|  |
| --- |
| **Etat de l’art**  **les IHM pour l’embarqué** |

|  |  |  |
| --- | --- | --- |
| Type de document  **Etat de l’art** | référence  **EdA\_RADOME** | Version  **02** |

|  |  |  |
| --- | --- | --- |
| Rédacteur | Vérificateur | Approbateur |
| **Gautier Duval** | **Gautier Duval** | **Guillaume BONNEFOY** |

Historique des versions

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Auteur | Date | Objet de la version |
| 01 | Gautier DUVAL | 05-03-2014 | Version initiale |
| 02 | Gautier DUVAL | 03-11-2014 | Ajout sur websockets |
|  |  |  |  |

Contenu

[1. INTRODUCTION 6](#_Toc402776812)

[1.1. Objectifs du document 6](#_Toc402776813)

[1.2. Contexte 6](#_Toc402776814)

[2. HISTORIQUE 7](#_Toc402776815)

[2.1. Définition IHM 7](#_Toc402776816)

[2.2. Domaines d’applications 8](#_Toc402776817)

[2.2.1. les contraintes de conception des IHM embarquées 9](#_Toc402776818)

[2.2.2. Le type d’IHM embarquées existant 9](#_Toc402776819)

[2.2.3. LES TOOLKITS GRAPHIQUES 10](#_Toc402776820)

[3. Solutions d’affichage déporté 13](#_Toc402776821)

[3.1.1. MirrorLink 13](#_Toc402776822)

[3.1.2. CarPlay 13](#_Toc402776823)

[3.1.3. Citrix XenApp 14](#_Toc402776824)

[4. Constat 16](#_Toc402776825)

[5. HTML5 17](#_Toc402776826)

[5.1.1. Qu’est-ce que HTML5 17](#_Toc402776827)

[5.1.2. POSSIBILITES DU HTML5 17](#_Toc402776828)

[6. HTML5 POUR l’Embarqué ? 23](#_Toc402776829)

[6.1.1. Avantages du HTML5 24](#_Toc402776830)

[6.1.2. Incertitudes 24](#_Toc402776831)

[7. Acronymes, LEXIQUE et notions 25](#_Toc402776832)

[8. Sources 27](#_Toc402776833)

Liste des Figures

**Aucune entrée de table d'illustration n'a été trouvée.**

Liste des Tableaux

**Aucune entrée de table d'illustration n'a été trouvée.**

# INTRODUCTION

## Objectifs du document

Ce document a pour but de faire un état des lieux sur les technologies et/ou implémentations actuels en terme d IHM dans les systèmes embarqués, de donner une vue globale des produits existants. Ceci nous permettra de proposer un produit / concept innovant ou qui se démarque par une ou plusieurs spécificités par rapport au marché actuel.

## Contexte

Dans ce document nous allons essayer de faire un état de l’art sur les interfaces graphiques pour les systèmes embarqués de manière générale dans un premier temps, puis de ce que la technologie HTML5 et ses nouvelles fonctionnalités pourrait apporter au design et l’accessibilité des IHM pour l’embarqué.

Nous allons dresser un rapide présentation historique des premières IHMs, ensuite nous ferons un bilan des domaines d’utilisation et des tendances actuelles et enfin les possibilités d’évolution futur.

# Evolutions des IHMs

## Définition et historique

Les interactions homme-machine (IHM) définissent les moyens et [outils](http://fr.wikipedia.org/wiki/Outil) mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une [machine](http://fr.wikipedia.org/wiki/Machine). Les [ingénieurs](http://fr.wikipedia.org/wiki/Ing%C3%A9nieur) en ce domaine étudient la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes qui soient [ergonomiques](http://fr.wikipedia.org/wiki/Ergonomie), efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation.

L’Interface Homme Machine est un ensemble de dispositifs matériels et logiciels permettant à un utilisateur de communiquer avec un système interactif.

Les principales dates qui ont marqué l’histoire des IHM :

* 1963 : Invention du premier écran graphique.
* 1973 : En janvier 1973 est présenté le premier micro-ordinateur, le [Micral](http://fr.wikipedia.org/wiki/Micral) conçu par [François Gernelle](http://fr.wikipedia.org/wiki/Fran%C3%A7ois_Gernelle) de la société [R2E](http://fr.wikipedia.org/wiki/R2E) avec une mémoire RAM 8 Ko.
* 1980 : Premier ordinateur grand public commercialisé



* 1990 : le Windows 3.0 :



* Au fil des années la taille des ordinateurs s’est réduite et les capacités ont augmentés notamment avec la génération des MacBook de chez Apple.



## Domaines d’applications

On peut classer les IHM en deux parties :

* Une première partie destinée au grand public comme par exemple les Smartphones, tablettes…etc. Ce type d’IHM n’est pas soumis à des contraintes de fonctionnement très compliqués.
* La deuxième partie est destinée au monde de l’industrie principalement dans le domaine du Ferroviaire, Aéronautique, Médical et l’automobile. Dans ce chapitre nous allons nous attarder sur les contraintes de conception et de fonctionnement qui freinent le développement des IHM Web dans l’industrie.

### les contraintes de conception des IHM embarquées

Les principales contraintes auxquelles sont confrontées les conceptions d’IHM pour systèmes embarqués sont :

* La taille des écrans limitée est l’une des principales contraintes en embarqué,
* Les OS embarqués limités sont limités en ressource (mémoire, processeur, réseaux, énergie) et cela réduit les possibilités de développement des IHM embarqués,
* Les drivers spécifiques,
* Le développement doit être optimisé afin de répondre aux problèmes de temps de réponse (temps-réel ou contraintes temporelles fortes) et de mémoire,
* L’IHM doit être relativement simple à utiliser

### Le type d’IHM embarquées existant

Le terminal de navigation (GPS): Ce type de terminal est intégré principalement dans les voitures, les futures versions devront intégrer aussi la connexion Internet et télévision numérique en plus de la fonction GPS.



* Le terminal de paiement par carte bancaire.

Les écrans de signalisation dans les trains qui ne cessent d’évoluer.

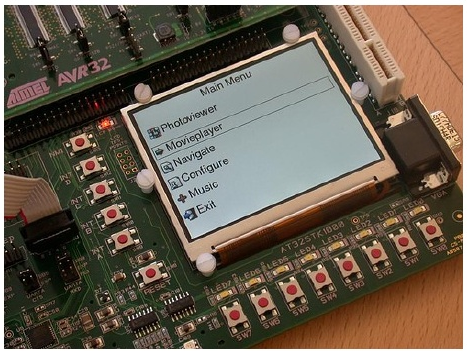
### LES TOOLKITS GRAPHIQUES

Afin d’afficher des objets graphiques d’interactions avec l’utilisateur (tels que des menus de navigation, des boutons d’actions, images, menu déroulant, fenêtres…) le programme s’appuie sur des Framework logiciels appelés bibliothèques graphiques (ou « toolkit ») . Une bibliothèque graphique est une [bibliothèque logicielle](http://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) spécialisée dans les fonctions [graphiques](http://fr.wikipedia.org/wiki/Graphique), qui sont classables en trois types (apparus dans cet ordre chronologique et de complexité croissante):

* Les bibliothèques de tracé d'éléments 2D : permettent de tracer les éléments graphiques de base que sont les lignes, les polygones et d'afficher des [pixels](http://fr.wikipedia.org/wiki/Pixel) ce qui permet d'afficher des [icônes](http://fr.wikipedia.org/wiki/Ic%C3%B4ne_(informatique)) et des [images](http://fr.wikipedia.org/wiki/Image_num%C3%A9rique).)
* Les bibliothèques d'interface utilisateur : ce sont les éléments graphiques qui permettent à l'utilisateur d'interagir avec le programme. elles sont la base de l'ergonomie des ordinateurs et permettent de construire une représentation graphique au programme avec des fenêtres, boutons, ascenseurs.
* Les bibliothèques 3D : apparues en dernier chronologiquement, les bibliothèques 3D permettent de faire de la [synthèse d'image 3D](http://fr.wikipedia.org/wiki/Synth%C3%A8se_d%27image_3D) c’est-à-dire de dessiner des éléments en volume.([OpenGL](http://fr.wikipedia.org/wiki/OpenGL), [DirectX](http://fr.wikipedia.org/wiki/DirectX)) Les bibliothèques 3D actuelles tirent parti, quand elles le peuvent, des capacités des [cartes accélératrices](http://fr.wikipedia.org/wiki/Carte_acc%C3%A9l%C3%A9ratrice).

Les solutions utilisées jusqu’ici de manière courante et classique pour réaliser des interfaces graphiques dans le monde de l’embarqué, sont basées sur des codes natifs (C, C++ …) et compilés venant s’appuyer sur des toolkits et bibliothèques graphiques diverses et variées. On peut trouver fréquemment parmi elles :

* Qt (et QML) 
* GTK+ 
* EFL 
* SDL 
* WxWidget C:\Users\EPR_Consultant07\Desktop\Logo_wxWidgets.png



# Solutions d’affichage déporté

### MirrorLink



Le Car Connectivity Consortium (CCC), composé de divers fabricants automobiles et d'électronique, a fait la démarche d’établir une norme de l'industrie pour les applications et les dispositifs de certification qui sont à la fois sûr et utile pour les conducteurs, appelé MirrorLink. L'effort conjoint par les constructeurs automobiles et les fabricants de téléphones vise à développer des normes ouvertes pour définir des opérations de smartphones liés aux voitures. Un élément unique de MirrorLink est sa capacité à montrer l'interface smartphone à partir d'une variété de plates-formes OS mobiles sur l'affichage tête-haute audio.



### CarPlay



CarPlay est un nouveau standard qu’Apple a introduit pour ses appareils iOS afin de travailler avec les fabricants de systèmes intégrés pour la voiture.

Le but de CarPlay est de fournir un accès direct aux fonctionnalités, contrôle et utilisation de l'appareil sous iOS directement par les fabricants de systèmes de contrôle dans la voiture d'origine.

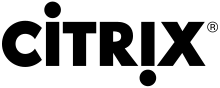
L'intégration vise plusieurs fonctions que les appareils iOS intègrent actuellement, il s'agit notamment de :

* Siri: Eyes Free mode : pour les opérations ne requérant pas de voir ou de toucher.
* La navigation par satellite (Satnav).
* Le contrôle et l’instruction téléphonique.
* Le contrôle de la musique.
* Le contrôle et la réponse aux messages ( iMessage)



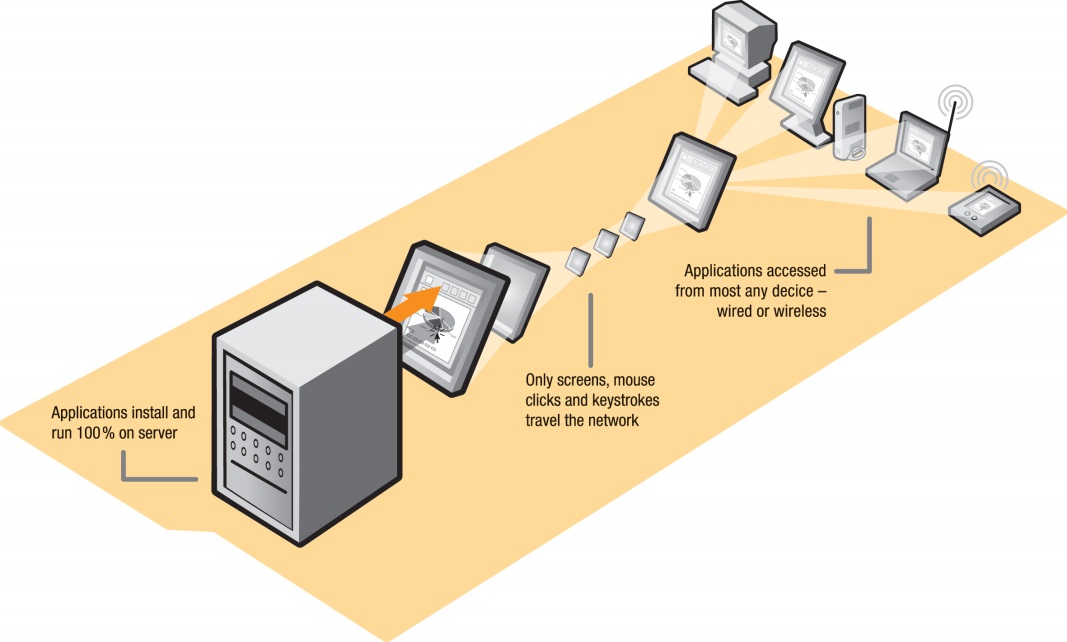
<http://www.lemonde.fr/technologies/article/2014/03/04/apple-lance-carplay-l-appli-qui-connecte-l-iphone-a-la-voiture_4377126_651865.html>

### Citrix XenApp



Citrix XenApp est un logiciel de la société Citrix systems permettant d'accéder à distance à des applications à partir de clients légers. Il s'agit d'un logiciel serveur permettant de distribuer des applications ou des services sur un réseau et d'y accéder (« solution d'infrastructure d'accès »).

Par exemple l'installation d'une application se fait sur le serveur et se lance sur celui-ci. L'application s'exécute donc sur le serveur, en utilisant les ressources (CPU, mémoire, espace disque) du serveur. L'affichage et les commandes de clavier/souris sont transmises au poste de travail de l'usager via un réseau local. Cela a pour but de réduire le travail du poste client, permettant ainsi d'utiliser un poste de capacité réduite pour exécuter des applications nécessitant beaucoup de ressources.



Pour lancer une application, on se connecte à un portail web contenant les applications auxquelles on a accès. On peut aussi installer sur le poste client un agent qui servira à afficher les applications disponibles à l'utilisateur.

Le logiciel client initie la connexion au serveur en spécifiant l'application désirée. L'affichage ainsi que les commandes clavier/souris sont transmises via le réseau local en utilisant un protocole appelé ICA. Celui-ci permet un transfert rapide des données grâce à une méthode de compression développée par Citrix.

Ce système permet donc de rendre disponible des applications diverses sans avoir à les installer dans chacun des postes clients qui le nécessitent.

# Constat

Les toolkits graphiques et produits présentes précédemment ont leur avantage, mais peuvent présenter des inconvénients

* recompiler le code à chaque modifications dans le design de l IHM
* dépendance directe à la plate-forme/matériel sur lequel doit tourner l IHM

Aussi, face à la multiplication des diverses plateformes de visionnage, si on veut un affichage déporté sur un support annexe (tablette, smartphone..) cela implique de développer à nouveau un autre code pour cette plateforme mobile (langage Java pour les systèmes basés sur l’OS Android, langage objective C pour les systèmes basés sur l’OS Apple…)

N’y aurait-il pas une solution qui s affranchirait de cette problématique ? Une technologie commune pour générer ces IHMs ?

# HTML5

### Qu’est-ce que HTML5

Le [W3C](http://www.w3.org/)(World Wide Web Consortium), l'organisme chargé de réguler les standards du Web, a publié fin décembre 2012  la définition complète de la spécification HTML5. La version actuelle de ce format de représentation des pages Web est donc désormais considérée comme stable, la publication officielle étant prévue en 2014 après les phases consacrées à l’interopérabilité et aux tests.

HTML5 est un langage qui regroupe en fait un ensemble de technologies (SVG, CSS4, WebGL, File API, MathML, etc.). Avalisé par Adobe, Apple, Google et Microsoft, HTML5 doit favoriser le développement de services dits « rich media » accessibles à partir de nombreuses plates-formes (téléphones mobiles, tablettes, PC, télévisions, équipements d’info-divertissement automobiles, panneaux de signalisation numériques, systèmes embarqués, etc.).

 Le futur standard est également censé faciliter le développement d’interfaces utilisateur plus riches à base uniquement de technologies web et donc décodables telles quelles par le navigateur, sans avoir à télécharger de programmes ni utiliser d'applications natives. HTML5 offre par ailleurs de nouvelles capacités au navigateur, notamment celle de communiquer avec des éléments matériels du terminal (processeur, webcam, batteries...) au travers d'API parfaitement documentées.

### POSSIBILITES DU HTML5

HTML5 est une évolution de HTML 4.01, datant de 1999. Le but principal de cette nouvelle version était de réduire le nombre de plugins nécessaires en plus de la page web (notamment pour les éléments multimédia) mais aussi le volume de script présent dans chaque page. Un des axes de développement était également la nécessité du cross-platform (tablettes, téléphones …) pour pouvoir s’adapter aisément à des supports différents.

Cette réflexion a abouti à la création de HTML 5 en 2006 (travail en cours, release stable attendue pour fin 2014).

Le nouveau standard HTML propose notamment :

* De nouvelles balises qui viennent étoffer l’ensemble des éléments existants
* Un système de dessin (canvas et vectoriel)
* Une intégration plus intuitive des éléments multimédia
* De nouveaux mécanismes facilitant le dynamisme et l’interactivité des pages

#### Extension des éléments existants

Eléments sémantiques – non-sémantiques

Un élément non-sémantique est un élément de mise en forme qui ne renseigne pas sur son contenu (ex *<div>*, *<span>* …)

Pour éviter l’emploi de ces éléments peu parlant au niveau fonctionnel, HTML5 introduit une batterie de balises de mise en forme, dont le rôle est clairement défini selon la zone d’utilisation dans la page HTML traditionnelle :



* *<header>*
* *<nav>*
* *<section>*
* *<article>*
* *<footer>*
* …

Ajout d’éléments

De nouvelles balises ont été créées pour répondre à de nouveaux besoins :

* *<canvas>* pour créer un dessin dans un espace de pixels donné
* *<video>* et *<audio>* pour inclure un lecteur vidéo/audio dans la page

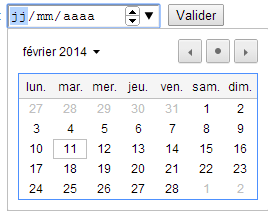
Ces balises ont été ajoutées pour supplanter l’utilisation de plugins (principalement Flash) de lecture de vidéo et de dessin.

De plus, de nouvelles balises ont été créées pour étoffer les formulaires (forms) :

* *<datalist>*, pour renseigner un ensemble d’options pré-remplies dans un champ texte
* *<keygen>*, pour générer une clé unique
* *<output>*, pour représenter un calcul de données numériques du formulaire

Extension fonctionnelle des éléments existants

De nombreux types ont été rajoutés dans la balise <input> (nouvelles valeurs possibles de l’attribut « input ») :



* color
* date
* datetime
* …

Ces nouvelles valeurs d’attribut permettent de créer avec HTML de façon native un ensemble de gadgets qui, précédemment étaient créés de façon très hétéroclite avec un ajout conséquent de HTML et de JavaScript.

* De plus, de nouveaux attributs ont été rajoutés dans la balise *<input>* et *<form>,* principalement pour ajouter des limitations de saisie dans les valeurs (min, max) et plus généralement, des mécanismes de validation (regexp, novalidate) et d’aide à la saisie (autocomplete). Ces mécanismes étaient auparavant créés avec des ajouts, encore une fois conséquente, de HTML+ JavaScript.

Nouvelles possibilités graphiques : canvas et SVG

#### Le canvas HTML5

PRINCIPE DU CANVAS HTML

L’élément <canvas> est un container permettant de dessiner des éléments graphiques avec du JavaScript. Un ensemble de méthodes de base existe pour dessiner des lignes, rectangles, cercles et textes.

Il est également possible d’afficher des images dans le canvas.

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the HTML5 canvas tag.

</canvas>

<script>

var c=document.getElementById("myCanvas");

var ctx=c.getContext("2d");

ctx.fillStyle="#FF0000";

ctx.fillRect(0,0,150,75);

</script>

</body>

</html>



**CANVAS ET 3D**

Grâce à une API graphique libre de droit basée sur le standard OpenGL ES 2.0, WebGL, il est possible de tracer une scène 3D dans un canvas HTML5.

<http://www.khronos.org/webgl/>



#### Le SVG dans du HTML5

PRINCIPE DU SVG

Le SVG (Scalable Vector Graphics) est un format de données conçu pour décrire un ensemble de graphiques vectoriels et basés sur XML.

Sur le principe, on dispose des mêmes primitives de dessin que le canvas (lignes, textes, cercles …).

<!DOCTYPE html>

<html>

<body>

<svg width="300" height="200">

<rect id="myRect" height="100px" width="100px" fill="blue"/>

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>



Le format vectoriel ajoute cependant l’avantage de l’absence de dégradation de l’élément graphique après zoom, ce qui peut justifier son utilisation dans certains cas.

CAS D’UTILISATION :

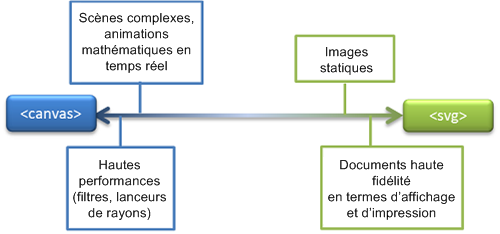
* Zoom et dezoom sur une grande image
* Cartographie

<http://ie.microsoft.com/testdrive/Graphics/RealWorldDataAndDiagrams/PeriodicChart.xhtml>

<http://www.openstreetmap.org/>

Canvas versus SVG

|  |  |
| --- | --- |
| Canvas | SVG |
| Basé sur pixel (.png dynamique) | Basé sur forme |
| Un seul élément HTML | Plusieurs éléments graphiques, qui deviennent des parties intégrantes du DOM |
| Modifié par script uniquement | Modifié par script et CSS |
| L’interaction modèle/utilisateur de l’événement est granulaire (x,y) | L’interaction modèle/utilisateur de l’événement est abstraite (rect, chemin) |
| La performance est meilleure sur une surface plus petite, un plus grand nombre d’objets (>10k), ou les deux | La performance est meilleure sur un plus petit nombre d’objets (<10k), une surface plus grande, ou les deux |



Par exemple, on préférera l’utilisation du canvas pour des jeux utilisant dynamiquement un grand nombre d’objets graphiques par rapport au SVG.

**Le multimédia**

HTML5 permet de disposer d’un lecteur audio et vidéo intégré, et donc d’éviter le recours aux plugins pour les éléments multimédia.

<!DOCTYPE html>

<html>

<body>

<video width="320" height="240" controls>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

</body>

</html>



Les éléments audio et vidéo sont des ressources de la DOM et possèdent donc des attributs qui peuvent être modifiés (largeur/hauteur) et des méthodes qui peuvent être appelées (lecture/pause).

#### Autres évolutions et apports

* Géolocalisation

HMTL5 offre une API de géolocalisation qui permet de récupérer la position de l’utilisateur (si celui-ci accepte au travers de son navigateur) et de l’exploiter.

L’API propose également un système évènementiel pour rester averti des changements de position de l’utilisateur (ex : implémentation d’une application de guidage).

* Le concept de Web Storage

HTML5 propose une alternative au système de cookies pour stocker des données utilisateur : le web storage. Il s’agit d’un ensemble de paires clé/valeur stockés par le navigateur en local, et accessibles au travers de JavaScript.

Par rapport au système de cookies, il n’y a pas d’échange avec le serveur (pas de transaction pour obtenir la valeur des données), et il existe moins de limite de taille (quelques Mo contre quelques Ko pour les cookies). Par contre, les données sont accessibles facilement par l’utilisateur, donc ce système n’est pas valable pour les données sensibles.

* Les Web Workers

Le Web Worker est un système qui permet de déclencher l’utilisation d’un script JavaScript dans un thread séparé du thread d’exécution principal du navigateur.

Ce système peut être extrêmement utile pour paralléliser les traitements d’une application graphique évoluée dans une page HTML5.

* Les WebSockets

Les Websockets sont une autre des nouveautés implémentées dans les navigateurs compatibles HTML5.

Le Websocket proposera à terme une implémentation native et unifiée dans les navigateurs et serveurs web d'un canal bidirectionnel permettant :

* la notification au client d'un changement d'état du serveur
* l'envoi de données en mode « pousser » (méthode Push) du serveur vers le client (sans que ce dernier ait à effectuer une requête)

Cf le document annexe

# HTML5 POUR l’Embarqué ?

Il y a quelques règles communes dans la conception d'interfaces utilisateur (UI) dans les systèmes embarqués, telles que « utiliser les boîtes de dialogue et les touches familières aux utilisateurs »  ou « garder les choses simples en limitant l'activité des utilisateurs autant que possible, dans le contexte de la fonction en cours ». La conception de ces boutons, boîtes de dialogue, écrans d'aide et d'autres widgets n’est pas toujours facile à réaliser dans les systèmes embarqués dont les ressources sont limitées. Ce qui est nécessaire, c'est un moyen simple et efficace de créer ces widgets, et le HTML5 pourrait aider en cela.

HTML5 devient de plus en plus populaire, et peut ajouter des fonctionnalités et des capacités à votre appareil, étendre le cycle de vie d'un produit actuel et préparer des applications pour de futures plateformes. Aujourd'hui l’idée de construire une IHM basée sur HTML5 + Websockets est devenue pratique. Les outils d'interface graphique disponibles via HTML5 + CSS3 + Javascript commencent être incroyables.

HTML5 est la plate-forme agnostique (l’objectif principal est l'indépendance de plateforme). La plupart / toutes le gros IU seraient déchargées au navigateur du coté client et n’utiliserait donc pas les ressources de l’appareil embarqué. En utilisant des Websockets, la communication entre le dispositif embarqué et le navigateur serait en fait plus proche d’une architecture « peer-to –peer » que client-serveur.

HTML5 et ses technologies connexes (CSS3, JavaScript, AJAX, JSON, etc.) offrent une excellente solution, non - propriétaire pour la construction d’IHM riches, et indépendantes du matériel. Le statut de recommandation du W3C pour le HTML5 est prévu en 2014. Beaucoup de fonctionnalités HTML5 sont déjà en place dans les navigateurs Web d'aujourd'hui (à la fois sur les appareils de bureau et les appareils mobiles).

### HTML5 pour l’info-divertissement automobile

HTML5 pourrait être la technologie de choix pour les IHM dans les véhicules, et en particulier dans la conception de systèmes embarqués d'info-divertissement, l'avènement du téléphone intelligent ayant tout changé. HTML5 est un standard largement adoptée qui est déjà fait ses preuves dans un large éventail de mises en œuvre à travers une variété d’industries, y compris l’automobile. Si quelque chose peut aider les constructeurs à offrir ce que leurs clients veulent dans leurs systèmes d'info-divertissement en voiture à un coût raisonnable aujourd'hui et de demain, c'est probablement l’HTML5.

L’HTML5 est en passe de devenir non seulement une technologie d’IHM populaire, mais l'environnement préféré pour délivrer riches interfaces utilisateur flexibles. L’HTML n'est plus seulement la norme pour la présentation de contenu web, mais une technologie viable pour IHM pour toutes sortes d’applications.

### QNX CAR

[QNX Software Systems](http://embedded-computing.com/topics/qnx-software-systems) développe un middleware hautes performances pour les interfaces utilisateur embarqué, le QNX® Aviage® HMI Suite 2.0. La suite offre aux développeurs embarqués un cadre complet pour le développement d’interface utilisateur qui combine les riches graphiques d'Adobe Flash Lite avec la fiabilité et la performance de la QNX Neutrino®RTOS (système temps-réel)

La suite QNX Aviage HMI 2.0 permet aux développeurs de mettre en œuvre des interfaces utilisateur entières dans Adobe Flash, ce qui élimine le long processus de création d IHM avec API graphiques ou des outils (toolkits) encombrants et lourd en codage. En conséquence, les développeurs peuvent créer des IHM dynamiques et conviviales pour les unités voiture d'info-divertissement, de systèmes d’automatisation industrielle, d’appareils médicaux, et d'autres systèmes embarqués en beaucoup moins de temps . La suite est également un élément clé de QNX CAR, un programme novateur qui aide les fournisseurs de l'électronique automobile à accélérer la réalisation de prototypes et de réduire les coûts d’ingénierie.



### TESLA Model S

La Tesla Model S est le deuxième véhicule produit par Tesla Motors. Entièrement électrique, il a la particularité d'embarquer une planche de bord intégralement numérique, sous la forme d'un écran tactile central de 17 pouces accompagné d'un autre écran en lieu et place des habituels compteurs de vitesses et compte-tours.



Le constructeur exploite la taille de son écran, qui peut se scinder en deux pour afficher un maximum d'informations, mais l'utilisateur peut choisir de faire basculer chacune de ces fenêtres en plein écran. Malgré le nombre impressionnant de fonctionnalités auxquelles elle donne accès (du contrôle de la climatisation à la fermeture centralisée en passant par le réglage de l'amortissement), l'interface reste simple, grâce aux différents onglets situés en haut de l'écran tactile. La connectivité est assurée par une carte SIM compatible 3G

.

### Avantages du HTML5

* Indépendance de la plateforme
* Accès multiplateforme (sur PC, tablette, Smartphones …) des lors que l’on dispose d’un navigateur HTML pour un code a priori unique
* HTML5 est un standard non-propriétaire
* Design potentiellement riche
* Support natif de vidéo, audio,…
* CSS permet entre autre d’adapter à chaque marque/constructeur un code html générique et l’approche Responsive Design (adapter l’affichage selon le support de visualisation)

### Incertitudes, Verrous technologiques

* Aspect performance (capacité du hardware pour gérer les affichages HTML avec un moteur de rendu)
* Pas vraiment de temps réel au sens « dur » (pas de déterminisme au niveau du délai temporel etc.)
* Gestion de la sécurité à garder à l’esprit
* Possibilité / compatibilité du multithreading et des Websockets

# Acronymes, LEXIQUE et notions

* **AJAX**

L'architecture informatique Ajax (acronyme d'Asynchronous JavaScript and XML) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.

Ajax combine JavaScript, les CSS, XML, le DOM et le XMLHttpRequest

* **CSS**

Les feuilles de style en cascade1, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.

* **HTML**

L’Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C’est un langage de balisage permettant d’écrire de l’hypertexte, d’où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques.

* **IHM**

Interface Homme Machine

* **JS**

JavaScript (souvent abrégé JS) est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur.

* **JSON**

JSON (JavaScript Object Notation) est un format de données textuelles, générique. Il permet de représenter de l’information structurée comme le permet XML par exemple.

* **OS**

En informatique, un système d'exploitation (souvent appelé OS pour Operating System, le terme anglophone) est un ensemble de programmes qui dirige l'utilisation des capacités d'un ordinateur par des logiciels applicatifs. Il reçoit de la part des logiciels applicatifs des demandes d'utilisation des capacités de l'ordinateur — capacité de stockage des mémoires et des disques durs, capacité de calcul du processeur. Le système d'exploitation accepte ou refuse de telles demandes, puis réserve les ressources en question pour éviter que leur utilisation n'interfère avec d'autres demandes provenant d'autres logiciels.

Ex : Mac OS X, Linux, iOS , Android, Windows

* **SVG**

Le Scalable Vector Graphics (en français « graphique vectoriel adaptable »), ou SVG, est un [format de données](http://fr.wikipedia.org/wiki/Format_de_donn%C3%A9es) conçu pour décrire des ensembles de [graphiques vectoriels](http://fr.wikipedia.org/wiki/Image_vectorielle) et basé sur [XML](http://fr.wikipedia.org/wiki/Extensible_Markup_Language). Ce format inspiré directement du [VML](http://fr.wikipedia.org/wiki/Vector_Markup_Language) et du [PGML](http://fr.wikipedia.org/wiki/PGML) est spécifié par le [World Wide Web Consortium](http://fr.wikipedia.org/wiki/World_Wide_Web_Consortium).

* **Thread**

Un thread ou fil (d'exécution) ou tâche. Autres appellations: processus léger, unité de traitement, unité d'exécution, fil d'instruction, processus allégé) est similaire à un [processus](http://fr.wikipedia.org/wiki/Processus_(informatique)) car tous deux représentent l'exécution d'un ensemble d'instructions du [langage machine](http://fr.wikipedia.org/wiki/Langage_machine) d'un [processeur](http://fr.wikipedia.org/wiki/Processeur). Du point de vue de l'utilisateur, ces exécutions semblent se dérouler en [parallèle](http://fr.wikipedia.org/wiki/Parall%C3%A9lisme_(informatique)). Toutefois, là où chaque processus possède sa propre [mémoire virtuelle](http://fr.wikipedia.org/wiki/M%C3%A9moire_virtuelle), les threads d'un même processus se partagent sa mémoire virtuelle. Par contre, tous les threads possèdent leur propre [pile d’appel](http://fr.wikipedia.org/wiki/Pile_d%27ex%C3%A9cution).

* **TOOLKIT**

Toolkit est un mot anglais qui est utilisé en [informatique](http://fr.wikipedia.org/wiki/Informatique) et le plus souvent dans le contexte des [interfaces graphiques](http://fr.wikipedia.org/wiki/Interface_graphique). Ce mot, qui signifie littéralement boîte à outils désigne ici un « [widget toolkit](http://fr.wikipedia.org/wiki/Widget_toolkit) » (une [bibliothèque logicielle](http://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) pour la conception d'interface graphique)

* **(G)UI**

(Grapical) User Interface (interface utilisateur graphique). Similaire à IHM. Une interface graphique est un dispositif de [dialogue homme-machine](http://fr.wikipedia.org/wiki/Interface_homme-machine), dans lequel les objets à manipuler sont dessinés sous forme de [pictogrammes](http://fr.wikipedia.org/wiki/Pictogramme) à l'[écran](http://fr.wikipedia.org/wiki/%C3%89cran_(informatique)), que l'usager peut utiliser en imitant la manipulation physique de ces objets avec un [dispositif de pointage](http://fr.wikipedia.org/wiki/Dispositif_de_pointage), le plus souvent une [souris](http://fr.wikipedia.org/wiki/Souris_(informatique)).

* **XML**

L'Extensible Markup Language (« langage de balisage extensible » en français) est un langage informatique de balisage générique. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG… Elle est reconnaissable par son usage des chevrons (< >) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche…) entre systèmes d'informations hétérogènes (interopérabilité)

# Sources

* <http://www.w3schools.com/html/html5_intro.asp>
* <http://en.wikipedia.org/wiki/HTML5>
* <http://msdn.microsoft.com/fr-fr/library/ie/gg193983(v=vs.85).aspx>
* <http://electronicdesign.com/embedded/html5-will-be-technology-choice-vehicle-hmis>
* <http://www.automotive-eetimes.com/en/bridging-the-chasm-between-html5-and-the-hardware-with-pps-messaging.html?cmp_id=71&news_id=222902243>
* <http://www.epanorama.net/newepa/2012/10/01/html5-for-hmi/>
* <http://electronicdesign.com/embedded/html5-will-be-technology-choice-vehicle-hmis>
* <http://qnxauto.blogspot.fi/2012/07/so-why-all-fuss-over-html5.html>
* <http://qnxauto.blogspot.fr/2012/02/everything-you-wanted-to-know-about.html>
* <http://www.qnx.com/products/qnxcar/index.html>
* <http://www.fujitsu-ten.com/business/technicaljournal/pdf/39note1.pdf>