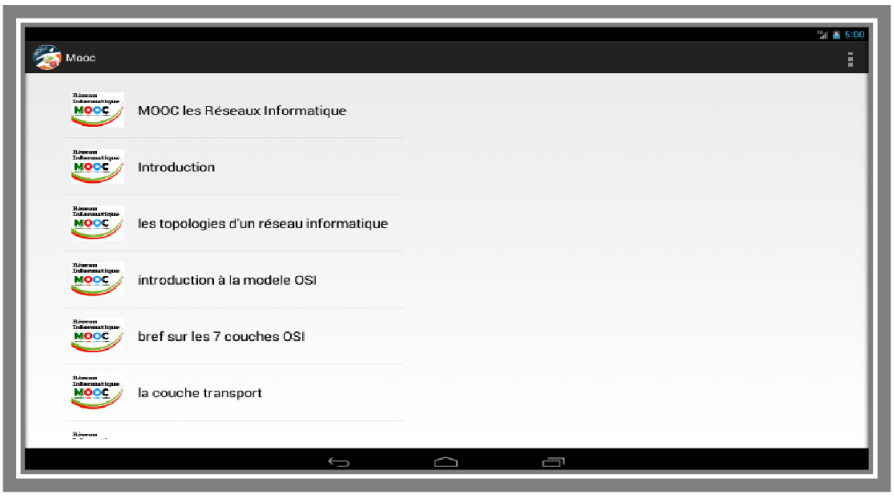


FIGURE 4.8 – interface adapter à écran de grande taille .

2)Adaptation de contenu :

Dans notre cas d'étude en choisi le type de connexion comme une Contexte lie avec l'adaptation de contenu :

- Le type de la connexion internet : pour affiche des contenue supportable avec le type de la connexion.

Nous avons adapté la vidéo en format MP3 dans le cas d'une connexion (2G). Cette adaptation mis en place si elle est approuvée par l'apprenant, à partir de la rotation de l'écran.

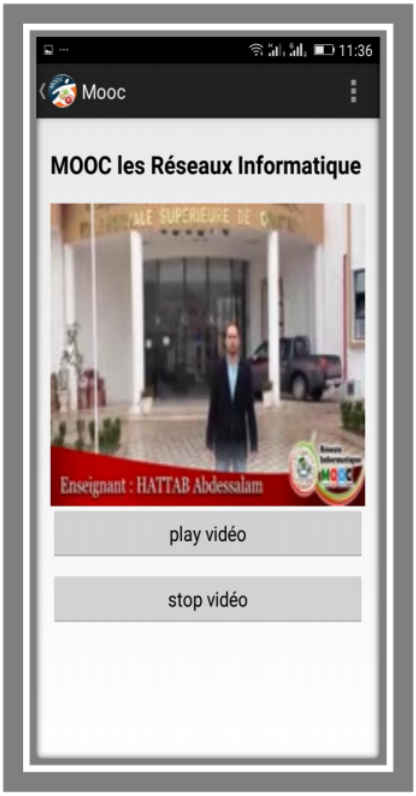


FIGURE 4.9 – adaptation de contenu au type de connexion 3G en écran petit taille.



FIGURE 4.10 – adaptation de contenu au type de connexion 2G en écran petit taille.

Les deux figures 4.9 et 4.10 en dissus représentent l'adaptation de contenu par rapport le type de la connexion ou l'utilisation d'un éeran de petite taille.

Les deux figures 4.11 et 4.12 représentent l'adaptation de contenu par rapport le type de la connexion ou l'utilisation d'un éeran de grande taille.



FIGURE 4.11 – adaptation de contenu au type de connexion 3G en écran de grande taille.

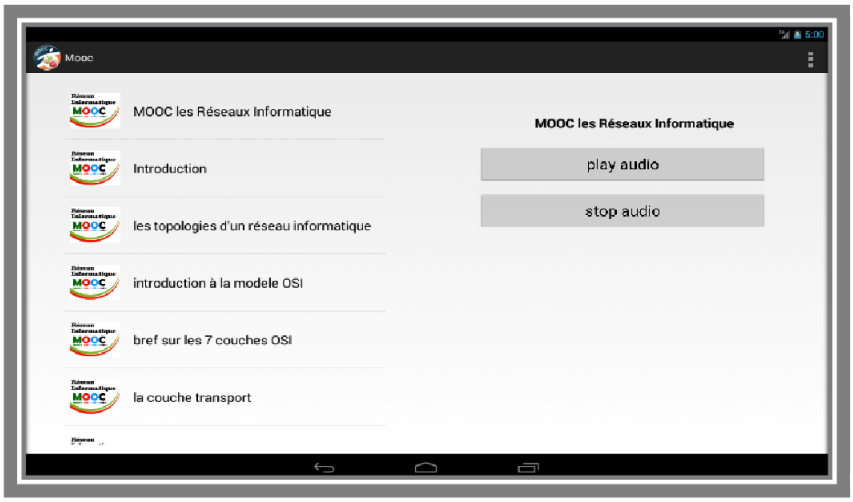


FIGURE 4.12 – adaptation de contenu au type de connexion 2G en écran de grande taille.

Etape 4 : évaluation de l'adaptation de l'interface :Vérifier l'utilité et l'utilisabilité de l'interface

Après la proposition de l'interface par l'application l'apprenant évaluer l'interface, si l'interface répond à la besoin, l'apprenant utilise l'application sans changement, si non l'apprenant changer dans l'interface. Dans notre cas si l'apprenant adapté l'audio vers la vidéo il peut basculer vers l'audio avec le changement de la type de connexion et la rotation de la diapositive.

Etape 5 :deploiement de l'interface adaptée

4.5 Conclusion

Ce chapitre a été consacré à la présentation de notre cas d'étude, ce dernier démontrer la fiabilité de notre méthode de conception, Nous avons mis en exergue les différentes interfaces de notre application, la plus par des interfaces sont exposées pour mettre l'accent sur l'aspect d'adaptation que nous considérons important dans la conception de nos interfaces.

Conclusion générale

La recherche dans le domaine des interfaces dynamique attire l'attention d'un grand nombre des chercheurs dans différents domaines.
Bien que les chercheurs proposent plusieurs méthodes de conception des interfaces dynamiques, les méthodes proposées présentent plusieurs limites tels que : ne prend pas en compte les caractéristiques de l'appareil, la connexion et la distribution des services, l'absence de la combinaison entre l'adaptation des interfaces par rapport au contexte et la modalité (tactile, vocale, ect...) d'une part et la combinaison entre adaptation manuelle et automatique d'autre part.
Nous avons proposé une méthode de conception d'interface graphique dynamique qui contient des widgets (listes dérolantes, boutons, textview, imageview, videoview...) avec des propriétés liées à l'adaptation en temps réel (ajoute, suppression, déplacement), ainsi nous avons utilisé déférèrent modalité (tactile, vocale, ect.), cette méthode permet d'adapter le contenu des interfaces aux besoins réels de l'utilisateur.
Après la proposition de la méthode, nous avons illustré l'utilisation de cette dernière pour plusieurs scénarios.
Malgré notre proposition enveloppée par la méthode de conception des interfaces dynamique distribuées et qui prend en compte un ensemble des besoins d'adaptation dynamique d'interface. Il nous reste beaucoup de pistes de recherches à découvrir. Parmi ces pistes, la prise en compte de la dynamique par rapport au changement de la plateformes de développement et aux changement de contexte en temps réel .

Bibliographie

[BER04] S.Berti ,F.Correani, G.Mori, F. Paterno, et C. Santoro, Teresa : a transformation-based environment for designing and developing multi-device interfaces,In CHI Extended Abstracts'04, pages 793-794, (2004).

[BOU10] W.Bouchelligua, A.Mahfoudhi, L.Benammar, S.Rebai. et M.Abed. (2010),An mde approach for user interface adaptation to the context of use. In HCSE, volume6409 de Lecture Notes in Computer Science, pages 62-78. Springer.

[BRO 98] Brown et all .(1998). Triggering information by context. In personnal Technologies. v. 2, p. 19.

[CAL03] G.Calvary, et all .A unifying reference framework for multi-target user interfaces. Interacting With Computers, Vol. 15/3, 289-308.(2003).

[CER04] H.CERVANTES, " Vers un modèle a composants orienté services pour supporter la disponibilité dynamique ", pour obtenir le grade de DOCTEUR de l'Université Joseph Fourier,29/03/2004.

[CHA07] T.chaari, " Adaptation d'applications pervasives dans des environnements multi-contextes ", Le grade de docteur en informatique, L'institut national des sciences appliquées de Lyon 28/09/2007.

[CHE00] G.CHE et D.KOTS , ,survey of context-aware mobile computing research , 2000.

[CHE 14] M .CHEBBI " utilisation des patrons de conception pour l'adaptation des interfaces graphiques des applications mobiles dépendantes du

contexte ", mémoire présenté comme exigence partielle de la maîtrise en informatique, Université Montréal, juillet 2014.

[DEY00] A.Dey et G.Abowd Towards a better understanding of context and context-awareness. In Computer Human Interactions (CHI2000). Workshop on the What, Who, Where, When and How of Context-Awareness, 2000.

[DEY01] A.Dey., G.Abowd, et D.Salber. (2001). A conceptual framework and toolkit for supporting the rapid prototyping of context-aware applications. Human-computerInteraction, 16(2-4 (special issue on context-aware computing)) :97-166.

[DER12] M.DERDOUR, thèse doctorat en Sciences , UNIVERSITE BADJI MOKHTAR-ANNABA,(ALGERIA),2012.

[HAC09] S.Hachani et al . Une approche générique pour l'adaptation dynamique des IHM au contexte . In IHM, ACM International Conférence Proceeding Series, pages 89-96. ACM.

[JOH97] Johnson, R.E. (1997). Frameworks=(components+ patterns). Communications of the ACM, 40(10), 39-42.

[LAM04] T.LAM, thèse de docteur de L'INPG , L'INSTITUT NATIONAL POLYTECHNIQUE ,GRENOBLE ,(France) 2004.

[LAY99] N.Layaïda. adaptabilite pistes d'etude pour la définition d'une infrastructure d'accés au contenu multidédia pour des machines hétérogènes.rapport techniques .Grenoble :INRIA Rhone-Alpes,octobrs 1999.

[LEM01] T.Lemlouma et N.layaida .A Framework for Media Resource Manipulation in an Adaptation and Negotiation Architecture.OPRA project.Grenoble :INRIa Rhone-Alpes,august 2001.

[MEN02] G.Mankhaus . Adaptive User Interface Generation in a Mobile Computing Envirenment.PHD this Austria :Salzburg University,2002.

[MIR09] M.MIRAOUI, " architecture logicielle pour l'informatique diffuse : modélisation du contexte et adaptation dynamique des services ", pour obtenir diplôme de doctorat université de Québec, 14/07/2009.

[ORE99] P.Oreizy et Al . An architecture-based approach to self-adaptive software,IEEE Intelligent Systems, 14(3) :54-62, Mai/Juin 1999.

[SAT01] M.Satyanarayanan, Pervasive Computing :Vision and challenge .IEEE Personnal Communications ,August 2001 ,Vol.8,N°4,pp.10-17.

[SIM07] J.Simonin ,thèse Doctorat, Aide en ligne adaptative et assistants conversationnels animés : mise en oeuvre et évaluation ergonomique,Nancy-université,2007,Frace.

[SOT08] Sottet, J.-S. (2008). Méga-IHM : malléabilité des Interfaces Homme-Machine dirigées par les modèles. Thèse de doctorat. Thèse de doctorat Informatique préparée au Laboratoire d'Informatique de Grenoble (LIG), Université Joseph Fourier.

[THE01] D.Thevenin (2001). L'adaptation en Interaction Homme-Machine : le cas de la plasticité. Thèse de doctorat. Thèse de doctorat Informatique préparée au Laboratoire de Communication Langagière et Interaction Personne-Système (IMAG), Université Joseph Fourier 238 pages.

[VEY14] Veytizou.J, Guillaume Thomann et François Villeneuve. De la caractérisation des spécificités motrices vers la conception d'interfaces personnalisables . Handicap 2014 Les technologies d'assistance : de la compensation à l'autonomie, Jun 2014, Paris, France. 6 p., 2014.

[VIN08] Vincent Ganneau,Gaëlle Calvary , Rachel Demumieux . EMMA : Modèle Utilisateur pour la Plasticité des Interfaces Homme-Machine en Mobilité. Actes des quatrièmes Journées Francophones sur la Mobilité et l'Ubiquité, UBIMOB'08, pp. 1-8

[1] http ://www-sop.inria.fr/members/Thierry.Vieville/cours/adaptation.html

[2] http ://www.pierrelucgrenier.com/2011/ll/22/mobilite-quelques statistiques/

[3] Le grand dictionnaire terminologique

[aa] @ArticleLiris-2996, title = Adaptation in Context-Aware Pervasive Information Systems : The SECAS Project, author = Tarak Chaari and Frédérique Laforest and Augusto Celentano, year = 2007, month =

dec, journal = Int. Journal on Pervasive Computing and Communications(IJPCC), volume = 3, number = 4, pages = 400-425, publisher = Emerald Group Publishing Limited, DOI = 10.1108/17427370710863130, language = en, url = http ://liris.cnrs.fr/publis/ ?id=2996