



Développement de Widgets

Livre blanc

Juillet 2009

Auteurs

Ce document a été réalisé par des consultants de C2S :

- Lionel Laské, Directeur Innovation,
- Smaïl Aïssaoui, Ingénieur Consultant,
- Hervé Ayeau, Ingénieur Consultant.

Pour tous renseignements complémentaires, vous pouvez nous contacter sur <http://www.c2s.fr>

Sommaire

Auteurs	2
Sommaire	3
Objet du document	5
Définition	5
Exemples de Widgets	6
Plateformes étudiées	6
Méthodologie de l'étude.....	6
Windows Vista.....	7
Présentation générale	7
Architecture logicielle des Widgets Vista	7
Développer un Widget pour Vista.....	8
Synthèse sur la plate-forme	11
iPhone.....	12
Présentation générale	12
Architecture logicielle de l'iPhone.....	12
Développer une application pour iPhone	13
Synthèse sur la plate-forme	17
Android.....	18
Présentation générale	18
Description de l'architecture logicielle d'Android.....	18
Développer une application pour Android.....	19
Synthèse sur la plate-forme	22
Yahoo ! Widget.....	23
Présentation générale	23
Architecture logicielle d'un Widget Yahoo.....	23
Développer un Widget Yahoo	25
Synthèse sur la plate-forme	30
Yahoo ! Mobile	31
Présentation Générale	31
Architecture logicielle des Widgets Yahoo Mobile	31
Développer un Widget Yahoo Mobile.....	33

Synthèse sur la plate-forme	36
Windows Mobile	37
Présentation générale	37
Architecture logicielle de l'écran Aujourd'hui.....	37
Développer une application pour l'écran Aujourd'hui.....	38
Synthèse sur la plate-forme	41
Samsung Touch Wiz.....	42
Présentation générale	42
Architecture logicielle des Widgets Samsung	42
Développer un Widget Samsung.....	43
Synthèse sur la plate-forme	45
webOS	46
Présentation générale	46
Architecture logicielle d'une application webOS	46
Développer une application webOS.....	47
Synthèse sur la plate-forme	51
Conclusion	52
Références.....	53
A propos de C2S	54

Objet du document

Les Widgets sont partout. Lancés à l'initiative des sites webs comme NetVibes, Windows Live ou iGoogle, leur objectif est de permettre à un utilisateur de personnaliser sa page d'accueil et d'y faire apparaître les choses vraiment importantes pour lui. Chacun peut ainsi « piocher » dans une galerie de Widgets, ou développer ses propres Widgets, pour construire son environnement idéal.

Avec Konfabulator (rapidement racheté par Yahoo) puis avec l'arrivée de Windows Vista, les Widgets ont progressivement migrés du navigateur vers le bureau du PC en tant qu'élément de personnalisation au même titre que le fond d'écran.

Les Widgets sont maintenant présents sur nos terminaux mobiles où ils s'apprêtent à jouer le même rôle que sur un PC, étant mis en avant comme l'élément différenciant des terminaux.

Impossible donc de passer à côté de cette technologie. L'objet de ce livre blanc est d'explorer ce phénomène sur le plan technique c'est-à-dire, précisément, de comparer la manière de développer des Widgets sur différentes plateformes.

Notre objectif est d'apporter un éclairage sur les similitudes et les différences du développement de Widget afin de mieux en comprendre les contraintes.

Définition

Commençons par une définition : qu'est ce qu'un « Widget » ? On nomme « Widget » une application installée sur le bureau d'un terminal qui affiche de manière synthétique une information locale ou distante qui se met à jour régulièrement. Le mot « terminal » s'entend comme un PC ou un terminal mobile. Par contre, nous excluons de notre étude les « Widgets Web » de type Windows Live, NetVibes ou iGoogle.

Un Widget peut proposer une interface permettant d'interagir avec lui, par exemple pour afficher une vue détaillée des informations. Un Widget dispose d'une interface de configuration qui permet de paramétrer le type ou la forme des informations qu'il affiche. Par extension sur certaines plateformes on utilisera le terme « Widget » pour désigner une application lorsque c'est le seul élément de personnalisation du terminal.

Le mot « Gadget » est également utilisé, notamment par Microsoft, pour parler de « Widget ». Nous lui préférons le mot « Widget » qui est plus communément rencontré et qui est moins péjoratif...

Exemples de Widgets

Voici quelques exemples de Widgets répondant à la définition ci-dessus et qu'on trouve sur les différentes plates-formes:

- Cours de la bourse,
- Horloge,
- Performance CPU,
- Flux RSS,
- Météo,
- Calendrier,
- Sytadin (Trafic routier en Ile-de-France),
- Contacts,
- Annuaire,
- Diaporama photo,
- Niveau de batterie.

Plateformes étudiées

Pour cette étude, nous avons choisi d'étudier les plates-formes suivantes :

- Windows Vista,
- iPhone,
- Android,
- Yahoo Widget,
- Yahoo Mobile,
- Windows Mobile,
- Samsung Touch Wiz,
- Palm Web OS.

Méthodologie de l'étude

L'étude de chaque plateforme s'effectuera sur un mode similaire. L'objectif est chaque fois de répondre aux questions suivantes :

- Quels sont les outils nécessaires pour développer un widget sur cette plate-forme ?
- Quel est l'architecture d'un widget sur cette plate-forme ?
 - Quelle technologie pour développer l'interface ?
 - Quelle technologie pour développer les traitements ?
- Comment peut-on déployer un nouveau widget sur cette plate-forme ?
- Comment implémenter les fonctionnalités élémentaires d'un widget :
 - Boîte de configuration,
 - Accès à une URL distante en HTTP,
 - Rafraîchissement des données à intervalle régulier,
 - Accès à une information locale de la machine (date, fichier, ...),
 - Interaction clavier ou souris avec le widget.
- Comment développer un widget type: Sytadin, Bouygues Epargne Salariale, Annuaire C2S, ... ?

Windows Vista

Présentation générale

Windows Vista est le système d'exploitation client proposé par Microsoft pour se substituer à Windows XP. Une des nouveautés de Windows Vista est la possibilité d'afficher des Widgets, appelés « Gadgets » ici, dans une barre spécifique située sur le côté de l'écran, la « Sidebar ».

On retrouve également les Widgets dans Windows 7, mais ils peuvent être positionnés n'importe où sur le bureau au lieu d'être limité à la « Sidebar ».

Pour générer un écosystème autour du développement de Widgets, Microsoft propose l'hébergement gratuit de Widgets sur son site Windows Live. Le site héberge à ce jour plusieurs centaines de Widgets en libre téléchargement.

Architecture logicielle des Widgets Vista

Pré-requis

La SideBar Vista est installée par défaut dans Vista. Un ensemble de Widgets de base (cours de la bourse, météo, diaporama, ...) est également proposé en standard. Aucun outil n'est donc nécessaire pour faire fonctionner des Widgets.

Description générale

Le schéma ci-dessous détaille l'architecture de la SideBar sous Vista et Windows 7.



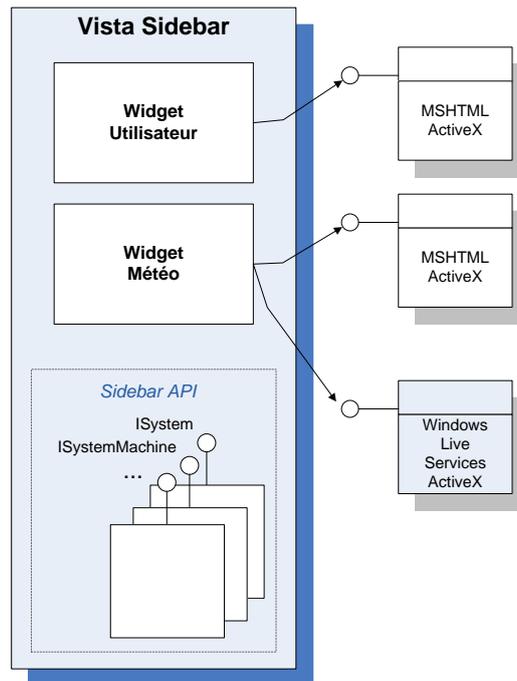


Figure 1: Architecture de la SideBar Vista

La SideBar est une application Windows qui héberge, via le composant ActiveX MSHTML, une instance d'Internet Explorer pour chaque Widget installé. Les Widgets standards météo et cours de la bourse repose de plus sur un composant ActiveX qui communique avec la plate-forme Windows Live.

Architecture d'un Widget Vista

Un Widget pour Vista est constitué :

- D'un fichier XML de configuration décrivant les propriétés du Widget,
- Un ensemble de pages HTML correspondants à l'interface du Widget son affichage et sa configuration,
- Un ensemble de fichiers JavaScript correspondants aux traitements réalisés par le Widget,
- Un ensemble d'images utilisées par les pages HTML.

Développer un Widget pour Vista

Outils de développement

Développer un Widget pour Vista est équivalent à développer une page HTML. Tout outil permettant le développement de pages HTML et de Scripts peut donc être utilisé pour développer un Widget : Visual Studio, DreamWeaver, FrontPage,

Déploiement

Un Widget est un fichier compressé contenant un répertoire avec toute l'arborescence des fichiers HTML, JavaScript, Images et XML qu'il contient. L'extension de fichier .ZIP est néanmoins à remplacer par l'extension .Gadget.

Le déploiement s'effectue par un double-clic sur le fichier « .Gadget ». Il est également possible de faire héberger le Widget sur la plate-forme Windows Live de Microsoft, le Widget est alors téléchargé et installé automatiquement après son téléchargement.

Implémentation des principales fonctionnalités

Boîte de configuration

La boîte de configuration est simplement une page HTML représentant son paramétrage. L'utilisateur accède à la boîte de configuration en cliquant sur l'icône « clé » représenté sur à droite du cadre du Widget.



Figure 2: Accès à la boîte de configuration d'un Widget

La SideBar fournit également une API JavaScript permettant de stocker et de récupérer les informations de configuration d'un Widget entre deux redémarrages de la machine.

Accès à une URL distante en http

Un Widget étant une page HTML, il est possible d'appeler une URL distante en HTTP via la technologie AJAX.

Exemple :

```
httpreq = new XMLHttpRequest("Microsoft.XMLHTTP");  
httpreq.open("GET", urltocal, true);  
httpreq.onreadystatechange = processRequest;  
httpreq.send(null);
```

On peut également référencer un image distante depuis le code HTML du Widget.

Rafraichissement des données à intervalle régulier

L'utilisation des timers JavaScript permet de déclencher dans un Widget un traitement à intervalle régulier.

Accès à une information locale de la machine (date, fichier, ...)

La SideBar intègre une API utilisable depuis les Widgets qu'elle héberge, qui propose l'accès à de nombreuses informations locales : informations systèmes, fichiers, ... Chaque fonction est utilisable via un appel du type « System.API.Nomdefonction ».

Exemple :

```
System.Debug.outputString("%CPU is " +System.Machine.CPU.usagePercentage);
```

Un Widget peut également intégrer l'appel à un composant ActiveX pour effectuer des opérations qui ne seraient pas présentes dans l'API.

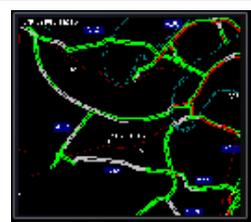
Interaction clavier ou souris avec le Widget

Tous les champs de saisie HTML sont utilisables dans un Widget : Texte, Bouton, Case à cocher, Liste,... De la même manière, l'ensemble des événements JavaScript est utilisable normalement (déplacement souris, clic, ...).

Exemple de fonctionnalités: Sytadin, Bouygues Epargne Salariale, Annuaire C2S

Plusieurs Widgets Vista ont été réalisés par C2S.

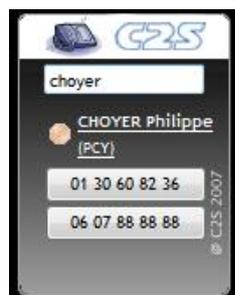
Sytadin



Représentation de l'état du trafic routier en Ile-de-France.
Widget classé dans les dix Widgets les plus téléchargés sur le site de Microsoft avec plus de 200 000 téléchargements à ce jour.

Téléchargeable sur <http://gallery.live.com/livItemDetail.aspx?li=a74dc3ee-cd09-4799-b996-24433f79d5db>

Annuaire C2S



Accès à des recherches dans l'annuaire C2S. Affichage des informations de présence MOCS, clic-to-call sur les plates-formes Alcatel et MITEL. Accès à l'annuaire via l'interface du web service Annuaire Groupe.

Bouygues Epargne Salariale



Affichage des gains sur les produits d'épargne Bouygues: Bouygues Confiance 3, Bouygues Confiance 4 et Bouygues Partage. Configuration des sommes engagées et de l'affichage, récupération en temps réel du cours de l'action via Windows Live.

Synthèse sur la plate-forme

La grande force de la plate-forme de Widget Vista est son intégration au système qui permet une utilisation sans aucune installation préalable. L'accès intégré à Windows Live pour le téléchargement de nouveaux Widgets permet également à un utilisateur d'ajouter facilement de nouveaux Widgets.

Le développement et le packaging de Widget est également grandement simplifié par le fait qu'il est complètement identique au développement d'une page HTML avec les contraintes liées à la taille limitée de la zone.

L'inconvénient majeur des Widgets Vista est qu'ils ne peuvent fonctionner que sous Vista et Windows 7.

iPhone

Présentation générale

L'iPhone est un smartphone lancé par Apple en 2007. L'appareil combine les usages d'un téléphone mobile, d'un terminal internet mobile, d'un assistant personnel et d'un baladeur numérique. Parmi les fonctionnalités qui démarquent l'iPhone des produits concurrents figurent une interface constituée d'un écran tactile multipoint, remplaçant les boutons ou claviers traditionnels, des détecteurs de lumière et de proximité pour optimiser la batterie de l'appareil et verrouiller l'utilisation du clavier tactile lorsqu'il est porté à l'oreille, un écran large dont le visionnage en mode paysage est automatiquement détecté via un dispositif gyroscopique lorsque l'utilisateur fait pivoter l'appareil et la représentation graphique de la boîte de messagerie vocale.



Architecture logicielle de l'iPhone

Description générale

Le système d'exploitation de l'iPhone est similaire à Mac OS X, en particulier, le kernel de l'iPhone est basé sur celui de Mac OS X.

Le kernel (bas niveau) n'est jamais utilisé directement pour le développement d'application. Le développement d'applications fait appel à des couches de services de plus haut niveau qui sont constituées de :

- Cocoa Touch : UIKit and Foundation frameworks (langage Objective C),
- Media: Graphics Technologies, Core Audio, OpenAL, Video Technologies,
- Core Services: Data, Network Security,

- Core OS: Threading (POSIX threads), Networking (BSD sockets), File-system access, Standard I/O, Bonjour and DNS services, Locale information, Memory allocation, Math computations.

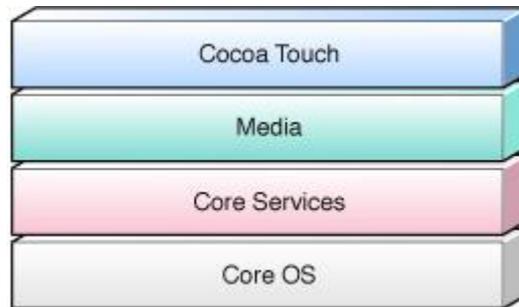


Figure 3 - iPhone OS technology layers

Développer une application pour iPhone

Pour iPhone, on ne parle pas de « Widgets » car il n'est pas possible de déposer un élément sur le bureau, à l'exception de l'icône de lancement d'une application. De plus, plusieurs applications ne peuvent s'exécuter en même temps sur l'iPhone (voir plus loin). Nous étudions donc ici le développement d'applications iPhone.

Pré-requis logiciels et matériels

L'environnement de développement d'applications pour iPhone est constitué de :

- Un macintosh,
- L'« iPhone SDK » téléchargeable sur le site d'Apple qui contient en outre :
 - « XCode », l'outil de développement propriétaire du Mac,
 - Les bibliothèques et plugins pour le développement d'applications iPhone,
 - Un émulateur de l'iPhone
- Une licence développeur afin de publier ses applications sur l'AppStore.

Cet environnement permet de générer du code binaire pour l'iphone. Il est également possible de réaliser des applications web qui « miment » le comportement des applications natives : utilisations des contrôles du système, possibilité d'un lancement depuis le bureau, ... Pour cela, il existe un plugin pour l'environnement Open Source Eclipse.

Outils de développement

Apple fournit bien plus qu'un atelier de génie logiciel pour développer des applications sur sa plateforme. En effet, XCode se veut être une suite d'outils de développement permettant notamment de :

- Editer le code,

- Générer les binaires en debug et en release,
- Déploiement sur le simulateur ou sur l'iPhone,
- Gérer les sources,
- Instrumentation, Performance tuning, ...

L'interface graphique se construit à partir de l'outil « Interface Builder ».

Cet utilitaire permet à partir de glisser/déposer de composants graphiques existants ou étendus de créer des fenêtres générées sous forme de ressources à inclure à un projet XCode.

Architecture d'une application iPhone, le modèle MVC de Mac

Une application iPhone se construit sur la base du modèle de programmation MVC (Modèle Vue Contrôleur) de Cocoa.

La spécificité de ce modèle MVC est que le contrôleur a un rôle de médiateur entre la vue et le modèle. Il contrôle les échanges dans les deux sens.

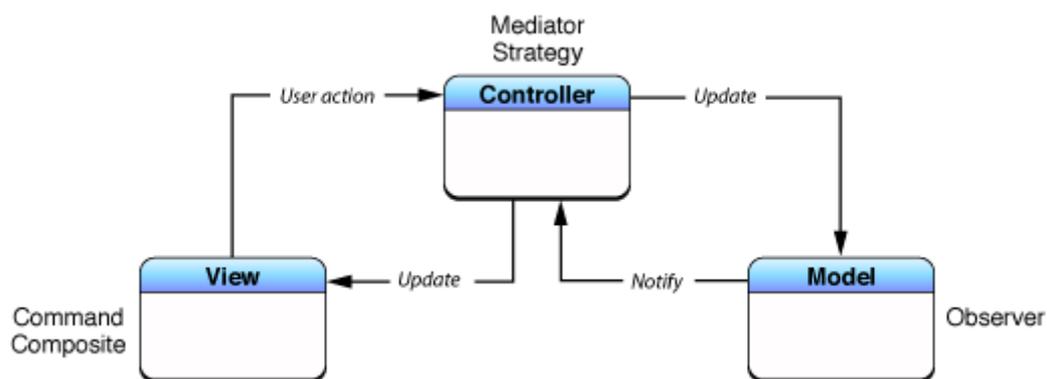


Figure 4- Cocoa version of MVC as compound design pattern

Déploiement

Le déploiement d'applications iPhone passe par l'« App Store », une chaîne spécifique disponible pour iPhone et iPod Touch. Pour les applications commerciales, Apple reverse 70 % des revenus aux éditeurs. Pour les entreprises, Apple propose un mode de diffusion interne des applications qui nécessitent le déploiement via iTunes auquel est attaché un certificat spécifique pour l'entreprise.

Par l'intermédiaire de certains outils de hacking (Jailbreak), il est également possible de forcer le déploiement d'une nouvelle application sans passer par l'« App Store ».

Une package iPhone est une arborescence de fichiers (binaires, ressources) et un manifest décrivant l'application. Le packaging s'effectuera de préférence à l'aide de XCode qui génère automatiquement le manifest.

Implémentation des principales fonctionnalités

Boite de configuration

Une boite de configuration est simplement une fenêtre « UIView » développée en langage Objective C.

(cf. sample [UICatalog](#))

Accès à une URL distante en http

Le Framework **CFNetwork** intègre les fonctionnalités permettant d'effectuer des tâches à travers un réseau :

- Requête http / Web service
- Transfert de fichier FTP
- BSD Sockets,
- ...

CFNetWork est une API en C.

(cf. sample [WiTap](#))

Rafraîchissement des données à intervalle régulier

Il est possible de rafraîchir les données en développant des Timer ou des « POSIX Thread ».

Plus intéressant, Cocoa intègre un mécanisme avancé de databinding, qui permet de notifier et de mettre à jour la vue via le contrôleur de tout changement de donnée (cf. Figure 2- Cocoa version of MVC as compound design pattern).

Accès à une information locale de la machine (date, fichier, ...)

Une API de gestion des fichiers (équivalent du namespace System.IO de .NET) est disponible.

Cette API permet d'ouvrir des flux de données à travers le réseau ou sur l'appareil en local.

Plus bas niveau, il est possible d'ouvrir un fichier à travers les fonctions du C (fopen, FILE *, ...).

Interaction clavier ou souris avec le Widget

- Le clavier est intégré de façon native aux champs de saisies, c'est-à-dire que lorsqu'un champ de saisie obtient le focus, le clavier virtuel de l'iPhone apparaît, tout comme avec Windows Mobile.
- Il est possible de récupérer les événements « appuie de touches » par l'implémentation de délégués.

Les applications pour iPhone sont des applications événementielles. Toutes les fenêtres gèrent donc une boucle des messages. Il suffit alors de récupérer la notification voulue pour traiter l'événement.

Exemple de fonctionnalités: Sytadin, Bouygues Epargne Salariale, Annuaire

Le Widget Sytadin existe déjà et est sans grand intérêt pour comprendre le fonctionnement d'une application iPhone.

Une application type annuaire serait plus intéressante, mais nécessite plusieurs jours de développements pour un néophyte (compter environ 5 jours).

A noter que beaucoup d'applications pour iPhone sont des coquilles vides dans le sens où ce sont des applications avec :

- Une unique fenêtre contenant un seul contrôle : le browser web safari.
- Un fichier de configuration contenant l'url du site contenant le fonctionnel.

Exécution en background, services

Chose étonnante, il est impossible de développer un service, ou de lancer une exécution en tâche de fond. Deux applications ne peuvent tourner en même temps sur l'iPhone, ce qui signifie que lorsqu'on passe d'une application à une autre, la première est interrompue pour laisser la main à la seconde.

Particularités ergonomiques de la plateforme

Sans conteste, le **Multi-Touch** est « LA » particularité ergonomique de l'iPhone.

Le multi-touch permet d'être notifier de la pression sur l'écran à deux endroits simultanément (ce qui permet d'effectuer le zoom). Par programmation, le multi-touche s'implémente par 5 fonctions :

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;  
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;  
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;  
- (void)touchesChangedWithEvent:(UIEvent *)event;  
- (void)touchesCancelled;
```

Chaque UIEvent est constitué d'une collection de UITouch. Un UITouch permet d'identifier, de localiser une action sur l'écran.

Le Framework ne permet donc pas de détecter directement un événement zoom, ... Cette action doit être interprétée en consolidant les événements basiques déclenchés par le système.

Synthèse sur la plate-forme

Le développement sur iPhone s'appuie sur une plate-forme de développement éprouvée : l'ensemble des outils utilisés pour le développement d'applications Mac. Cela permet une réelle productivité des développements passé la période d'apprentissage de la plate-forme.

L'inconvénient majeur des développements iPhone est un manque d'ouverture : d'abord sur le déploiement d'applications propriétaires et ensuite sur les APIs (l'API Bluetooth par exemple est limitée à la communication avec des périphériques Apple).

Android

Présentation générale

Android est un système d'exploitation open source pour smartphones, PDA et terminaux mobiles conçu Google, et annoncé officiellement en novembre 2007. Afin de promouvoir ce système d'exploitation ouvert, Google a su fédérer autour de lui une trentaine de partenaires réunis au sein de l'Open Handset Alliance. Même si peu de terminaux existent aujourd'hui, de nombreux constructeurs s'intéressent fortement à ce nouveau système et on annoncés des terminaux.

Notons enfin l'engouement chez les développeurs qui ont pu s'essayer à cette nouvelle plateforme par l'Android Developer Challenge organisé par Google en 2008 et 2009.



Description de l'architecture logicielle d'Android

Description générale

Android est aujourd'hui constitué d'un système d'exploitation, et de différentes couches de services.

Le noyau du système est basé sur un kernel linux version 2.6 qui n'est jamais utilisé directement pour le développement d'applications.

Le développement d'applications fait appel à des couches de services de plus haut niveau qui sont elles mêmes utilisées par les applications délivrées avec Android (Contacts, telephone, navigateurs, ...).

On découpe l'architecture d'Android en 5 niveaux :

- Les applications d'Android,
- Application Framework qui est une API java, utilisée pour développer les interfaces graphiques, ...
- La couche Librairies qui propose des fonctionnalités plus bas niveau en C/C++ (à ce jour, il n'est pas possible de s'interfacer avec ces libraires),
- Le runtime,
- Le noyau.

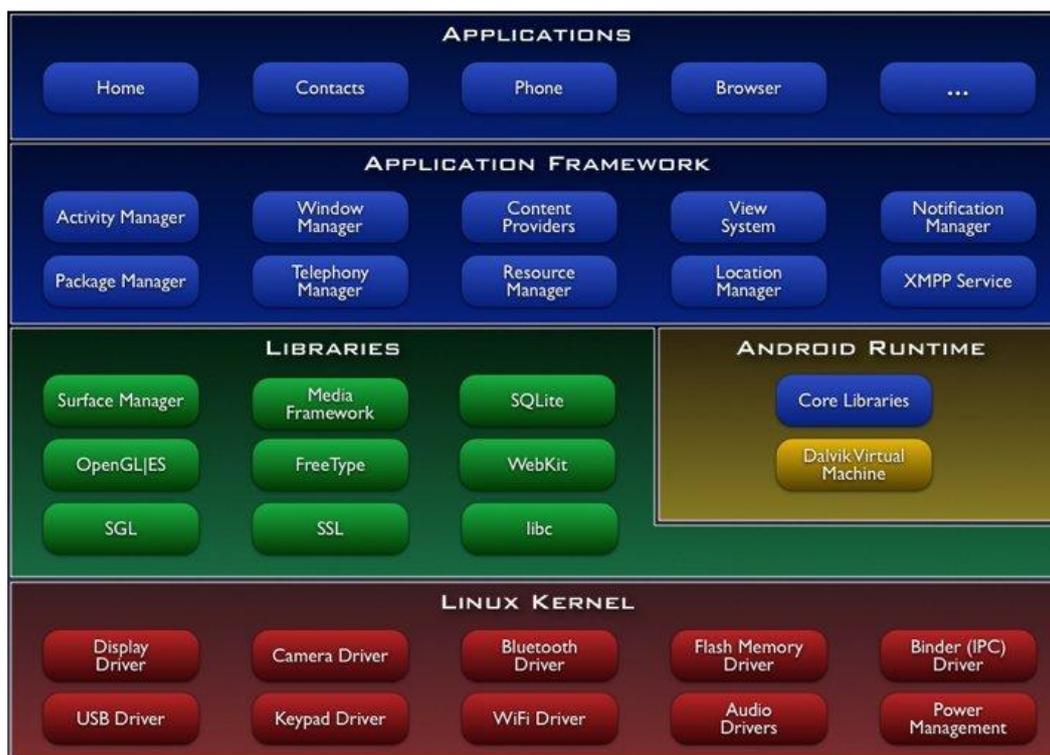


Figure 5 – Architecture logicielle d'Android

Développer une application pour Android

Les Widgets n'étaient pas supportées initialement par Android qui permettait simplement de déposer sur le bureau des icônes de lancement des applications. Néanmoins, depuis Android 1.5, la notion de Widgets existe réellement, il s'agit d'applications qui respectent des [conventions graphiques](#) particulières pour s'intégrer sur le bureau.

Pré-requis logiciels et matériels

L'environnement de développement d'application pour Android se constitue de :

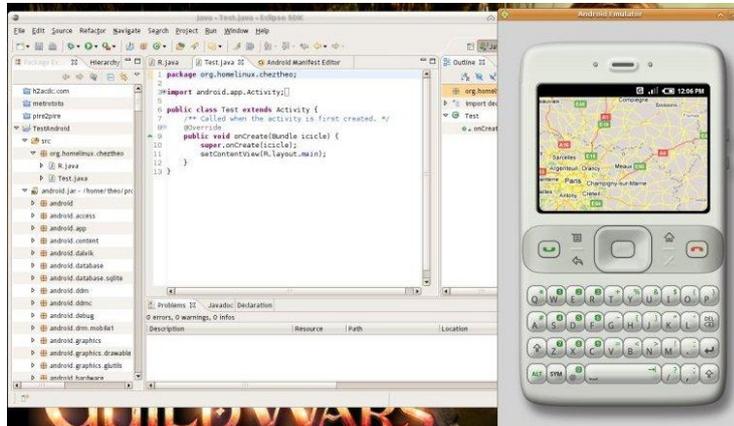
- Un ordinateur sous Windows XP ou Vista, Linux (i386) ou MaC OS X Intel,
- Eclipse 3.2, 3.3 complété des plugins JDT et WST,
- D'un SDK téléchargeable sur le site Google qui contient en outre :
 - Un plugin pour Eclipse,
 - Un émulateur Android.

Outils de développement

Les applications pour Android sont des applications obligatoirement écrites en Java.

Le SDK d'android fournit un ensemble d'utilitaires qui permettent de compiler, déployer et émuler les applications pour Android indépendamment de tout IDE.

Toutefois, le SDK fournit également un plugin qui s'intègre à Eclipse. On privilégiera cette dernière solution qui a l'avantage de faciliter la création de l'application, son débogage ainsi que son déploiement.



En outre, la plupart des plugins Eclipse pour Java peuvent être utilisés pour le développement d'application Android (tests unitaires, gestion de code source, ...).

Architecture d'une application Android

Les applications Android se construisent à partir de 4 types de composants selon le fonctionnel à implémenter :

- **Activity**
Le bloc le plus utilisé, il permet d'exposer une vue et d'être notifié des interactions utilisateurs. Pour développer une activité il suffit de surcharger la classe de base « **Activity** »
Le graphe de navigation entre écrans est géré par les classes appelées « **Intent** ».
- **Intent Receiver**
Permet de « se mettre à l'écoute » d'un événement déclenché par une autre application (réception d'un appel téléphonique, ...).
- **Service**
Permet d'implémenter une application type « **Service** », dont la durée de vie est illimitée et ne nécessite pas d'interface homme machine.
- **Content Provider**
Permet de partager des données entre plusieurs applications.

Déploiement

Eclipse permet de compiler une application Android en fichier « **.APK** ».

Un **.APK** est une archive compressé contenant le code java généré en byte-code ainsi que les ressources nécessaires.

Une fois constituée, l'archive .APK doit être déposée dans l'émulateur puis exécutée.

A ce jour, il n'existe pas de mode de déploiement plus industriel pour cette plateforme, qui rappelle le n'est qu'en phase de test.

Implémentation des principales fonctionnalités

Boite de configuration

Une boite de configuration est simplement une fenêtre « **View** » équivalent d'une **Frame** en Java.

A noter qu'il existe deux manières de construire une fenêtre :

- De façon programmatique : ajouts de contrôles dans une fonction d'initialisation de la fenêtre (analogue à WinForm.NET)
- En définissant dans un fichier XML la structure de l'écran (analogue à XAML en WPF .NET)

Accès à une URL distante en http

Le package **javax.net** intègre les fonctionnalités permettant d'effectuer des tâches à travers un réseau :

- Requête HTTP / Web service,
- Transfert de fichiers en FTP,
- Socket, ...

Pour appeler une URL distante, on utilisera plus particulièrement la classe `URLConnection` de ce package.

Rafraîchissement des données à intervalle régulier

Il est possible de rafraîchir les données en développant des **Timer** (classe `java.util.Timer`) ou des **Thread** (classe `java.lang.Thread`). Il est aussi possible d'implémenter des applications de type « Service ».

Accès à une information locale de la machine (date, fichier, ...)

Les APIs standards (`java.io` pour les streams, `java.util` pour les dates, ...) du JDK de Sun sont utilisables par une application Android.

Interaction clavier ou souris avec le Widget

- Le clavier est intégré de façon native aux champs de saisies, c'est-à-dire que lorsqu'un champ de saisie obtient le focus, le clavier virtuel d'Android apparaît, tout comme avec Windows Mobile.

- Il est possible de récupérer les événements « clicks » par l'implémentation de « Listener » (similaire au JDK de sun).

Exemple :

```
public class SendResult extends Activity
{
    /**
     * Initialization of the Screen after it is first created. Must at least
     * call getContentView\(\) to
     * describe what is to be displayed in the screen.
     */
    protected void onCreate(Bundle savedInstanceState)
    {
        ...

        // Listen for button clicks.
        Button button = (Button)findViewById(R.id.corky);
        button.setOnClickListener(mCorkyListener);
    }

    // Create an anonymous class to act as a button click listener.
    private OnClickListener mCorkyListener = new OnClickListener()
    {
        public void onClick(View v)
        {
            // To send a result, simply call setResult() before your
            // activity is finished.
            setResult(RESULT_OK, "Corky!");
            finish();
        }
    };
};
```

abonnement au listener

implémentation du listener

Exemple d'applications: Sytadin, Bouygues Epargne Salariale, Annuaire

Le développement d'un Widget comme Sytadin est sans grand intérêt pour comprendre le fonctionnement d'une application Android, en effet, il suffit d'utiliser le browser web d'Android et d'implémenter un timer pour interroger une URL distante.

Une application type annuaire serait plus intéressante mais n'a pas pu être réalisée pendant l'étude.

Exécution en background, services

Android offre la possibilité d'exécuter des applications en tâche de fond (composant Service). Plusieurs applications peuvent s'exécuter en parallèle et communiquer entre elles à travers des APIs de haut niveau.

Synthèse sur la plate-forme

La plate-forme Android est une plate-forme réellement prometteuse pour le développeur. Le modèle de développement proposé (reposant sur Java et sur l'Open Source), les paradigmes de programmation et l'intégration native des différentes API Google permettent d'envisager la réalisation d'applications à réelle valeur ajoutée.

Yahoo ! Widget



Présentation générale

Yahoo! Widget, anciennement appelé Konfabulator, est un logiciel qui permet d'afficher sur le bureau d'un ordinateur de type Windows XP, Windows Vista ou Mac, des Widgets (ou Widgets). Très populaires lorsqu'ils ont été lancés par Konfabulator, il existe désormais plusieurs centaines de Widgets en téléchargement sur Internet.

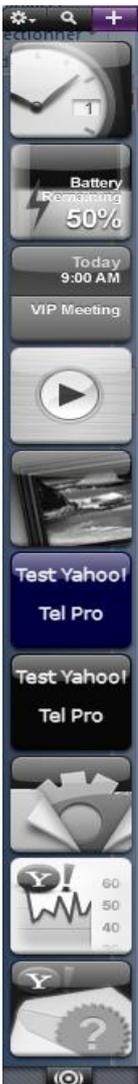
Architecture logicielle d'un Widget Yahoo

Pré-requis logiciels

- L'utilisation des « Yahoo ! Widget » nécessite l'installation préalable d'un moteur d'exécution qui installe la barre Yahoo sur le bureau.
- Il existe un kit de développement des Widget Yahoo qui contient :

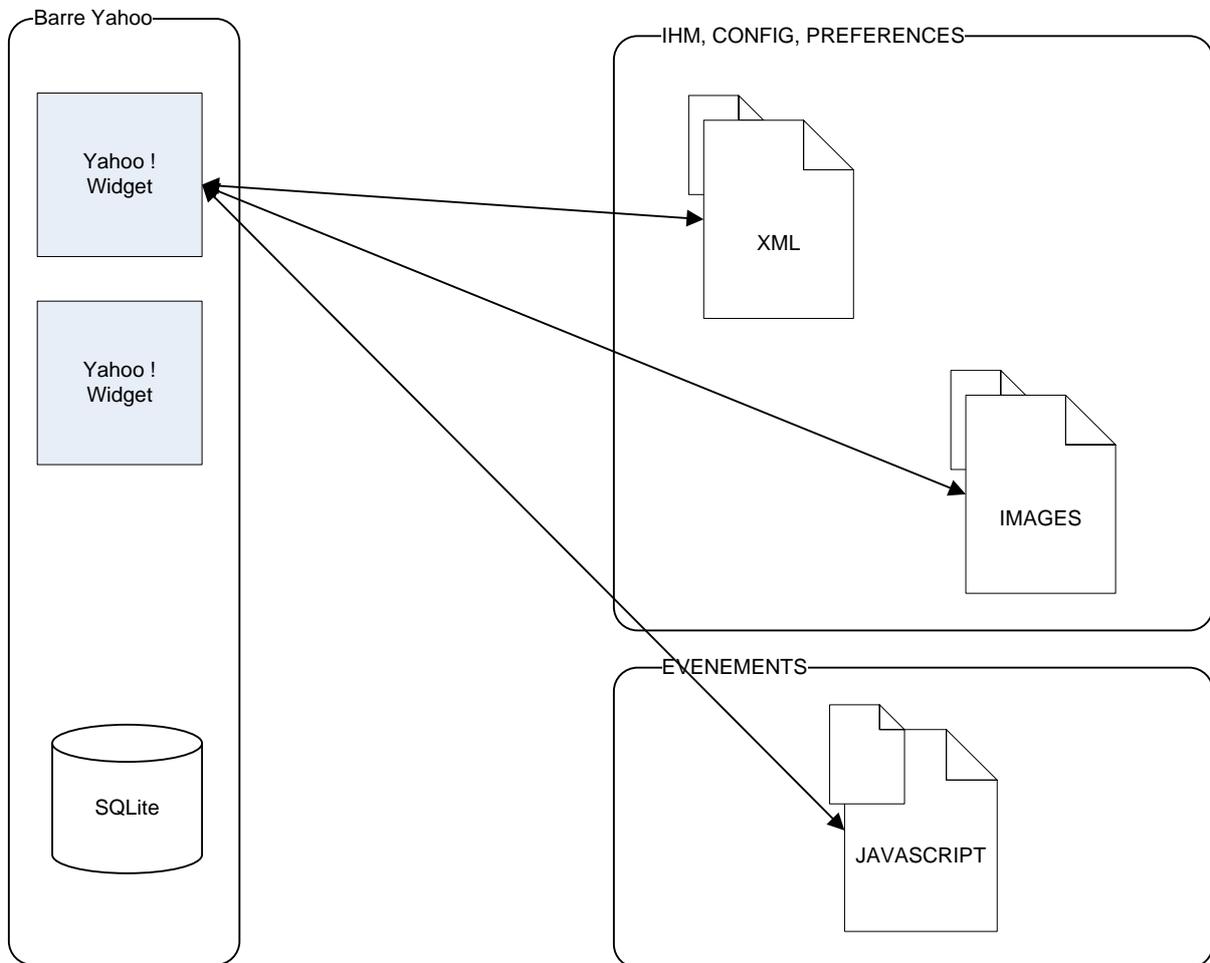
- o [*Yahoo! Widgets 4.5 for Windows : Barre Yahoo,*](#)
- o [*Konfabulator™ Reference Manual v4.5 \(PDF\) : Guide de développement,*](#)
- o [*Widget Converter Tool v1.3 : Permet de convertir le projet widget en « .widget »,*](#)
- o [*Widget Creation Tutorial v1.1 : tutorial expliquant comment créer un widget.*](#)

Il n'existe pas d'IDE pour développer les widget. Cependant, il suffit de prendre un simple éditeur comme Visual studio, Notepad ou Ultratedit pour pouvoir modifier l'affichage ou les traitements du Widget.

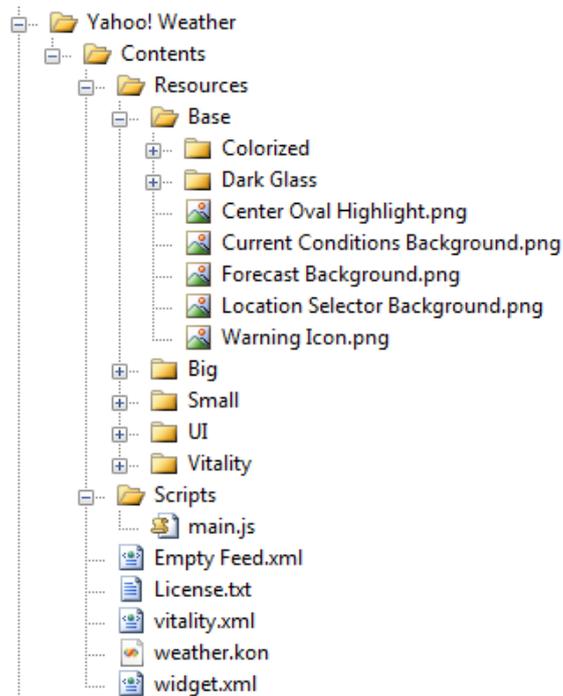


Architecture d'un Widget

Le schéma ci-dessous présente l'architecture d'un widget Yahoo. On peut voir qu'un Widget est en fait un ensemble de fichiers XML, JavaScript et d'images.



Ces éléments sont regroupés de manière arborescente dans des répertoires (voir capture d'écran ci-dessous) :



A la différence des Widgets Vista, les Widgets Yahoo ne s'exécutent pas dans un véritable navigateur web, c'est ce qui fait qu'ils ne sont pas décrits par des fichiers HTML. Par contre, la barre Yahoo intègre un véritable moteur JavaScript.

Développer un Widget Yahoo

Développer l'interface et les traitements

Les Traitements sont développés en JavaScript. Ils peuvent soit être externalisés dans un fichier JS soit être intégrés dans le fichier XML.

Les différents éléments d'un Description des différents éléments :

- **Fichier widget.xml** : Fichier de description du Widget,
- **XML (.xml ou .kon)** : Permet la mise en forme du widget. Permet également de décrire la configuration du widget,
- **JavaScript** : Utilisés pour les traitements,
- **Images** : Images référencées par le widget.

```
<?xml version="1.0" encoding="utf-8"?>
<metadata>
  <name>Yahoo! Weather</name>
  <version>3.0</version>
  <identifiant>com.yahoo.widget.weather</identifiant>
  <image usage="dock" src="Resources/Vitality/Inactive.png"/>
  <author name="Arlo Rose, Marcus Harvey, Rob Marquardt" organization="Yahoo! Inc." href="http://www.yahoo.com"/>
  <copyright>(c) 2003-2007 Yahoo! Inc.</copyright>
  <description>Shows the current weather conditions a location of your choice, as well as a 5-day forecast and links to the extended forecast at We:
  <platform minVersion="4.0.3" />
</metadata>
```

Figure 6: Exemple de fichier XML .widget

Le fichier **[nom Widget].kon** : est le cœur d'un Widget Yahoo. En particulier, il contient la description sous forme XML de :

L'IHM :

Les balises utilisées sont :

- la balise `<widget></widget>` : qui contient tous les autres éléments,
- la balise `<window></window>` : qui est la fenêtre principale du widget,
- la balise `<frame></frame>` : qui permet de créer des zones d'affichages,
- la balise `<text></text>` : qui permet d'afficher du texte.
- la balise `<image></image>` : qui permet d'afficher une image.

Les actions :

Toutes les actions sont définies en JavaScript. Elles peuvent soit être intégrées dans le fichier XML .kon, soit dans un fichier .js. Lorsqu'elles sont dans le fichier XML, elles sont situées entre des balises CDATA.

Les timers :

Les timers sont définis dans le fichier XML. Les balises `<timer></timer>` permettent de déclencher un traitement à intervalle régulier.

```
<!-- Timers -->
<timer name="connectionTester" interval="60" ticking="false" onTimerFired="fetchData( )"/>
<timer name="sleepPauseTimer" interval="7" ticking="false" onTimerFired="fetchData( ); weatherFetchTimer.ticking=true; sleepPauseTimer.ticking=
<timer name="controlsTimer" interval="4" ticking="false" onTimerFired="controlsTimerFired( )"/>
<timer name="weatherFetchTimer" interval="301" ticking="true" onTimerFired="fetchData( )"/> <!-- Setting this to a lower value won't retrieve ne
```

Les préférences :

Les préférences sont représenté par des balises XML `<preference></preference>` et permettent de positionner dans la fenêtre différents contrôles comme des listes déroulantes, checkbox ...

Déploiement

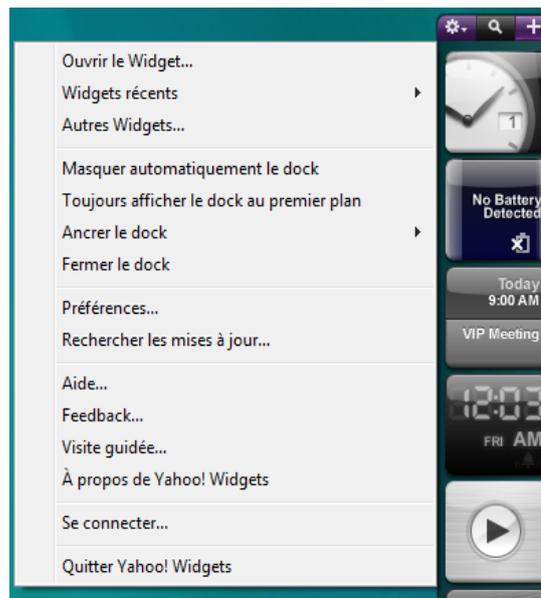
Pour déployer un nouveau widget, il suffit de respecter l'architecture de répertoire ci-dessus.

Une fois les développements terminés :

- Renommer le répertoire [Nom Widget] en [Nom Widget].widget
- Lancer la ligne de commande suivante :
`Converter.exe -flat -o [Chemin de destination] [Nom Widget].widget`

Cette ligne de commande permet de compresser tous les fichiers dans un fichier unique .widget

- Ouvrir le widget dans la barre Yahoo :



Le widget est alors disponible dans la barre Yahoo.

Implémentation des principales fonctionnalités

Boite de configuration

La configuration d'un Widget Yahoo se fait par l'intermédiaire de XML et de Javascript. La boite de configuration est gérée en grâce à la balise `<preference></preference>`. On peut ajouter des éléments (label, checkbox...) dans le fichier .kon. Par exemple:

```
<preference name="nightTextColor">
  <title>Night Text:</title>
  <group>colorPanel</group>
  <type>Color</type>
  <defaultValue>#ffffff</defaultValue>
  <description>Color will be used for text when the Widget is
  Colored or has no background.</description>
</preference>
```

La modification des préférences est signalée par l'évènement `onPreferencesChanged` et l'objet `preferences` permet d'accéder aux contrôles dans la fenêtre.

Exemple :

```
<action trigger="onPreferencesChanged">
  <![CDATA[
    if(preferences.Test.value == "Vert")
    {
      background.src="Resources/Background.png";
    }
    else
    {
      background.src="Resources/Background2.png";
    }
  ]]>
</action>
```

Accès à une URL distante en HTTP

L'accès à une URL distante en http se fait en JavaScript avec l'objet **XMLHttpRequest** utilisé habituellement pour réaliser des requêtes AJAX.

Exemple :

```
var request = new XMLHttpRequest( );
request.open("GET","http://c2ssp007/WSAnnuaireGpeC2S/ServiceGroupe.asmx/get
Collaborator?name="+TextBox.data+"&givenName=&initials=&company=&maxResults
=5", true);
request.onreadystatechange = function ( )
{
    if( this.readyState == 4 ) {
        if( this.status == 200 ){
            var doc = this.responseXML;var element =
            doc.getElementsByTagName('telephoneNumber').item(0);
            if(doc.getElementsByTagName('telephoneNumber').length ==
            1){testHerve.data = element.firstChild.data ;}else{
            testHerve.data = "Trop de resultats." ; } }else{ testHerve.data
            = "Aucun resultat pour "+TextBox.data; } }
};
request.send( );
```

Rafraichissement des données à intervalle régulier

Les rafraichissements se font grâce aux balises « timers » intégrés dans le fichier XML.

Exemple :

```
<timer name="connectionTester" interval="60" ticking="false"
onTimerFired="fetchData( )"/>
```

Accès à une information locale de la machine (date, fichier, ...)

L'accès à une information locale de la machine peut être effectué en JavaScript via une API spécifique proposée par la barre Yahoo.

Interaction clavier ou souris avec le Widget

Les interactions clavier / souris se font grâce au JavaScript et sont décrits dans le fichier XML. On peut utiliser les évènements habituels : onkeypress, onkeyup, onmousemove...

Exemple d'applications: Sytadin, Bouygues Epargne Salariale, Annuaire

Pendant l'étude, nous avons réalisé un application accédant au Web Service de l'annuaire groupe.

En voici la description :



L'appel est fait en JavaScript et l'IHM est géré par L'XML

Ex :

```
TestIMG.onClick = function ()
{
    var request = new XMLHttpRequest( );
    request.open(                                     "GET",
    "http://c2ssp007/WSAnnuaireGpeC2S/ServiceGroupe.aspx/getCollaborator?
    name="+TextBox.data+"&givenName=&initials=&company=&maxResults=5",
    true);
    request.onreadystatechange = function () { if( this.readyState == 4
    ){ if( this.status == 200 ){ var doc = this.responseXML;var element =
    doc.getElementsByTagName('telephoneNumber').item(0);
    if(doc.getElementsByTagName('telephoneNumber').length           ==
    1){testHerve.data = element.firstChild.data ;}else{ testHerve.data =
    "Trop de resultats." ; } }else{ testHerve.data = "Aucun resultat pour
    "+TextBox.data; } } };
    request.send( );
}
}
```

IHM :

```
<window title="Digital Clock">
  <name>mainWindow</name>
  <width>216</width>
  <height>85</height>
```

```

<visible>false</visible>
<shadow>0</shadow>

<image src="Resources/Background.png">
  <name>background</name>
  <hOffset>0</hOffset>
  <vOffset>0</vOffset>
</image>
<frame name="forecastTabButtonInactive" vOffset="3" width="200"
height="100">
<text
  data      = "Nom : "
  hAlign    = "center"
  name      = "testHerve"
  hOffset   = "65"
  vOffset   = "18"
  style     = "font-family: 'Arial Bold'; font-size: 11px; color:
#FFFFFF; opacity: 0.5; -kon-shadow: 0px 1px #000000"
/>
<textArea
  name      = "TextBox"
  hOffset   = "50"
  vOffset   = "25"
  width     = "110"
  style     = "font-family: 'Arial'; font-size: 12px; background-
color: #c0c0c0; color: #000000;"
  editable  = "true"
  lines     = "1"
  scrollbar = "false"
/>
<text
  data      = ""
  hAlign    = "center"
  name      = "testHerve"
  hOffset   = "105"
  vOffset   = "75"
  style     = "font-family: 'Arial Bold'; font-size: 11px; color:
#FFFFFF; opacity: 0.5; -kon-shadow: 0px 1px #000000"
/>
<image src="Resources\Floater Button.png">
  <name>TestIMG</name>
  <hOffset>65</hOffset>
  <vOffset>45</vOffset>
</image>
</frame>
</window>

```

Synthèse sur la plate-forme

Yahoo ! Widget est l'ancêtre des plates-formes de Widgets existantes. Il dispose d'une base installée très importante et d'un support multi-plateforme intéressant : Windows XP, Windows Vista et Mac.

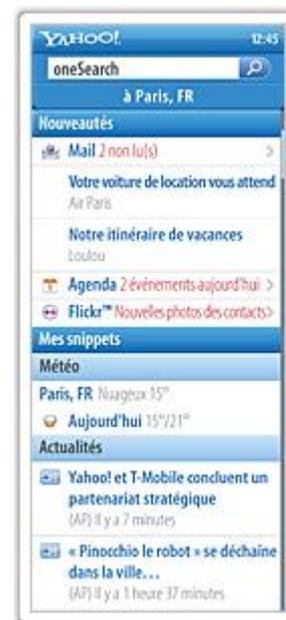
Yahoo ! Mobile

Présentation Générale

Yahoo ! Mobile est l'adaptation de la plate-forme Yahoo ! Widgets pour les terminaux mobiles. Yahoo ! Mobile est une application appelée Yahoo ! GO installable sur un grand nombre de terminaux mobiles, y compris les nouvelles plates-formes Android et iPhone. Yahoo ! GO propose un ensemble de services « standards » : actualités, météo, cours de la bourse, ... qui se mettent à jour via une connexion Internet.

Deux types de Widgets Mobiles existent:

- Les snippets : minis Widgets qui apparaissent sur la « home page » ,
- Les widgets : un Widget en lui-même.



Ce paragraphe décrit la manière d'implémenter de nouveaux Widgets à installer sur Yahoo ! GO. On ne s'intéressera pas aux snippets qui ne sont qu'un sous ensemble des fonctionnalités des Widgets.

A noter que même si Yahoo ! Widget et Yahoo ! Mobile sont tous les deux développés par Yahoo, il n'y a aucune compatibilité entre ces deux plates-formes.

Architecture logicielle des Widgets Yahoo Mobile

Pré-requis logiciels

Le développement des Widgets Mobile, nécessite :

- **L'installation de Yahoo ! GO :**
Solution Yahoo permettant la gestion des Widgets, messages, navigation web. A installer sur votre mobile.
- **L'installation d'un SDK,** Il contient :
 - Documentation BluePrint : Le BluePrint est un framework CSS permettant la mise en forme du Widget,
 - Le Schema XML des pages à réaliser,
 - Des exemples de Widget.

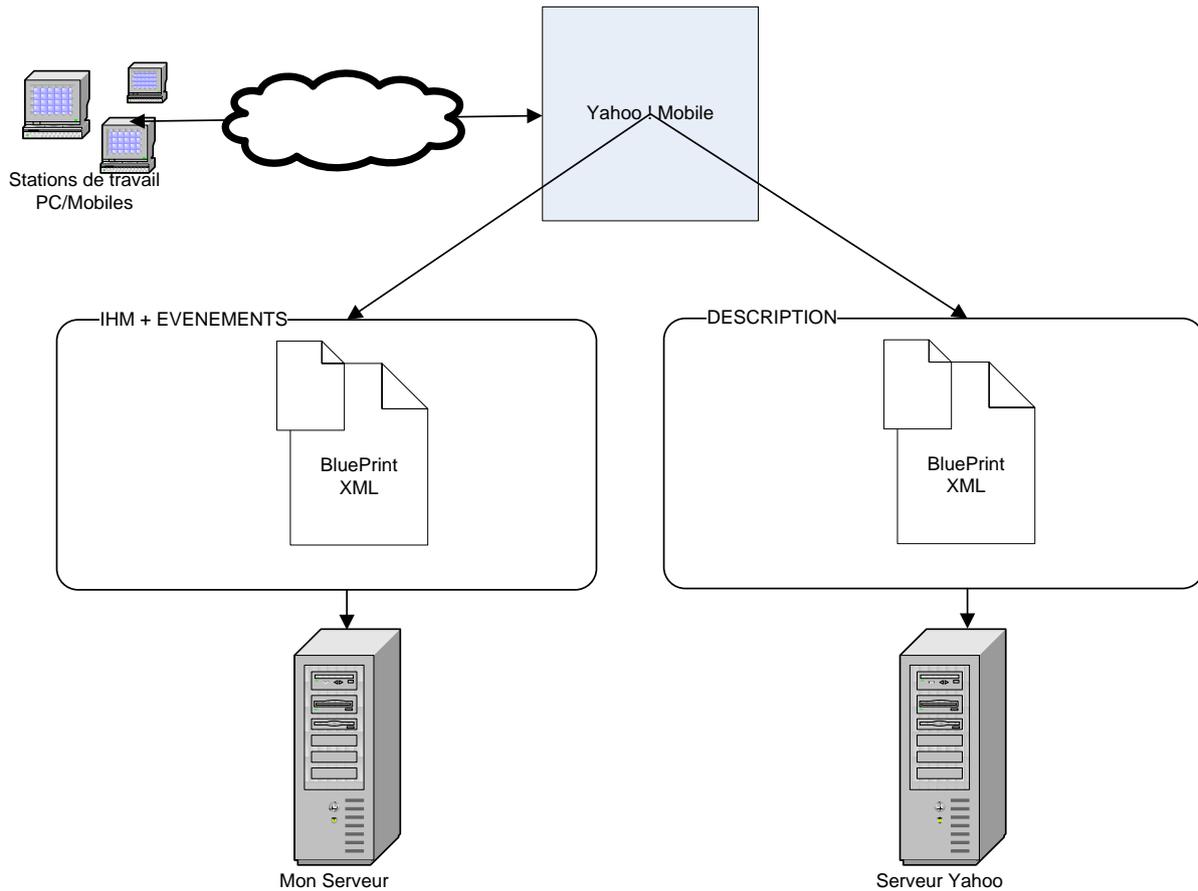
Par contre, il n'existe pas à proprement parler d'outil de développement pour réaliser des Widgets Mobile.

Architecture d'un Widget

A la différence des autres plateformes, les Widgets Mobiles ont une « intelligence » très limitée. L'essentiel des traitements est donc réalisé sur le serveur avec lequel ils doivent communiquer pour chaque action qu'ils ont à exécuter.

Le schéma suivant décrit l'infrastructure nécessaire. Elle se décompose en deux éléments :

- Le serveur Yahoo où sont centralisées les descriptions de tous les Widgets Mobiles existants,
- Le serveur de traitement qui exécute les actions déclenchées par les Widgets.

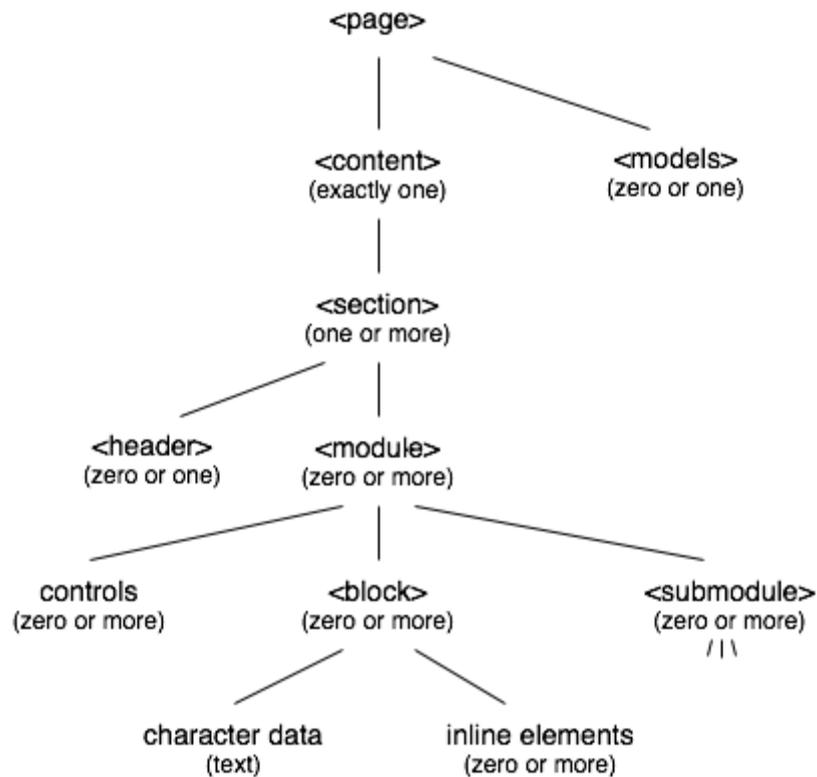


Développer un Widget Yahoo Mobile

Développer l'interface

Les Widgets Mobiles sont implémentés en XML dans un langage appelé BluePrint. Le BluePrint permet de décrire toute l'interface d'un Widget.

La structure d'une page est la suivante :



Exemple :

```

<page>
  <content>
    <section>
      <header layout="simple">
        <layout-items>
          <block class="title">Breaking News!</block>
        </layout-items>
      </header>
      <block>New York Giants win Super Bowl 2008! Final score
17-14</block>
      <navigation-bar>
        <next>
          <label>Read more</label>
          <load event="activate" resource="http://..."
        </next>
      </navigation-bar>
    </section>
  </content>
</page>
  
```

Trois types d'éléments sont décrits:

- Les Containers : qui permettent de grouper des éléments.
- Les Controls : qui permettent d'afficher différents contrôles (textarea,input,)
- Les Inline elements

Développer les traitements

Les traitements sont également décrits en XML (Blueprint). Le langage est basé sur les spécifications XForms. Pour traiter des données, il faut implémenter des modèles qui permettent de définir l'action souhaitée. Par exemple pour gérer un submit il faut implémenter le modèle :

```
<models>
  <model>
    <submission method="urlencoded-post" resource="http://.../" />
  </model>
</models>
```

Les contrôles ont un attribut « ref » qui permet de binder des champs à l'écran et d'envoyer des données au serveur.

Exemple :

```
<input ref="name">
  <label>Name:</label>
</input>
<submit>
  <label>Submit</label>
</submit>
```

Au click sur le bouton, l'évènement submission est activé. L'évènement « send » permet aussi d'envoyer des données.

Les traitements sont déclenchés coté serveur qui doit donc disposer d'un langage serveur comme le PHP. Une classe PHP est d'ailleurs fournie dans le SDK « Blueprint.php » et permet de générer les pages serveurs dynamiquement.

Déploiement

Pour déployer un nouveau Widget, il faut tout d'abord respecter son architecture et posséder un identifiant yahoo. Une fois les développements terminés :

- Déployer le fichier Blueprint sur votre serveur. (Penser à ajouter l'extension .bp ou utiliser un .htaccess)
- Zipper le répertoire qui contient le fichier config.xml, gallery.xml, les images et les screenShots.

- Uploader le fichier .zip à l'adresse suivante : <http://mobile.yahoo.com/developers/test/upload>
- Si aucune erreur ne survient, le widget est alors disponible sur votre home Page.

Implémentation de certaines fonctionnalités

Boite de configuration

Une boite de configuration peut être implémentée par les snippets. Il existe un contrôle settings qui permet de définir les paramètres du widget.

Exemple :

```
<snippet>
  <settings>
    <input ref="city">
      <label>City:</label>
    </input>
    <input ref="state">
      <label>State:</label>
    </input>
    <input ref="zip">
      <label>Zip Code:</label>
    </input>
    <select1 ref="list_type">
      <item>
        <value>short</value>
        <label>Highlights</label>
      </item>
      <item>
        <value>long</value>
        <label>All</label>
      </item>
    </select1>
  </settings>
  ...
</snippet>
```

Accès à une URL distante en http

L'accès à une URL distante est possible via les modèles. Il suffit de définir les modèles comme si dessous :

```
<models>
  <model>
    <submission method="get" resource="http://.../" />
  </model>
</models>
```

Les traitements sont alors déclenchés côté serveur.

Rafraîchissement des données à intervalle régulier

Il n'existe aucune méthode pour rafraîchir des données à intervalle régulier.

Interaction clavier ou souris avec le Widget

L'interaction avec le clavier et la souris se fait grâce aux actions send, activate, submission...dans le fichier XML BluePrint.

Le clavier est directement accessible dès qu'on se positionne sur un champ de saisie.

Synthèse sur la plate-forme

Yahoo Mobile est présentée comme l'équivalent en mobilité de la plate-forme Yahoo Widget. Elle dispose néanmoins de fonctionnalités très limitées qui reposent principalement sur l'appel de fonctionnalités serveurs. De plus l'installation nécessaire d'un runtime limite son intégration sur le terminal.

Windows Mobile

Présentation générale

Windows Mobile est le système d'exploitation proposé par Microsoft pour équiper des terminaux mobiles et notamment des Smartphones. Windows Mobile est fortement déployé en entreprise notamment grâce à sa capacité à se synchroniser avec Exchange. Un autre avantage de Windows Mobile est, comme nous le verrons ci-dessous, de partager les mêmes outils de développement que la plate-forme Windows.

Mise à part l'installation d'applications, il est possible de personnaliser Windows Mobile en réalisant des mini-applications qui s'installent dans l'écran « Aujourd'hui » du terminal.

Ce paragraphe présente l'étude réalisée sur le développement d'une application pour l'écran « Aujourd'hui » de Windows mobile. Il faut noter qu'un certain nombre d'éditeurs ou de constructeur proposent un remplacement complet de cet écran pour y installer un mécanisme propriétaire de Widget (par exemple HTC ou PointUI).



Architecture logicielle de l'écran Aujourd'hui

Un « TodayScreen Item » est une application Windows Mobile qui permet une interaction avec l'utilisateur depuis l'écran « Aujourd'hui » d'un appareil Windows Mobile.

Cet item est constitué d'une DLL en C/C++ qui doit contenir une « classe » implémentant une interface spécifique.

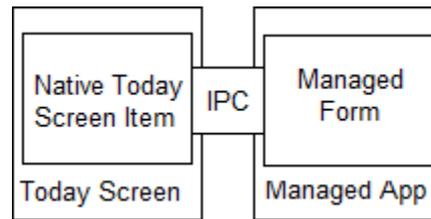
Plus précisément, cette DLL doit implémenter :

- Le point d'entrée : **DllMain**
- La fonction **InitializeCustomItem** qui charge l'item sur le bureau
- La fonction **CustomItemOptionsDlgProc** qui permet de lancer une boîte de configuration (optionnelle)
- Le callback **Wndproc** pour être notifier des événements de l'utilisateur.

Le TodayScreen est en donc le point d'entrée d'un Widget pour Windows Mobile.

Bien qu'il soit développé en C/C++, il peut inter-opérer (appel système, encapsulation MC++, ...) avec une application réalisée à l'aide du Compact Framework .NET. C'est la solution que nous privilégions ici.

Le développement d'une application pour le Today Screen L'architecture d'un gadget Windows mobile se traduit donc de la façon suivante :



Nous pouvons voir qu'elle se décompose en deux parties, l'une en C/C++ qui s'intègre dans le Today Screen et l'autre pour l'application .NET en elle-même.

Développer une application pour l'écran Aujourd'hui

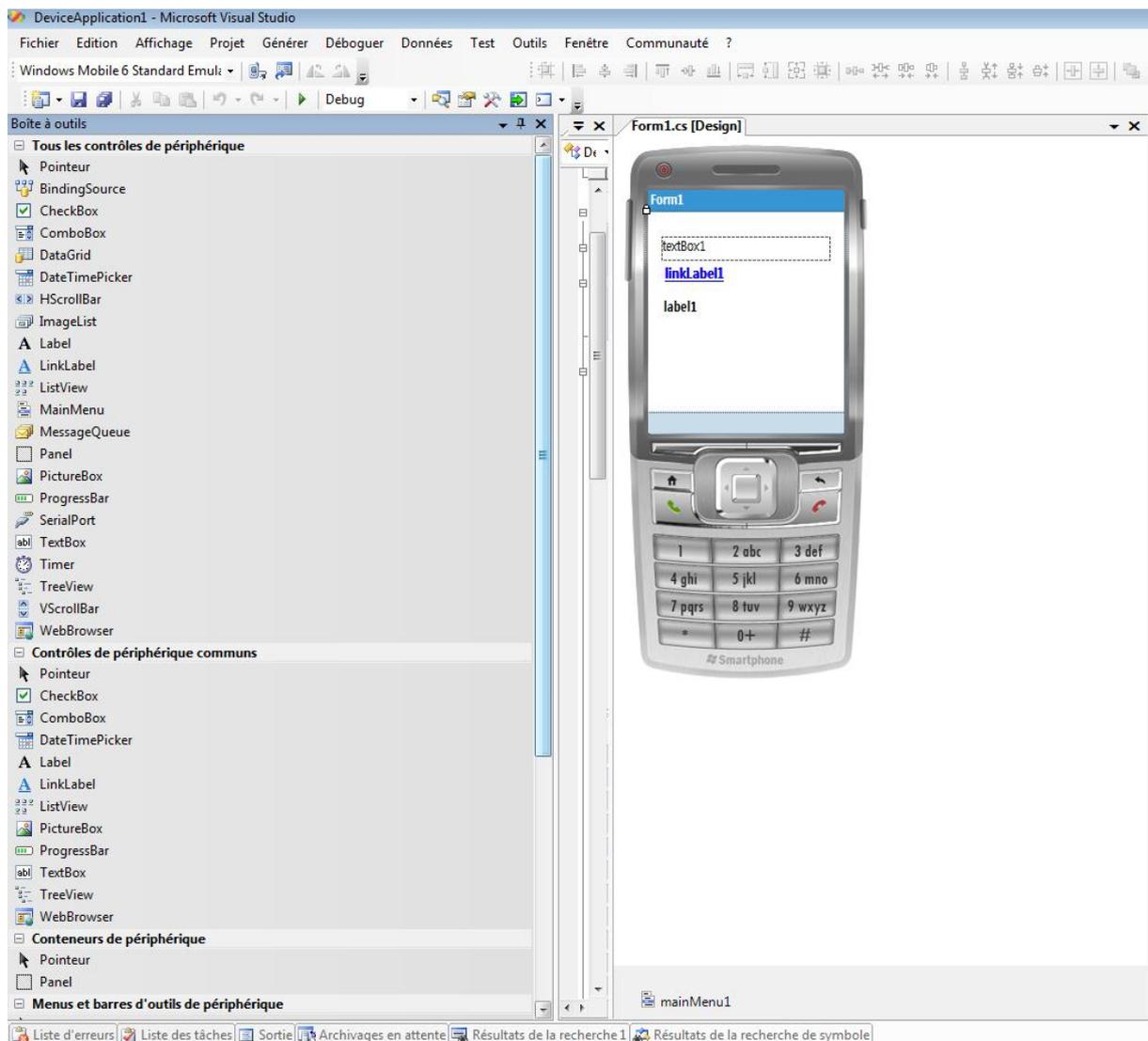
Outils de développement

Le développement d'application Mobile nécessite l'installation de :

- Microsoft Visual Studio 2005/2008.
- Microsoft .NET Compact Framework v2 SP1. SP2 recommandé.
- ActiveSync.
- Windows Mobile 6 Professional SDK, ou Windows Mobile 6 Standard SDK ou les deux.

Le .Net Compact Framework de Microsoft est une version du .NET Framework spécialement conçue pour être exécutée sur des terminaux mobiles de type Windows CE ou Windows Mobile tels que des PDAs, des téléphones portables, des panels PC industriels... Le .Net Compact Framework est un sous-ensemble du .Net Framework, certaines classes sont communes avec le .Net Framework et d'autres sont spécifiques au Compact Framework comme l'InputPanel Windows CE. Le Compact Framework .NET supporte le développement avec **Visual Basic** et **Visual C#**.

L'utilisation du .NET Compact Framework pour la réalisation d'une application pour le TodayScreen permet d'utiliser les mêmes outils de développement que ceux utilisés pour développer une application client lourd pour C2S. L'interface et les traitements sont donc réalisés directement dans Visual studio.



Déploiement

Le déploiement d'une nouvelle application sur Windows Mobile nécessite la réalisation d'un fichier « .CAB ». Dans le cas de la réalisation d'un Widget pour le TodayScreen, le .CAB contient deux principaux éléments:

- La DLL du TodayScreen qui doit être inscrite dans la base de registre (par le .CAB, cf. Figure suivante)
- l'application .NET elle-même (.EXE).

Il suffit alors d'exécuter le .CAB sur le mobile pour installer le Widget.

A noter que Microsoft a lancé un MarketPlace dont l'objectif est d'héberger les applications Windows Mobile comme l'AppStore de l'iPhone.

```
HKLM,Software\Microsoft\Today\Items\My Today Item,Type,0x00010001,4
HKLM,Software\Microsoft\Today\Items\My Today Item,Enabled,0x00010001,1
HKLM,Software\Microsoft\Today\Items\My Today Item,Options,0x00010001,1
HKLM,Software\Microsoft\Today\Items\My Today Item,DLL,0x00000000,"\Windows\Sample.DLL"
```

Figure 7-clés à inscrire sur le Device pour l'item « My Today Item » dont l'implémentation se trouve dans « \Windows\Sample.DLL »

Implémentation des principales fonctionnalités

Boite de configuration

Le TodayScreen intègre un point d'entrée pour implémenter une boite de configuration (donc au niveau de la DLL C).

Cependant, il est possible d'implémenter une fenêtre standard .NET dédiée à la configuration et qui serait appelé par la DLL en C.

Accès à une URL distante en HTTP

Dans le Compact Framework .NET, le namespace System.web contient une version réduite des capacités et fonctionnalités offerte par le framework .NET complet pour la manipulation de requête HTTP.

De plus l'IDE de Visual Studio nous facilite l'appel à un web services en proposant une fonction « ajouter une nouvelle référence web ».

Rafraichissement des données à intervalle régulier

- Dans l'application .NET :
Pour gérer le rafraichissement des données, on peut utiliser le contrôle timer avec son évènement Tick.

```
private void timer1_Tick(object sender, EventArgs e)
{
}
}
```

On peut également intégrer un Timer et des délégués ou thread pour exécuter une méthode à intervalle régulier.

```
Timer t = new Timer(delegate(object o) { Debug.Print("Hello World!"); }, null, 0, 5000);
```

- Depuis la DLL TodayScreen sur le bureau :
Comme évoqué précédemment, la DLL du TodayScreen intègre une boucle des messages.

Un message spécifique permet notamment de forcer le rafraichissement de l'item sur le bureau à intervalle régulier.

Accès à une information locale de la machine (date, fichier, ...)

Les différentes APIs standards du .NET nous permettant facilement d'accéder aux informations locales de la machine. Par exemple via le namespace System.IO pour l'accès à des fichiers sur le terminal.

Interaction clavier ou souris avec le Widget

- Dans l'application .NET :
L'interaction clavier souris se fait grâce aux évènements .NET habituels liés aux contrôles.

Par exemple, 'Form1_KeyDown(object sender, KeyEventArgs e)' va permettre de gérer l'appuie sur les touches de l'appareil.
- Depuis la DLL en C TodayScreen sur le bureau
La DLL du TodayScreen intègre une boucle des messages. Le message **WM_LBUTTONDOWN** permet notamment d'être notifier du click sur le todayscreen.

Synthèse sur la plate-forme

Windows Mobile est une plate-forme relativement ancienne qui propose depuis le début la possibilité d'intégrer des applications dans son écran d'accueil. Pourtant la relative complexité de développement de ce type d'applications et le manque d'ergonomie pour leur installation fait que de nombreuses plateformes alternatives ont vue le jour. Aucune ne recueille néanmoins une adoption universelle.

Samsung Touch Wiz



Présentation générale

Touch Wiz est l'interface de Widgets proposée sur les terminaux Samsung. L'interface est disponible à la fois sur des terminaux Windows Mobile (Omnia) et sur des terminaux reposant sur le système d'exploitation propriétaire de Samsung (Pixon, Ultra).

Sur les premiers terminaux Samsung supportant Touch Wiz, les Widgets n'affichaient que des informations locales (Heure, Date, Image de fond, ...) ou étaient considérées comme des raccourcis vers des applications installées. Depuis le Pixon, Touch Wiz permet également l'affichage de contenus externes et il existe un kit de développement pour réaliser de nouveaux Widgets.

Architecture logicielle des Widgets Samsung

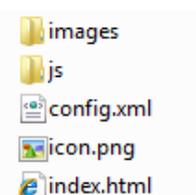
Description générale

Les Widgets Samsung correspondent à des pages HTML incluant des traitements JavaScript et des ressources (images et feuilles de styles).

Les Widgets s'exécutent sur le navigateur de la plateforme : cela peut être selon le cas Opera, Access, Jasmine ou Pocket IE. Le fonctionnement des Widgets reste cependant identique quelque soit le navigateur.

Architecture d'un Widget

Les Widgets Samsung s'appuient sur la spécification <http://www.w3.org/TR/widgets/> en cours d'élaboration au W3C. Concrètement, il s'agit d'un fichier compressé reprenant différents fichiers et répertoires décrits ci-dessous :



Le fichier **config.xml** contient les informations de configuration du Widget : son identifiant, son nom, sa description, l'icône et la page d'accueil.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <widget id="SYTADINC2SWIDGET" version="0.01" height="103" width="100" xmlns="http://www.w3.org/ns/widgets">
  <title>C2S Sytadin</title>
  <description>Gadget Sytadin par C2S</description>
  <icon src="icon.png" />
  <content src="index.html" />
  <access network="true" />
</widget>
```

Le fichier **index.html** correspond à la page d'accueil et aux différentes pages du Widget. En effet, pour des raisons de performance, l'ensemble des vues du Widget sont chargées dans une même page et sont affichés successivement en changeant la visibilité des éléments « div » auquel elles correspondent.

Les répertoires **images**, **js** (et éventuellement **css**) contiennent les ressources utilisées par le Widget.

Développer un Widget Samsung

Outils de développement

Le kit de développement des Widgets Samsung n'est pas public et n'est distribué qu'aux opérateurs et aux partenaires de Samsung. Ce kit se limite aujourd'hui à un document décrivant le format des fichiers attendus, la référence des APIs et un guide de design des écrans. Une plate-forme de développement intégrant un simulateur est à l'étude mais n'est pas encore disponible.

Samsung n'a pas communiqué sur l'ouverture de l'API au grand public. Il est néanmoins assez facile de se lancer dans la création de nouveaux Widgets en faisant du reverse engineering des Widgets existants.

Déploiement

Un Widget correspond à un fichier compressé contenant ses différents fichiers et auquel on donne une extension « .wgt ».

Samsung propose une plate-forme web sur laquelle il est possible de télécharger depuis un terminal des nouveaux Widgets. Néanmoins, l'ajout de nouveaux Widgets sur cette plateforme dépend d'un accord entre Samsung et l'opérateur qui distribue le terminal et n'est donc pas ouvert aux entreprises ou au grand public. L'ouverture prochaine d'un « AppStore » a cependant été évoquée.

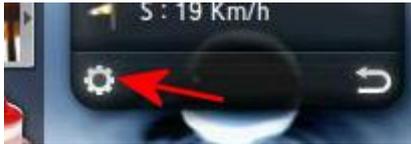
Il est malgré tout possible de télécharger ses propres Widgets sur un terminal. Il faut pour cela :

- mettre à disposition le fichier « .wgt » sur le web,
- s'assurer que le serveur http retournera le type MIME « application/vnd.samsung.widget » lors des demandes d'accès au fichier,
- lancer l'URL où se trouve le fichier depuis le navigateur du terminal.

Implémentation des principales fonctionnalités

Boîte de configuration

Il n'existe pas de mécanisme de configuration unifié pour les Widgets Samsung. Néanmoins, le guide de conception recommande la mise en place d'un icône spécifique pour accéder à un écran de configuration.



Les méthodes **widget.SetPreferenceForKey()** et **widget.preferenceForKey()** de l'API Samsung permettent ensuite de stocker et de récupérer les valeurs des paramètres de configuration.

Accès à une URL distante en http

Quelque soit le navigateur de la plateforme, il est possible d'utiliser des requêtes AJAX pour accéder à des données distantes. Il faut néanmoins tenir compte lors de l'appel des syntaxes spécifiques de chaque navigateur :

```
var request;
try{
    request = new XMLHttpRequest();
} catch (e){
    try{
        request = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try{
            request = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e){
            // Error !
        }
    }
}
```

Une autre manière d'accéder à une URL distante est de référencer dans la page du Widget une image se trouvant sur Internet.

Rafraîchissement des données à intervalle régulier

Les Widgets Samsung supportent les méthodes traditionnelles JavaScript de positionnement de timers sur une fenêtre HTML : **setInterval()** et **setTimeout()**.

```
window.setInterval(myfunction, 180000);
```

Accès à une information locale de la machine (date, fichier, ...)

Les Widgets Samsung peuvent accéder aux informations systèmes accessibles nativement en JavaScript (date et heure essentiellement). Des APIs spécifiques permettent également d'interroger l'état du réseau. Il est aussi possible de déclencher l'ouverture du navigateur de la machine sur une URL particulière en utilisant la méthode **widget.openURL()**.

Par contre, aucune API documentée ne permet d'accéder aux fichiers de la machine, à l'appareil photo ou aux contacts.

Interaction clavier ou souris avec le Widget

Les événements clavier et souris liés aux balises HTML sont normalement supportés par les Widgets Samsung. Aucune API spécifique n'est fournie.

Exemple de fonctionnalités: Sytadin, Bouygues Epargne Salariale, Annuaire C2S

C2S a développé le Widget Sytadin sur la plate-forme Touch Wiz. Il propose l'affichage d'une carte réduite, la configuration de la carte via une boîte de configuration, le rafraîchissement, et l'accès au détail de la carte à travers le navigateur.

**Synthèse sur la plate-forme**

La plate-forme Touch Wiz propose une approche permettant de supporter de manière transparente différents types de terminaux et de systèmes d'exploitation. L'utilisation du HTML et du JavaScript et la volonté de reposer sur les standards permettent le développement rapide d'un Widget.

L'inconvénient majeur est l'absence de volonté d'ouverture de Samsung qui empêche nécessairement le développement d'un véritable écosystème autour de Touch Wiz.

webOS

Présentation générale

WebOS est le système d'exploitation du Palm Pré. Le Palm Pré est un terminal mobile proposé par Palm en Juin 2009. Il est considéré comme un « iPhone killer » (concurrent de l'iPhone) auquel il apporte un véritable clavier et la possibilité d'exécuter plusieurs applications simultanément en passant de l'une à l'autre par l'intermédiaire d'un « Carrousel ».

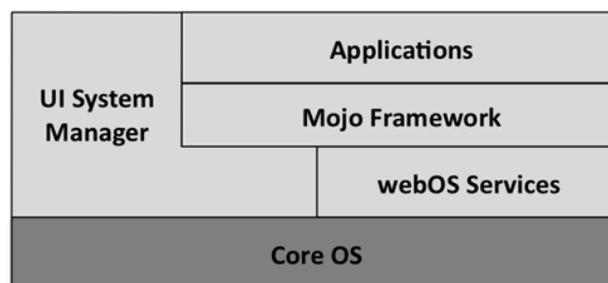
La particularité de WebOS est d'être un système d'exploitation sur lequel les applications sont considérées comme des applications web ce qui permet aux développeurs de s'approprier rapidement l'environnement.



Architecture logicielle d'une application webOS

Description générale

Le schéma ci-dessous présente une vue simplifiée de l'architecture de webOS.



Le cœur de web OS est basé sur un noyau Linux. Il implémente la gestion des drivers pour le système et les différents périphériques. Au dessus de ce noyau, les applications peuvent accéder à la gestion des fenêtres (UI) et aux fonctionnalités du système (webOS) par l'intermédiaire du framework JavaScript Mojo. La particularité de webOS est en effet que toutes les applications reposent sur JavaScript et sur HTML pour communiquer avec le système.

Les applications qui s'exécutent sur le système peuvent donc être considérées comme les onglets d'un navigateur que le Carrousel permet de faire défiler.

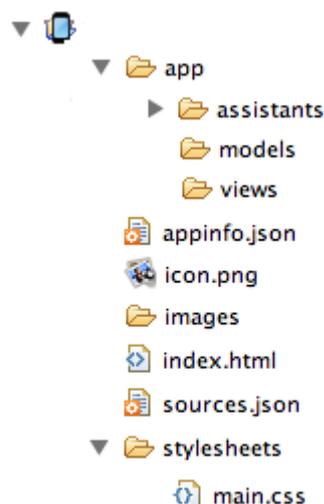
Architecture d'une application webOS

On ne parle pas de « Widgets » pour webOS mais d'applications. Le terme Widget étant utilisé par WebOS pour indiquer les contrôles (champs, boutons, ...) utilisables dans un écran.

Une application webOS est un ensemble de fichiers HTML, de fichiers JavaScript et de ressources (images et feuilles de styles). Le framework Mojo repose sur les concepts MVC (Modèle Vue

Contrôleur) et impose donc un certain découpage des fichiers pour séparer les vues et les traitements.

Le schéma ci-dessous représente l'organisation des répertoires d'une application webOS.



Le fichier **appinfo.json** contient sous forme d'une structure JSON les paramètres de l'application : son nom, numéro de version, l'icône associé, ...

Le fichier **index.html** correspond à la page d'accueil de l'application. Les différentes vues proposées sont ensuite décrites par les fichiers HTML se trouvant dans le répertoire **views**.

Le fichier **sources.json** contient une structure listant les différents traitements (fichiers .JS) manipulés par l'application. Ces traitements seront chargés dynamiquement par le contrôleur. Ils doivent être situés dans le répertoire **assistants** et seront associés à une des vues de l'application.

Les images et feuilles de styles sont situées dans des répertoires spécifiques : **images** et **stylesheets**.

Développer une application webOS

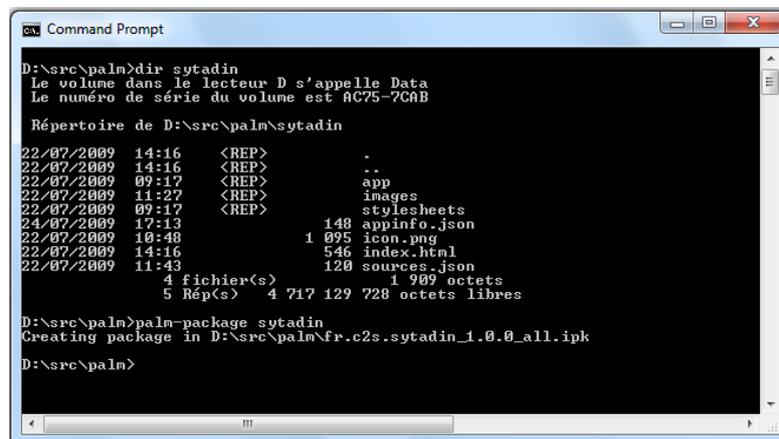
Outils de développement

Palm propose gratuitement un kit de développement permettant de développer des applications pour webOS. Ce kit de développement, qui est disponible pour Windows, MacOS et Linux intègre :

- Des outils de : génération, packaging et déploiement,
- Un émulateur basé sur l'outil VirtualBox,
- Des exemples d'utilisations des APIs,
- Le guide de référence du framework Mojo,
- Les sources du framework Mojo.

Déploiement

La première étape du déploiement d'une application WebOS consiste d'abord à la packager dans un fichier « .ipk ». Il s'agit schématiquement, d'un fichier compressé contenant l'ensemble des sources et ressources de l'application. La création d'un fichier « .ipk » se réalise simplement en lançant l'utilitaire « palm-package » du kit de développement sur le répertoire de l'application.



```
Command Prompt
D:\src\palm>dir sytadin
Le volume dans le lecteur D s'appelle Data
Le numéro de série du volume est AC75-7CAB

Répertoire de D:\src\palm\sytadin

22/07/2009 14:16 <REP>      .
22/07/2009 14:16 <REP>      ..
22/07/2009 09:17 <REP>      app
22/07/2009 11:27 <REP>      images
22/07/2009 09:17 <REP>      stylesheets
22/07/2009 17:13              149 appinfo.json
24/07/2009 10:48              1 095 icon.png
22/07/2009 14:16              546 index.html
22/07/2009 11:43              120 sources.json
                4 fichier(s)              1 909 octets
                5 Rép(s)        4 717 129 728 octets libres

D:\src\palm>palm-package sytadin
Creating package in D:\src\palm\fr.c2s.sytadin_1.0.0_all.ipk

D:\src\palm>
```

A ce jour, un package peut ensuite être installé sur le terminal :

- via l'AppStore de Palm : mais les conditions de déploiement sur l'AppStore ne sont pas encore connues,
- via un câble USB mais il faut alors passer le terminal en mode « développeur » (séquence de touche particulière à taper dans la recherche).

Implémentation des principales fonctionnalités

Boite de configuration

Contrairement à l'iPhone ou à Windows Mobile, il n'y a pas sur WebOS une gestion centralisée de la configuration des applications. Par contre, chaque application a la possibilité d'implémenter le menu « Préférences » qui lui permet de construire une page fournissant l'accès aux informations de configuration du terminal.



Les informations de configuration peuvent être stockées dans des cookies. Grâce au support de HTML 5 au niveau de WebOS, chaque application peut également disposer d'un espace de stockage SQL dédié, accessible via les APIs Mojo.

Accès à une URL distante en http

Comme toute application JavaScript, une application WebOS peut utiliser des requêtes AJAX pour accéder à des données depuis un site Internet. L'objet **XMLHttpRequest** est donc utilisable directement mais le framework Mojo propose également une encapsulation appelée **Ajax.Request**.

```
var request = new Ajax.Request(url, {
    method: 'get',
    evalJSON: 'false',
    onSuccess: this.requestSuccess.bind(this),
    onFailure: this.requestFailure.bind(this)
});
```

De plus comme les applications WebOS s'exécutent localement, elles ne sont pas limitées par les serveurs/domaines auxquels elles peuvent accéder.

Une autre manière d'accéder à une URL distante est de référencer dans la page de l'application une image se trouvant sur Internet. Le contrôle `ImageView` permet ainsi d'afficher une ou plusieurs images (par défilement vertical) dans une vue.

```
var attributes = {
    noExtractFS : true
};
this.model = {
    backgroundImage : 'images/glacier.png',
    onLeftFunction : this.wentLeft.bind(this),
    onRightFunction : this.wentRight.bind(this)
}
this.controller.setupWidget('myPhotoDiv', attributes, this.model);
this.myPhotoDivElement = $('myPhotoDiv');
this.myPhotoDivElement.mojo.centerUrlProvided('images/pre_01.png', 'images/pre_01_thumb.png');
```

Rafraîchissement des données à intervalle régulier

WebOS supporte les méthodes traditionnelles JavaScript de positionnement de timers sur une fenêtre HTML : **setInterval()** et **setTimeout()**.

Ces méthodes sont néanmoins peu précises et peu efficaces si la fenêtre est masquée. WebOS fournit donc également un mécanisme d'alarme temps réel qui permet à une application d'être réveillée/mis au premier plan à intervalle régulier.

Comme pour les autres services fournis par le système, cela fonctionne par un mécanisme d'abonnement via le contrôleur.

```
this.controller.serviceRequest('palm://com.palm.power/timeout', {
    method: 'set',
    parameters: params,
    onSuccess: this.onSuccessHandler,
    onFailure: this.onFailureHandler
});
```

Accès à une information locale de la machine (date, fichier, ...)

L'utilisation de HTML et de JavaScript peut sembler limité pour le développement d'une application intégrée au système. Cependant, l'une des forces de WebOS est de proposer l'ensemble des fonctionnalités systèmes à travers des services Mojo.

Il est ainsi possible directement depuis son application HTML :

- De lancer d'autres applications : Navigateur, Téléphone, Caméra, Google Map, Audio, Vidéo
...
- D'accéder en consultation ou mise à jour aux contacts, calendrier, messagerie, ...
- D'accéder à des informations système : Connectivité réseau, Localisation GPS, alimentation,
...

Il n'y a donc aucune limitation à accéder à des informations de la machine.

Interaction clavier ou souris avec l'application

Une application peut interagir avec l'utilisateur par l'intermédiaire des différents contrôles habituels : boutons, zones de saisie de texte, ... sur lequel les événements standards du JavaScript sont supportés.

Le framework Mojo offre également la capacité d'intercepter directement les événements claviers (touche appuyée puis relâchée) et les « clics » de l'utilisateur.

Exemple de fonctionnalités: Sytadin, Bouygues Epargne Salariale, Annuaire

C2S a réalisé à titre d'exemple l'application Sytadin pour WebOS. Elle permet de visualiser une carte et intègre les fonctionnalités standards pour la plateforme : zoom multitouch, défilement horizontal des cartes, mise à jour automatique, ...



Synthèse sur la plate-forme

En reposant sur le paradigme que le système d'exploitation n'est qu'un « super navigateur », WebOS propose une architecture qui permet le développement très rapide de nouvelles applications. Il parie également que HTML et JavaScript sont les technologies de choix pour le développement de Widgets. Ce sera sans nul doute un succès si l'ergonomie du terminal remporte l'adhésion des utilisateurs et attire ainsi les développeurs sur cette nouvelle plateforme.

Conclusion

Même si l'ergonomie des Widgets et leurs fonctionnalités peuvent être différentes d'une plate-forme à l'autre, l'étude des Widgets disponibles sur chaque plateforme montre qu'on retrouve le même type de Widgets sur les différentes plateformes. On constate également un véritable engouement actuel pour ce type d'application, plusieurs centaines (voire plusieurs milliers) de Widgets étant disponibles sur les plateformes les plus populaires.

L'étude technique réalisée montre une grande hétérogénéité des techniques de développement de Widgets que ce soit pour les interfaces graphiques (HTML, XML, par code), pour les traitements (JavaScript, ObjectiveC, Java, .NET) où plus généralement sur le modèle de développement de la plateforme. Deux des plateformes (Yahoo Mobile et Samsung Touch Wiz) essaient d'ailleurs d'adresser cette hétérogénéité en proposant l'exécution sur différentes plateformes.

Le tableau ci-dessous résume les caractéristiques des différentes plateformes :

	Vista	Yahoo	Yahoo Mobile	iPhone	Windows Mobile	Android	Samsung	WebOS
Ecran	HTML	XML	XML	Objective C	VB/C#	Java	HTML	HTML
Traitements	JavaScript	JavaScript	-	Objective C	C++/VB/C#	Java	JavaScript	JavaScript
API	Spécifique	Spécifique	-	Cocoa	.NET	JDK, Spécifique	Spécifique	Mojo
Configuration	Page spécifique	Interface dédiée	Interface dédiée	A développer	A développer	A développer	A développer	Page spécifique
URL distante	AJAX Images	AJAX Images	Appel Serveur	Tout type	Tout type	Tout type	AJAX Images	AJAX Images
Rafraîchissement	Timer JavaScript	Timer JavaScript	Non	Timer, Push	Timer	Timer, Thread, Services	Timer JavaScript	Timer JavaScript, Services
Accès local	Oui	Oui	Non	Oui	Oui	Oui	Non	Oui
Clavier/Souris	Evénmt JavaScript	Evénmt JavaScript	Simple	Evénmt système	Evénmt système	Evénmt système	Evénmt JavaScript	Evénmt JavaScript, système
Format déploiement	.gadget (fichier compressé)	.widget (généré)	.zip	binaire	.cab	.apk (archive compressé)	.wgt (fichier compressé)	.ipk (généré)
Plateforme déploiement	Windows Live, Direct	Yahoo, Direct	Yahoo	AppStore	AppStore, Direct	AppStore, Direct	AppStore, Direct	AppStore, Direct

Malgré cette hétérogénéité, le couple HTML et JavaScript reste le plus couramment rencontré. C'est donc naturellement que nous faisons le pari qu'il pourrait s'imposer dans l'avenir pour le développement de Widgets :

- Parce qu'il est un modèle simple à appréhender par des développeurs,
- Parce qu'il est cohérent avec les modes d'accès aux ressources web (HTTP, SOAP, REST, ...),
- Parce qu'il permet d'adresser à la fois les plateformes de Widgets embarqués et web,
- Parce qu'il commence à faire l'objet de normalisation par le W3C.

Références

Introducing Windows Vista Sidebar and developing a gadget

<http://channel9.msdn.com/posts/DanielMoth/Introducing-Windows-Vista-Sidebar-and-developing-a-gadget/>

Developing a Gadget for Windows Vista Sidebar: Part I

[http://msdn.microsoft.com/en-us/library/bb456468\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb456468(VS.85).aspx)

A virtual Earth Slide show Gadget for Windows Vista

<http://www.codeproject.com/KB/gadgets/Mobim.aspx>

Inside Vista SideBar

<http://www.codeproject.com/KB/gadgets/InsideBar.aspx>

iPhone Dev Center – Apple Developer Connection

<http://developer.apple.com/iphone/>

iPhone in action

<http://www.manning.com/callen/>

Android Developer

<http://developer.android.com>

Android Widget Design Guidelines

http://developer.android.com/guide/practices/ui_guidelines/widget_design.html

Yahoo! Widget Developers

<http://widgets.yahoo.com/tools/>

Yahoo! Mobile Developers Home

<http://mobile.yahoo.com/developers>

Creating Managed Today Screen Items

<http://www.christec.co.nz/blog/archives/279>

.NET Compact Framework Sample: Today Screen

<http://www.microsoft.com/downloads/details.aspx?FamilyID=5A37056B-A2B7-4968-89CB-5F7C3DA1EDAA&displaylang=en>

webOSdev

<http://developer.palm.com>

Palm webOS – The Insider’s Guide to Developing Applications in JavaScript with Mojo

<http://oreilly.com/catalog/9780596155254>

A propos de C2S



C2S est la société de services informatiques du Groupe Bouygues en matière de Systèmes et Technologies de l'Information :

- Spécialisée sur des domaines ciblés en avance de phase,
- disposant d'expertises reconnues, renforcées par des partenariats (éditeurs, Grandes Ecoles et Universités) et d'une fonction de Recherche & Développement dynamique,
- capable d'accompagner des métiers variés sur des projets innovants.

Depuis sa création, C2S a pour ambition d'anticiper et d'accompagner l'évolution des systèmes d'informations.

Son développement est influencé par une mission spécifique qui consiste à être au service du Groupe en se positionnant comme partenaire de confiance, sur des domaines ciblés, avec les objectifs suivants :

- Accompagner les différentes DSI du Groupe pour développer des solutions à valeur ajoutée, différenciantes et novatrices, en étant force de proposition pour l'implémentation des Technologies de l'Information et de la Communication,
- Contribuer à la capitalisation des connaissances et aux synergies au sein du Groupe par une approche mutualisée,
- Accompagner certaines sociétés du Groupe dans le développement de leurs offres commerciales (notamment Bouygues Telecom Entreprises).

C2S travaille pour tous les métiers du groupe ainsi que pour de grands groupes industriels et commerciaux à l'extérieur.

© 2009 C2S. Tous droits réservés.

Les informations contenues dans ce document représentent le point de vue actuel de C2S sur les sujets exposés à la date de publication. Tout extrait ou diffusion partielle est interdit sans l'autorisation de C2S.

Les noms de produits ou de sociétés dans ce document peuvent être les marques déposées de leurs propriétaires respectifs.