



Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Création d'applications mobiles via une application Java

Réalisé par :

- **KAID SLIMANE Imane**
- **GHEMBAZA Hayat**

Présenté le 27 mai 2015 devant la commission d'examination composée de MM.

- **SMAHI M. I.** (Encadreur)
- **BENAZZOUZ M.** (Examineur)
- **BENMOUNA.Y** (Examineur)

Année universitaire : 2014-2015



Dédicaces

*Nous dédions modestement le fruit de nos longues années d'étude tout
d'abord :*

**A nos chers parents, que nulle dédicace ne puisse exprimer ce qu'on
leurs doit, pour leurs bienveillance, affection et leur soutien Morale
mérite tous nos respect.*

*On vous dit merci pour tout ce que vous avez fait et de ce que vous nous
ferez plus tard, que dieu vous protège, vous laisse toujours devant nos
yeux et vous réserve une langue vie.*

** A tous nos sœurs et frères.*

** A toute nos grandes famille : oncles, tantes, cousines, et cousins.*

**A toutes mes ami(es), spécialement YAHLA reda.*

**Sans oublier toute la promotion de 3^{ème} Année Inf. /2015*

Avec qui on a partagé des très bons moments et tous ceux qui

Connaissent KAID SLIMAN IMANE et GHEMBAZA HAYAT

Remerciements

En préambule à ce mémoire nous remercions le bon Dieu qui nous aide et nous donne la patience et le courage durant ces longues années d'étude.

On tient à remercier vivement Monsieur

Smahi.M.I

Pour nous avoir conseillé et dirigé tout au long de la thèse ainsi que pour ses nombreux conseils avisés lors de la rédaction de ce mémoire. Qu'il trouve ici notre profonde reconnaissance pour la confiance qu'il nous a témoignée durant l'année.

Avec les remerciements aux membres du jury

Mr BENMOUNA.M et Mr BENZAZZOUZ.M

Pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critiques.

A tous mes enseignants qui m'ont initié aux valeurs authentiques, en signe d'un profond respect et d'un profond amour !

Enfin, je tiens à exprimer toute ma gratitude à tous ceux qui ont contribué de près ou de Loin à l'accomplissement de ce mémoire.

RESUME

Au cours de ce modeste travail, nous avons comme objectif principal de réaliser une application JAVA Desktop en respectant l'hierarchie de l'approche IDM. La particularité de notre travail était dans la manière d'établir une liaison entre l'abstraction de l'ingénierie dirigé par les modèles et de la concrétisé en une application. L'objectif principal est le développement d'une application qui permet aux « utilisateurs ordinaires » de générer des applications mobiles sans passer par le code source de manière à ce que cette application a comme entrée le modèle de l'utilisateur 'sous code java' et comme sortie un dossier exécutable qui contiendra des fichiers XML et JAVA ainsi que tout les packages nécessaires .

Mot clé ingénierie dirigée par les modèles, Android.

Abstract

In the courtyard of this modest work we had as main objective to achieve a java desktop application respecting the IDM approach hierarchy. The peculiarity of our work was in the way of establishing a connection between the abstraction of the model-driven engineering and embodied in an application. The main objective is the development of an application that allows "ordinary users" generate mobile applications through its source code so it is that this application has as input the user model (in java code) and output as an executable file that will contain JAVA and XML files and any packages needed .

ملخص

خلال مشروعنا تم التحدث عن مجال التطبيقات النقالة وكان هدفنا الرئيسي هو سماح للمستخدمين العاديين تحقيق التطبيقات النقال عن طريق سطح المكتب وذلك باعتمادنا علي نهج الهندسة بنماذج.

TABLE DES MATIERES

Résumé	5
table des figures	8
Introduction général	9
CHAPITRE I: PROGRAMMATION AVEC LE SYSTEM ANDROID	10
Introduction	11
Formation de développement mobile	11
1. L'utilisation des applications mobiles.....	11
2. Besoins de compétence en développement d'application mobile.....	11
L'univers Android	12
1. Introduction.....	12
2. Les avantages D'ANDROID [3].....	13
3. Architecture d'Android [4].....	14
4. Prise en main de l'environnement ANDROID [4]	16
5. Les éléments d'uneapplication Android[3].....	17
Conclusion	17
Chapitre II :l'ingénierie dirigée par les modèles(IDM)	18
Introduction	19
Les principes généraux de l'IDM	19
1. Qu'est-ce que l'IDM ?.....	19
2. Module et Méta-MODÈLE [6]	19
3. Pourquoi modéliser et Comment modéliser ?.....	20
4. Métamétamodèle et les niveaux OMG[6]	20
Transformation de modèles	21
Approche MDA[8]	22
1. La relation de IDM avec MDA.....	23
Vue général sur l' IDM	24

Conclusion	25
ChapitreIII :Réalisation d'application « AppMaker »	26
Introduction	27
outils de travail	27
1. Environnement du travail.....	27
2. Les Librairies des Programmation utilisée.....	27
Objectif	28
1. Vue générale de la démarche de l'application avec la concept IDM	28
Architecture générale	31
Interface principale	33
Fonctionnement d'application	33
Conclusion générale	35
<i>BIBLIOGRAPHIE</i>	36

TABLE DES FIGURES

Figure 1: les principaux champs pour développement d'application mobile.....	12
Figure 2:le logo d'open handset alliance [2]	13
Figure 3 : La partie d'application	14
Figure 4: La partie d'application Framework.....	14
Figure 5 : La partie librairies.....	15
Figure 6: la partie d'Android Runtime.....	15
Figure 7: la partie Linux Kernel	15
Figure 8: Pyramide de modélisation de l'OMG [7]	21
Figure 9:transformation des modelés [8].....	24
Figure 10:la vue général de l'application.....	28
Figure 11:la classe de Java de l'interface.....	29
Figure 12:capture d'écran d'un fichier XML.....	30
Figure 13: le projet Android générer	30
Figure 14:capture d'écran de l'application sous mobile.....	31
Figure 15 : la structure générale de l'application.....	31
Figure 16: cas d'utilisation	32
Figure 17: diagramme de séquence du déroulement d'application	32
Figure 18:l'interface de l'application	33

INTRODUCTION GENERAL

Au cours des dernières années, les appareils mobiles sont devenus de plus en plus une combinaison de deux techniques : informatique et mobile. Ce qui va pouvoir remplacer dans l'avenir proche les ordinateurs portables.

Ceci dit les systèmes d'exploitation pour mobile sont devenus de plus en plus importants, notamment pour les entreprises, les développeurs et les fournisseurs d'applications pour mobile. Avec les commodités et les performances ainsi que l'évolution des outils de programmations mobiles, ils peuvent être avantageux pour les entreprises basées sur les technologies mobiles.

Dans le cadre de notre projet nous avons développé une application Java Desktop, qui permet aux " utilisateurs " ordinaires (non programmeurs) de générer leurs propres applications mobiles de façon simple, de sorte à ne pas écrire de code source.

Dans ce travail notre proposition vise à intégrer une piste pour le développement avec la notion de l'IDM de manière à modéliser le fonctionnement de l'application, de réaliser des transformations de modèles et générer un modèle cible à partir d'un modèle source.

Ainsi, nous articulons notre rapport autour de trois chapitres : le premier chapitre consiste à étudier la programmation mobile sous Android. Le deuxième chapitre aura comme but de présenter les démarches d'ingénierie dirigées par les modèles (IDM). Finalement, nous décrirons dans le dernier chapitre le fonctionnement de notre application baptisée « *Android AppMaker* ».

CHAPITRE I: PROGRAMMATION AVEC LE SYSTEM ANDROID

INTRODUCTION

Grace au divers système exploitation mobile existant de nos jours tel que : « Android» ou bien «Windows Phone » l'utilisation des applications ne cesse de s'accroître ce qui a fait l'explosion des ventes des smartphones ces dernières années.

Android s'est imposé dans le quotidien de nombreuses personnes, ce système d'exploitation permet l'installation des applications de tous genres (jeux, bureautique, multimédia,...). Dans ce chapitre on s'intéresse au développement mobile et ce géant des systèmes phone appelé « Android » ! Sa création émane d'une idée simple et voilà que son succès fut l'indispensable de beaucoup d'entre nous, plus précisément dans le milieu de la téléphonie mobile.

FORMATION DE DEVELOPPEMENT MOBILE

Les applications mobiles prennent de plus en plus de place dans le paysage numérique et les projets mobiles explosent. [1]

1. L'UTILISATION DES APPLICATIONS MOBILES

Le marché des applications mobiles est très important. Environ 7 % de ce montant est consacré à l'achat de nouvelles applications mises en marché récemment, alors que le reste va à d'autres applications moins récentes. Une augmentation significative des budgets des entreprises pour le développement et l'utilisation d'appareils mobiles amène évidemment une augmentation de la demande pour le développement d'applications mobiles.

2. BESOINS DE COMPETENCE EN DEVELOPPEMENT D'APPLICATION MOBILE

Il est possible de regrouper les principaux champs de compétences techniques requis pour une équipe de développement d'applications mobiles sous quatre thèmes différents. Dans chacun de ces groupes, un certain nombre de champs de compétences ont été identifiés comme spécifiques au développement d'applications mobiles, ce sont des champs compétences qui nécessitent l'acquisition de connaissances et de savoir-faire associés directement au développement d'applications mobiles. Ces champs de compétences spécifiques sont dans la figure suivant :

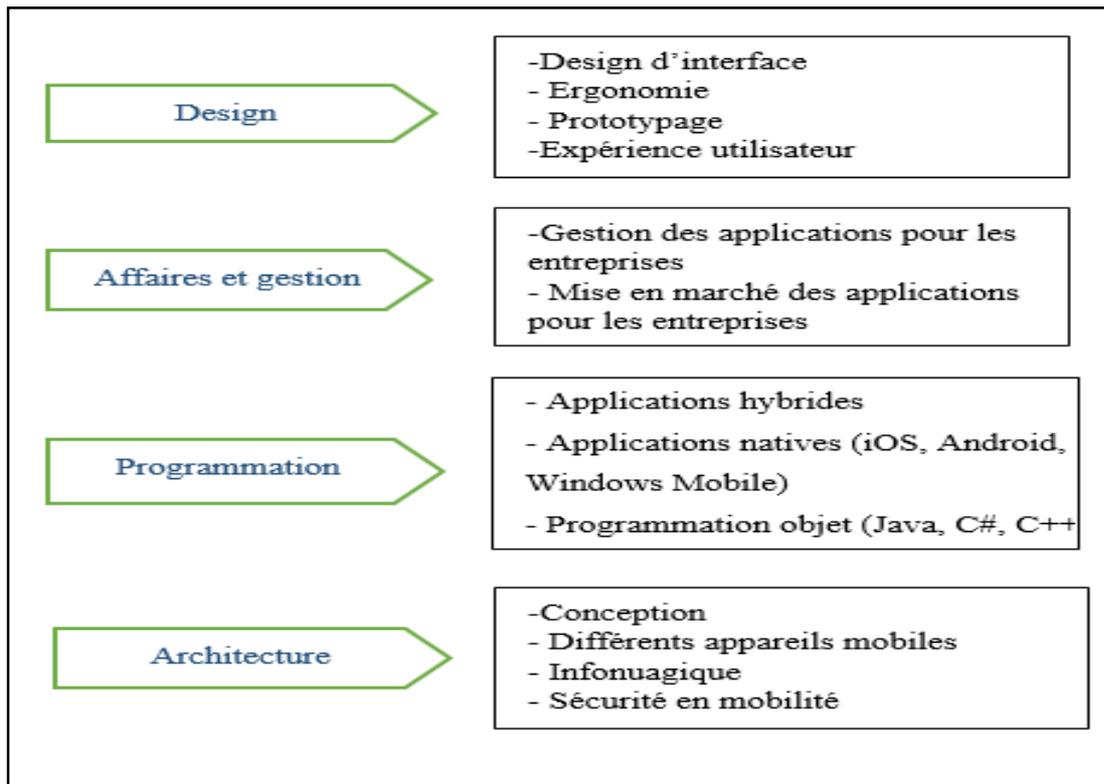


Figure 1: les principaux champs pour développement d'application mobile

Ce sont des champs de compétences nécessaires pour une équipe de développement d'applications mobiles, mais qui ne requièrent pas l'acquisition de connaissances et de savoir-faire associés directement au développement d'applications mobiles.

L'UNIVERS ANDROID

1. INTRODUCTION

Android est un système d'exploitation mobile adapté pour les écrans tactiles, et promu par Google et l'Open Handset Alliance, qui regroupe plus de 35 (fabricants, opérateurs et fournisseurs de logiciels). La Plate-formeAndroid fonctionnant sur GNU / Linux base source totalement gratuite et ouverte sous plusieurs systèmes Apache développer, de licence (SDK) et son code source est disponible sur Android depuis Novembre 2007.



Figure 2:le logo d'open handset alliance [2]

2. LES AVANTAGES D'ANDROID [3]

- Open source : Le contrat de licence pour Android respecte les principes de l'open source, c'est-à-dire que vous pouvez à tout moment télécharger les sources et les modifier selon vos goûts.
- Gratuit (ou presque) : Android est gratuit, autant pour vous que pour les constructeurs.
- Facile à développer : Toutes les API mises à disposition facilitent et accélèrent grandement le travail.
- Facile à vendre : Le Play Store (anciennement Android Market) est une plateforme immense et très visitée ; c'est donc une mine d'opportunités pour quiconque possède une idée originale ou utile.
- Flexible : Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android.
- Ingénieux : L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement.

3. ARCHITECTURE D'ANDROID [4]

-Pour bien comprendre la plateforme Android, nous détaillons par la suite l'architecture du système Android.

❖ Partie d'application

- Toute application Android repose sur un ensemble de services et systèmes parmi lesquels :
- Un ensemble de «Views» permettant de construire l'interface graphique de l'application
- Des «Content Providers» qui permettent aux applications d'accéder à des données d'autres applications ou de partager ses propres données,
- Un «Ressource Manager» pour accéder à des éléments autres que du code
- Un «Activity Manager» pour gérer le cycle de vie de l'application.



Figure 3 : La partie d'application

❖ Partie d'application Framework

-Le Framework d'application est la couche qui nous intéresse tout particulièrement. C'est elle qui fait le lien, grâce à un ensemble d'APIs Java, entre le système et l'application



Figure 4: La partie d'application Framework

❖ **Partie librairies**

-Android fournit un ensemble de librairies C/C++ utilisées par différents composants du système. Ces fonctionnalités sont rendues disponibles aux développeurs au travers du Framework d'application d'Android.



Figure 5 : La partie librairies

Partie d'Android Runtime

- Android inclut un ensemble de librairies fournissant la plupart des fonctionnalités des librairies standard de Java.



Figure 6: la partie d'Android Runtime

Partie Linux Kernel

- Android s'appuie sur le noyau Linux 2.6 pour les services système de base tels que la sécurité, la gestion de la mémoire et des processus, le réseau et la gestion des drivers. Le noyau sert de couche d'abstraction entre le matériel et le reste de la pile logicielle.



Figure 7: la partie Linux Kernel

4. PRISE EN MAIN DE L'ENVIRONNEMENT ANDROID [4]

Presentation du SDK

- SDK signifie Software Développement Kit, c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles, etc. pour un terminal et/ou un système d'exploitation spécifique. [5]

ADT pour Eclipse

- Eclipse est l'Environnement de Développement Intégré (ou IDE) le plus largement utilisé pour la programmation Java.
- Android Développement Tools, ou ADT, est très complet et surtout très pratique

Le plugin de développement d'Eclipse : est décomposé de la manière suivante :

- src/: les sources Java du projet
- libs/: bibliothèques tierces
- res/:
 - res/drawable: ressources images
 - res/layout: description des IHMs en XML
 - res/values: chaînes de caractères et dimensions
 - gen/: les ressources auto générées par ADT
 - assets/: ressources brutes (raw bytes)
- -bin/:
 - bin/classes: les classes compilées en .class
 - bin/classes.dex: exécutable pour la JVM Dalvik
 - bin/myapp.zip: les ressources de l'application
 - bin/myapp.apk: application empaquetée avec ses ressources et prête pour le déploiement.

ADV (L'Android Virtual Device)

- Est un émulateur de terminal sous Android, c'est-à-dire que c'est un logiciel qui se fait passer pour un appareil sous Android à votre ordinateur.

5. LES ELEMENTS D'UNEAPPLICATION ANDROID[3]

- Une application Android peut être composée des éléments suivants :
- des activités (android.app.Activity) : il s'agit d'une partie de l'application présentant une vue à l'utilisateur
- des services (android.app.Service) : il s'agit d'une activité tâche de fond sans vue associée
- des fournisseurs de contenus (android.content.ContentProvider) : permettent le partage d'informations au sein ou entre applications
- des widgets (android.appwidget.*) : une vue accrochée au Bureau d'Android
- des Intents (android.content.Intent) : permettent d'envoyer un message pour un composant externe sans le nommer explicitement
- des récepteurs d'Intents (android.content.BroadcastReceiver) : permettent de déclarer être capable de répondre à des Intents
- des notifications (android.app.Notifications) : permettent de notifier l'utilisateur de la survenue d'événements

CONCLUSION

Dans ce chapitre, nous avons fait une présentation du système d'exploitation Android en étalant ces fonctionnalités, son architecture ainsi que les principaux composants du système.

Nous avons aussi fait une vue générale sur le développement mobile. De ce fait nous entameront le prochain chapitre pour exposer l'approche IDM qui ce présente complémentaire permettent d'outiller notre application.

CHAPITRE II :L'INGENIERIE DIRIGEE PAR LES MODELES(IDM)

INTRODUCTION

Dans le chapitre précédent on avait parlé de l'univers d'Android avec ses différents outils et avantages qu'il fournisse

Ce deuxième chapitre fera l'objet d'une présentation de cette ingénierie. Nous introduirons dans un premier temps la notion de modèle, Nous détaillerons ensuite les deux axes principaux de l'IDM, et nous présenterons ensuite les principes de la transformation de modèle et les outils actuellement disponibles. Nous concluons enfin par une discussion sur les limites actuelles de l'IDM

LES PRINCIPES GENERAUX DE L'IDM

1. QU'EST-CE QUE L'IDM ?

- L'IDM (l'ingénierie dirigée par les modèles) est le domaine de l'informatique mettant à disposition des outils et langages pour créer et transformer des modèles.
- L'IDM correspond à l'utilisation systématique des modèles comme concept clé tout au long du processus d'ingénierie.
- L'IDM propose une automatisation du processus d'ingénierie, des modèles au code par affinements successifs des modèles.

2. MODULE ET MÉTA-MODÈLE [6]

- Définition (Modèle) : Un modèle est une abstraction d'un système, modélisé sous la forme d'un ensemble de faits construit dans une intention particulière.

Un modèle doit être suffisant et nécessaire pour permettre de répondre à certaines questions en lieu et place du système qu'il est censé représenter, exactement de la même façon que le système aurait répondu lui-même.

- Définition (Méta modèle) : Un méta modèle est un modèle qui définit le langage d'expression d'un modèle, il est effectivement indispensable d'associer à chaque carte la description du « langage » utilisé pour réaliser cette carte.

Ces deux relations permettent ainsi de bien distinguer le langage qui joue le rôle de système, du (ou des) méta modèle(s) qui jouent le rôle de modèle(s) de ce langage.

3. POURQUOI MODELISER ET COMMENT MODELISER ?

POURQUOI MODELISER ?

Un modèle permet de :

- Visualiser le système comme il est ou comme il devrait être
- Spécifier les structures de données et le comportement du système
- Fournir un guide pour la construction du système
- Documenter les systèmes et les décisions prises

COMMENT MODELISER ?

La manière de modéliser influence fortement

- La compréhension du problème.
- La solution.
- Il n'existe pas de modèle universel.
- Multiplicité des niveaux d'abstraction.
- Multiplicité des points de vue.
- Le modèle doit être pris du réel.

4. METAMETAMODELE ET LES NIVEAUX OMG[6]

Un métamétamodèle est un modèle qui décrit un langage de méta modélisation, c.-à-d. Le monde réel est représenté à la base de la pyramide (niveau M0). Les modèles représentant cette réalité constituent le niveau M1. Enfin, le métamétamodèle, unique et méta circulaire, est représenté au sommet de la pyramide (niveau M3).

Chaque hiérarchie définit un espace technique. Nous distinguons par exemple le modelware (espace technique des modèles), le grammarware (espace technique des grammaires définies par les langages tels que BNF2 ou EBNF3), le BDware (espace technique des bases de données), etc.

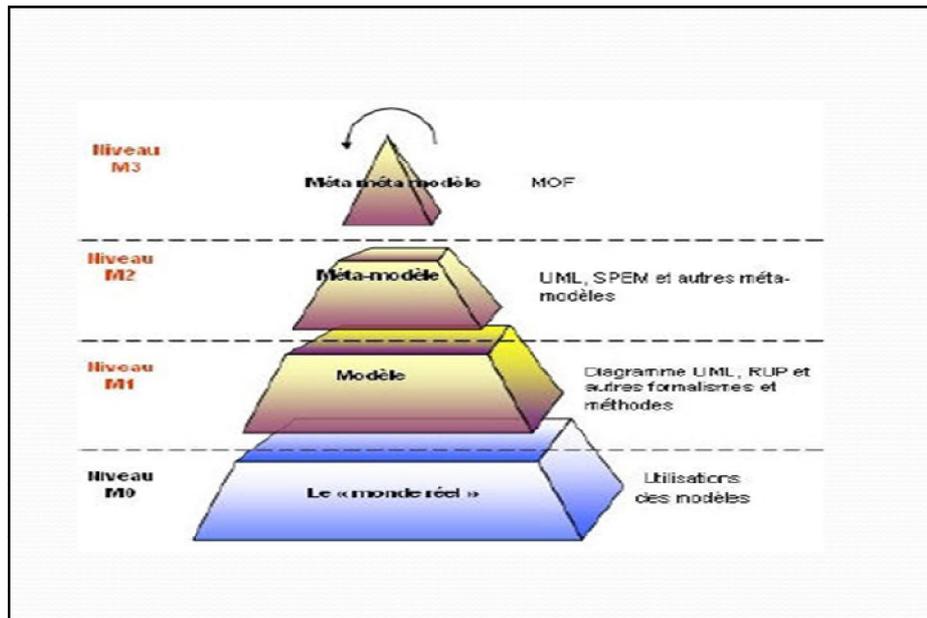


Figure 8: Pyramide de modélisation de l'OMG [7]

TRANSFORMATION DE MODELES

1. DEFINITION

Une transformation de modèles permet de passer d'un modèle *source* décrit à un certain niveau d'abstraction à un modèle *destination* décrit éventuellement à un autre niveau d'abstraction

- Ces modèles source et destination sont conformes à leur méta modèle respectif (méta modèles source et destination) et le passage de l'un à l'autre est décrit par des règles de transformation

2. LES RÈGLES DE TRANSFORMATION

- Une règle de transformation définit la manière dont un ensemble de concepts du méta modèle source est transformé en un ensemble de concepts du méta modèle destination.
- Il existe les langages impératifs qui décrivent *comment* la règle est exécutée et les langages déclaratifs qui décrivent *ce qui* est créé par la règle.
- Les règles se Basent sur trois étapes :

1-la vérification de la condition correspond à l'analyse du modèle source de la transformation de manière à détecter la présence d'un ensemble de concepts qui correspond à ce que la règle « attend » en entrée.

2-l'exécution de la règle par le moteur de transformation a lieu pour chaque ensemble de concepts qui valide la condition

3- la création consiste à générer un ensemble de concepts dans le modèle de sortie dont les champs sont remplis par les variables affectées durant l'exécution.

Approche MDA[8]

Après la technologie procédurale, la technologie objet et la technologie des composants, l'approche MDA (Model Driven Architecture) est un processus de l'ingénierie dirigée par les modèles (ou MDE pour Model Driven Engineering). Proposée par l'OMG (Object Management Group) en 2000, l'approche MDA est basée sur la séparation des préoccupations. Les modèles ne sont plus seulement un élément visuel ou de communication, mais sont, dans l'approche MDA, un élément productif et le pivot du processus MDA.

L'approche MDA distingue deux aspects principaux dans le processus de développement d'une application, l'aspect métier qui représente les fonctions de l'application, et l'aspect technique qui représente la technologie de mise en œuvre de l'application.

Le MDA définit trois niveaux de modèles représentant les niveaux d'abstraction de l'application, le CIM, le PIM et le PSM :

- **Modèle CIM- Computational Independent Model** : Les modèles d'exigence CIM décrivent les besoins fonctionnels de l'application, aussi bien les services qu'elle offre que les entités avec lesquelles elle interagit.
- **Modèle PIM- Platform Independent Model** : Les modèles PIM sont les modèles d'analyse et de conception de l'application.
- **Modèle PM- Platform Model** : Rarement utilisé, un PM décrit la structure, et les fonctions techniques relatives à une plateforme d'exécution et précise comment les utiliser.
- **Modèle PSM- Platform Specific Model**: Un PSM EST UN modèle de code qui décrit l'implémentation d'une application sur une plateforme particulière, IL est donc lié à une plateforme d'exécution.

➤ Code source : Représente le résultat final du processus MDA, le code source est obtenu par génération automatique (partielle ou totale) du code de l'application à partir du PSM. Le code source obtenu peut toujours être enrichi ou modifié manuellement.

1. LA RELATION DE IDM AVEC MDA

Les modèles CIM, PIM et PSM constituent les étapes principales de l'approche MDA. Le code est obtenu par génération automatique à partir du PSM, le PSM est obtenu par transformations successives des modèles CIM vers PIM et des modèles PIM vers PSM.

a) Transformations CIM vers PIM : Les modèles CIM expriment les besoins des utilisateurs. Cette étape consiste à construire, partiellement, des modèles PIM à partir des CIM. Le but est de retranscrire les informations contenues dans les CIM vers les modèles PIM. C'est ce qui va permettre de s'assurer que les besoins de l'utilisateur sont véhiculés et respectés tout au long du processus MDA.

b) Transformations PIM vers PIM : Les modèles PIM modélisent l'aspect structurel et dynamique d'une application. Cette étape s'exprime par l'enrichissement des modèles PIM. Enrichir des PIM consiste à leur rajouter de l'information utile et à spécifier leur contenu. Les PIM sont ainsi raffinés et les informations qu'ils contiennent, précisées.

c) Transformations PIM vers PSM : Cette étape consiste à créer des modèles PSM à partir des informations fournies par les modèles PIM, en y ajoutant des informations techniques relatives à la plateforme d'exécution cible. C'est ainsi que le lien avec la plateforme d'exécution se forme. Un PSM fournit les informations utiles à la génération du code de l'application et est dépendant de la plateforme d'exécution. Il est possible de créer autant de PSM qu'il y a de plateformes cibles.

d) Transformations PSM vers code La transformation des modèles PSM en code source consiste à générer le code source de l'application, de façon totale ou partielle, à partir des modèles PSM de l'application. Cette étape n'est pas à proprement dit considérée comme une transformation par MDA. En effet, une transformation selon MDA est définie par la transformation d'un modèle vers un autre modèle, chacun d'eux étant structuré par leur méta modèle. Or le code source n'a pas de méta modèle, la transformation des PSM vers le code est plutôt considérée comme une retranscription textuelle du modèle PSM.

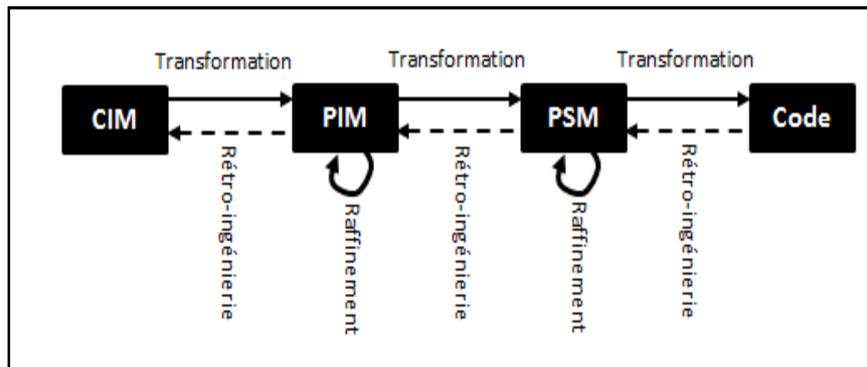


Figure 9:transformation des modelés [8]

Transformations inverses ou rétro-ingénierie, signifie la construction de modèles à partir d'applications existantes. Dans ce cadre, MDA identifie aussi les transformations inverses : Code vers PSM, PSM vers PIM et PIM vers CIM.

VUE GENERAL SUR L' IDM

L'IDM repose sur les principes suivants :

1-Capitalisation : les modèles doivent être réutilisables

Abstraction : les modèles doivent être indépendant des technologies de mise en œuvre afin d'adapter le logique métier à différents contextes et de permettre de faire plus facilement évoluer les applications vers de nouvelles

Technologies Modélisation : abordée selon une vision productive (pour générer le code final du logiciel pour une technologie de mise en œuvre donnée) proposition à la traditionnelle vision contemplative (but de documentation, spécification, communication)

Séparation des préoccupations l'IDM s'illustre généralement selon les deux principales préoccupations du métier et de la plate-forme de mise en œuvre mais d'autres préoccupations sont possibles.

CONCLUSION

Nous avons présenté, dans ce chapitre, les principes généraux de l'IDM, c'est-à-dire la méta-modélisation d'une part et la transformation de modèle d'autre part. Ces deux axes constituent les deux problématiques clé de l'IDM sur lesquelles la plupart des travaux de recherche se concentrent actuellement. Nous avons découvert aussi l'approche MDA et sa relation avec l'IDM. .

Le chapitre suivant fera l'objet d'une exposition approfondie de notre application et les démarches employées tout en démontrant l'intégration de l'IDM.

CHAPITRE III : REALISATION D'APPLICATION « APPMAKER »

INTRODUCTION

Nous avons vu dans les chapitres précédents les contextes techniques et théoriques de notre projet. Dans cette dernière partie nous allons aborder plus profondément les spécifications de notre application et nous nous intéresserons aux outils utilisés pour sa conception et sa réalisation, et révéler ses principales interfaces.

OUTILS DE TRAVAIL

1. ENVIRONNEMENT DU TRAVAIL

-L'environnement de développement Eclipse sur Linux

-Le SDK de l'Android

-ADT -IDM

-L'émulateur - Avec des spécifications :

"android-19"

Name: Android 4.4.2 Type: Platform API level: 19

-Les Images en format .png.

2. LES LIBRAIRIES DES PROGRAMMATION UTILISEE

-**java.awt**:Contient toutes les classes pour créer des interfaces utilisateur graphiques .

-**javax.swing** :Swing propose de nombreux composants dont certains possèdent des fonctions étendues, une utilisation des mécanismes de gestion d'événements performants et une apparence modifiable à la volée.

-**javax.xml** : Fournit des classes permettant le traitement des documents XML. Deux types d'analyseur enfichables sont pris en charge.

SAX(SimpleAPIforXML)

DOM (Document Object Model)

-**org.w3c.dom** : Fournit des interfaces pour le Document Object Model (DOM), qui est une composante de l'API Java API for XML Processing. Le Document Object Model niveau 2 Core API permet aux programmes d'accéder et mettre à jour dynamiquement le contenu et la structure des documents.

-**java.io** :Il fournit des fonctions du système d'entrée et de sortie et du système de fichiers java.io.InputStream, etc.

OBJECTIF

Notre application a pour objectif de simplifier à l'utilisateur la création d'une application Android d'une façon basique sans passer par le code source.

1. VUE GENERALE DE LA DEMARCHE DE L'APPLICATION AVEC LA CONCEPT IDM

Pour mieux expliquer le fonctionnement de notre application nous avons divisé les taches en deux catégories

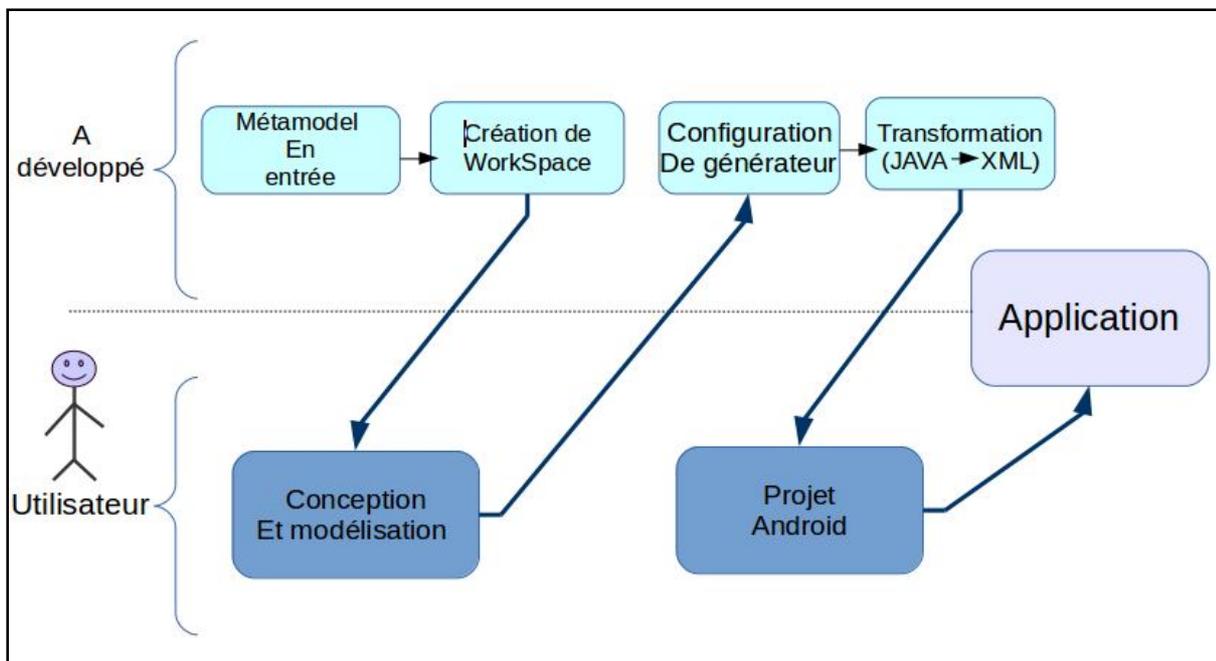


Figure 10:la vue général de l'application

- champ développeur comporte
 1. Méta-modél en entrée : la modélisation du projet du point de vue du développeur pour créer l'interface graphique de l'application <App-Maker>. (En Java le programmeur fait implicitement appel à une méta-classe pour construire une classe.)
 2. Création de Workspace : réalisation de l'interface à partir de la modélisation déjà conçu

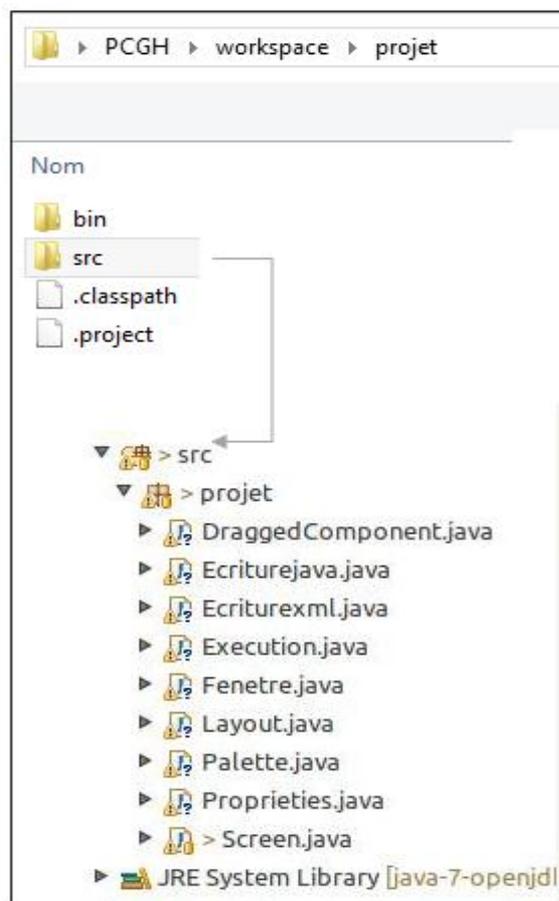


Figure 11: la classe de Java de l'interface.

3. configuration de générateur : insertion + Transformation java-> XML : traduction du code a partir de l'interface vers un autre langage XML qui génèrera l'exécution de l'application Android (en utilisant la manipulation des fichiers XML avec java pour réaliser l'interface Android.)

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/TextView"
        android:name="Screen1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/ButtonNewButton"
        android:layout_width="270dp"
        android:layout_height="wrap_content"
        android:text="NewButton" />

    <RadioButton
        android:id="@+id/RadioButtonRadio Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Radio Button" />
```

Figure 12:capture d'écran d'un fichier XML

4. projet Android : la création d'un fichier contenant tous les composants du plugin ADT

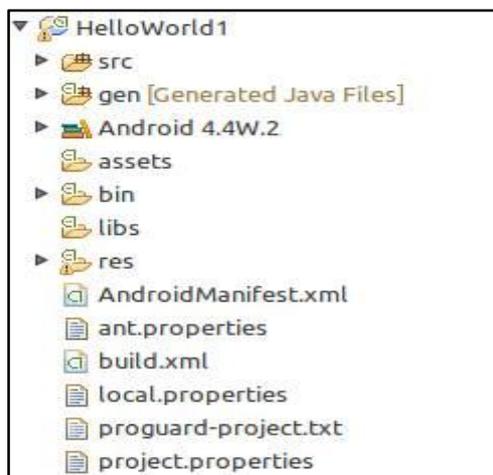


Figure 13: le projet Android générer

5. Application : qui est l'exécution du projet Android avec émulateur ou directement à partir d'un téléphone mobile.



Figure 14:capture d'écran de l'application sous mobile

- champ utilisateur comporte
 1. la modélisation de l'application Android du point de vue de l'utilisateur à partir de l'interface (workspace)
 2. exécution de la modélisation précédente pour obtenir une application Android.

ARCHITECTURE GENERALE

La structure générale est illustrée dans le schéma suivant :

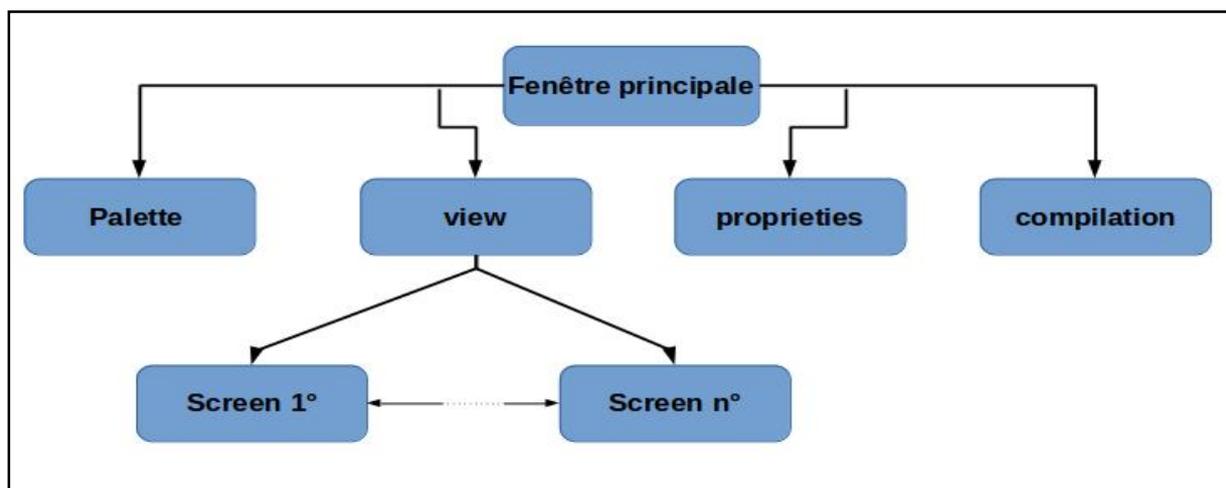


Figure 15 : la structure générale de l'application

- pour bien comprendre les besoins de l'utilisateur et les objectifs correspondant d'un système. voici les cas d'utilisation :

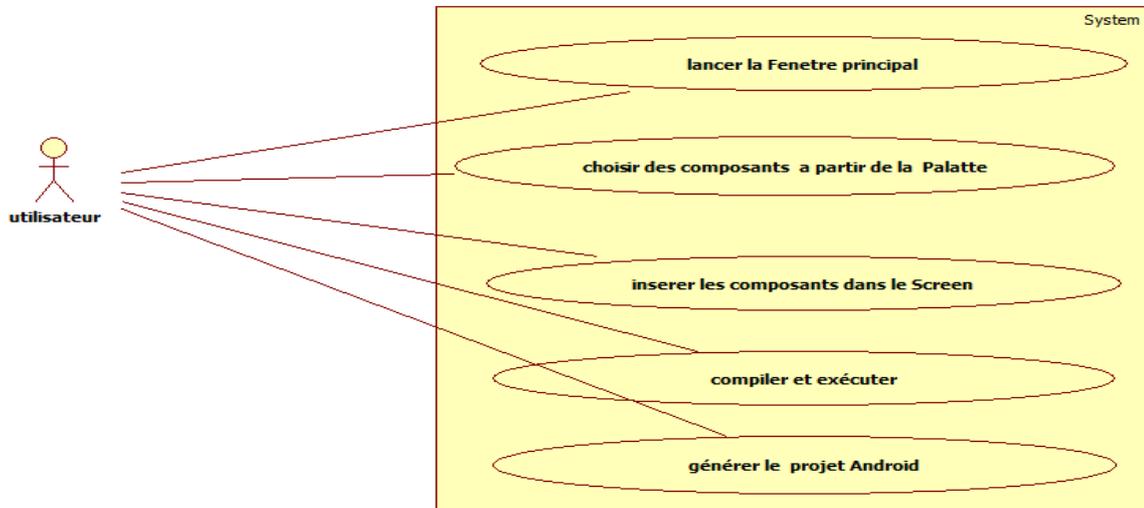


Figure 16: cas d'utilisation

- afin de voir au clair l'interaction de l'utilisateur avec la machine voici le diagramme de séquence :

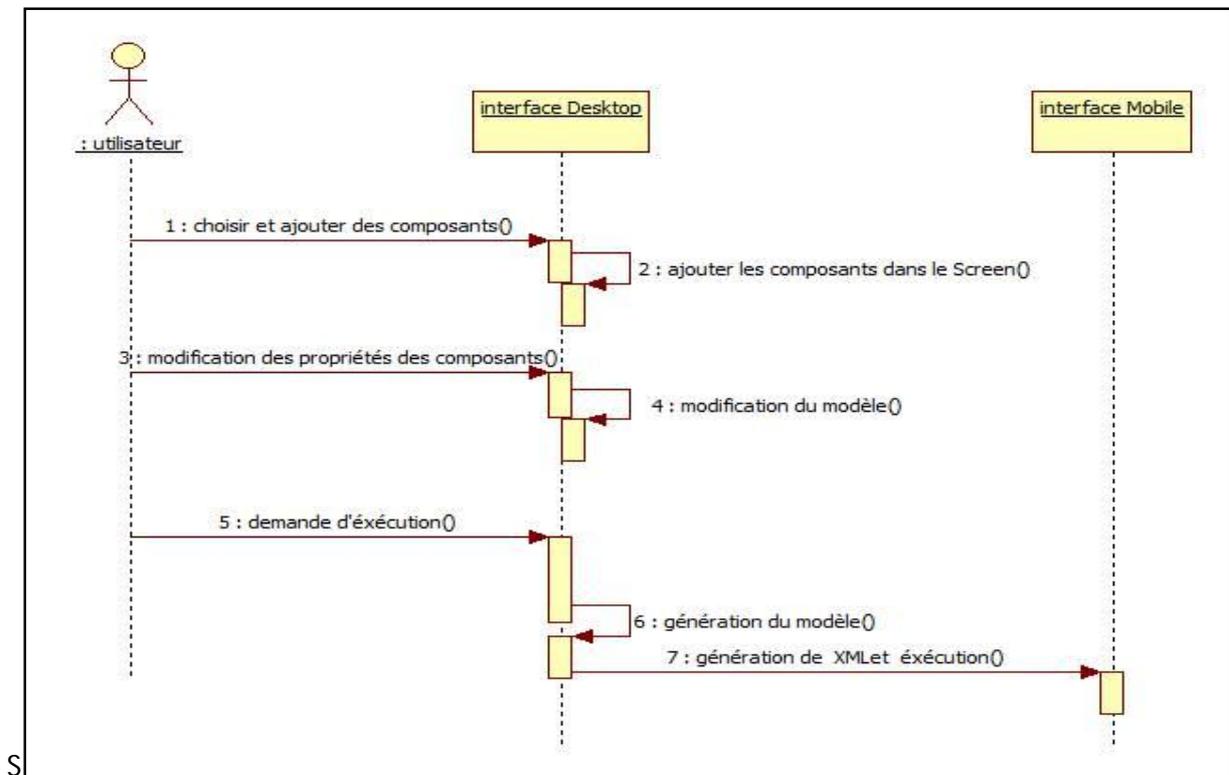


Figure 17: diagramme de séquence du déroulement d'application

INTERFACE PRINCIPALE

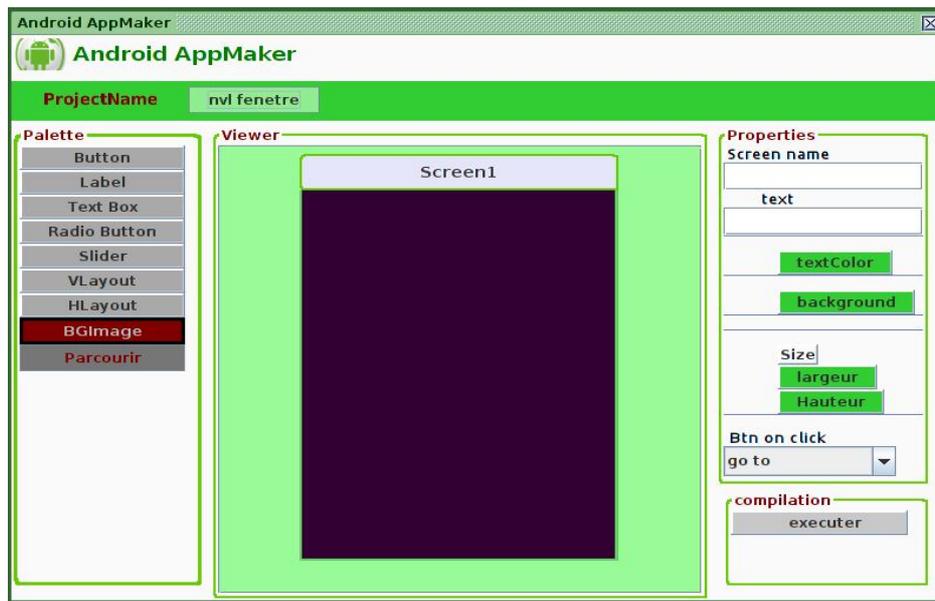


Figure 18: l'interface de l'application

Notre écran principal sert à offrir à l'utilisateur les divers outils qui font la composition dans son application. L'utilisateur peut sélectionner à partir de la palette le composant souhaité, l'ajouter et l'insérer dans l'écran virtuel, il pourra aussi modifier le nom du composant, sa couleur et sa taille, Comme il peut modifier le nom du screen et ajouter d'autres à partir du bouton «nvl fenêtre», également rajouter une photo comme arrière plan.

FONCTIONNEMENT D'APPLICATION

Explication du fonctionnement de quelque classe :

- classe principale «fenêtre»

Cette classe dont elle participe à la construction de l'interface mère (schématisé et répartie avec des horizontal et vertical box) cette dernière qui est l'espace de travail du manipulateur ainsi elle offre bien plus d'option suite à des appels fondés sur une structure arborescente des classes (Palette, Screen, DraggedComponent, proprietes....)

- classe «Palette»

Cette classe permet de créer et ajouter une liste de composants à l'interface

- classe «Screen»

Le rôle de cette classe consiste à créer une fenêtre en y introduisant des éléments de palette qui seront par la suite les composants finaux de l'application Android.

- classe «Propriétés»

Elle sert à modifier les caractéristiques des composants insérer dans le Screen (text, couleur de texte, size...)

- classe «Ecriturexml»

La classe qui génère le fruit de toute l'application, c'est un fichier de type ".xml " qui consiste à concevoir l'interface de l'application Android engendré d'une lecture des composants inséré dans le scén choisi par le créateur.

CONCLUSION GENERALE

Le mémoire se divise en 3 segments, dont le préambule comporte une introduction sur la programmation sur système Android, dans le second nous avons abordés l'ingénierie dirigée par les modèles "IDM", ce dernier a permis plusieurs améliorations significatives dans le développement des systèmes complexes en permettant de se concentrer sur une préoccupation plus abstraite que la programmation classique. Quant au dernier nous avons entamés la présentation de notre application tout en citant ses fonctionnalités et ses vertus.

La réalisation de ce projet nous a mené à explorer de nouveaux horizons de la programmation mobile tel que le fonctionnement du système Android, et d'apprendre d'avantage sur le langage de programmation " java " d'autant pour le langage XML.

BIBLIOGRAPHIE

-  [1] *livre sur TECHNO Compétences, le Comité sectoriel de main-d'œuvre en technologies de l'information et des communications, Marie-Josée Tessier, directrice générale.*
-  [2] http://en.wikipedia.org/wiki/Open_Handset_Alliance (dernière visite : 22/06/2015)
-  [3] *Créez des applications pour Android Par Frédéric Espiau (Apollidore) Licence Creative Commons 6 2.0 Dernière mise à jour le 7/10/2013*
-  [4] *MEMOIRE Présenté à Université Virtuel de Tunis De Master Professionnel en Nouvelles Technologies de Télécommunications et Réseaux*
-  [5] <http://www.mobizel.com/2015/04/definition-cest-quoi-un-sdk/>
(dernière visite : 22/06/2015)
-  [6] <https://hal.inria.fr/hal-00780187> *Subite on 23 Jan 2013*
-  [7] http://www.memoireonline.com/08/09/2643/m_Realisation-dun-systeme-multi-agents-dassistance-a-la-modelisation1.html
-  [8] <http://laine.developpez.com/tutoriels/alm/mda/generalites-approche-mda/>
(dernière visite : 22/06/2015)