

# Rapport Adaptation des IHM

Auteur : Adrien Casanova

## Etude bibliographique

La dernière décennie a été le témoin de la révolution mobile. Cependant, une autre révolution tout aussi importante, mais plus discrète, se met progressivement en place, celle des affichages publics. Les récentes avancées technologiques ont rendu les grands écrans disponibles pour le marché de masse à des prix abordables, leur permettant de s'intégrer de plus en plus aux espaces publics, intérieurs ou extérieurs. Cependant, cette distribution à grande échelle se fait sans grande attention portée à la manière d'interagir avec les utilisateurs. Il en résulte une réduction importante de l'efficacité de ces affichages conduisant même à un phénomène d'ignorance dans la plupart des cas.

Ainsi une réflexion importante sur le type d'informations affichées et la manière de les présenter doit aujourd'hui être engagée. Et comme pour l'adaptation mobile, des questions se posent : comment concevoir des interfaces adaptées aux affichages publics et à leur environnement ? Quelles sont les contraintes en présence ? Les enjeux ?

Alessandro Bendinelli et Fabio Paternò, de l'ISTI<sup>1</sup> de Pise, se sont intéressés à cette problématique et proposent un guide pour concevoir des interfaces adaptées aux affichages publics [1]. Mais avant de détailler le modèle qu'ils proposent ainsi que des applications concrètes, il convient de définir les buts, enjeux et contraintes liés aux affichages publics.

## 1. Affichages publics

Les affichages numériques envahissent nos espaces publics, intérieurs et extérieurs. Majoritairement représentés par de grands écrans, ils peuvent prendre différentes formes - vitrines, murs, miroirs, posters, etc. – et sont déjà présents dans de nombreux endroits tels que les gares, aéroports, musées, universités, centres commerciaux, restaurants, etc.

Ces affichages ont pour but principal d'attirer les passants et de saisir leur attention en leur fournissant des informations de types très divers : sollicitations commerciales, culturelles, de déplacements, promotions, publicités, informations pratiques, annonces, événements, ... ; Ainsi, ils sont principalement destinés à afficher des données sans offrir d'interaction directe.

L'ensemble des utilisateurs ciblés est aussi hétérogène que peut l'être une population dans un lieu donné. Les personnes fréquentant un lieu public peuvent être de tout âge, sexe, nationalité, confession et sont majoritairement non spécialistes. Ainsi ces affichages doivent être accessibles à tous.

---

<sup>1</sup> Istituto di Scienza e Tecnologie dell'Informazione

D'autres contraintes, environnementales, s'ajoutent. Celles-ci, liées au contexte du lieu, regroupent des facteurs tels que le bruit environnant – impactant directement les contenus multimédia et la concentration -, la luminosité - pouvant empêcher la lecture -, l'heure ou encore la gêne liée au fait d'avoir les mains prises - par des sacs ou valises par exemple-.

Un bon moyen de mettre en avant les défis associés à la conception d'interfaces pour des affichages publics et de comparer leur utilisation avec la lecture d'informations sur un site Internet grâce à un dispositif « traditionnel » (type ordinateur, smartphone ou tablette).

Lorsqu'un utilisateur parcourt un site web, il est généralement seul et a le temps de lire, parcourir ou chercher de l'information. De plus, il se trouve la plupart du temps dans une position assise, ou en tout cas statique, qui lui offre une plus grande concentration.

La lecture d'informations dans un espace public est tout autre. Les utilisateurs, le plus souvent mobiles, ne disposent que de très peu de temps pour observer les données qui leur sont présentées. De plus, contrairement au cas précédent, le contenu proposé est prédéfini et non changeable ou sélectionnable. Par conséquent une partie des passants n'y prêtera aucune attention.

Ainsi la lecture d'un site internet n'est pas adaptée à un écran large. Les informations sont difficilement lisibles compte tenu de la taille de la police et le contenu ne couvrant pas tout l'écran, il fait apparaître des zones vides et non utilisées. Une solution serait de s'inspirer du Responsive Web Design, actuellement adapté principalement au portage de sites web sur dispositifs mobiles. Cependant, bien qu'une telle solution permettrait de résoudre des problèmes liés au layout<sup>2</sup>, elle ne pourrait prendre au compte d'autres paramètres tout aussi importants.

Il apparaît alors que ces interfaces doivent être traitées très spécifiquement et que l'effort d'adaptation doit être fait à la conception et non à l'exécution. C'est dans cet optique que les chercheurs de l'ISTI ont proposé le guide suivant.

## 2. Guide pratique

En se basant sur des discussions avec des utilisateurs cibles, des experts du domaine, ainsi que sur des études de divers travaux de recherche, l'équipe de l'ISTI a défini neuf aspects, regroupés en trois catégories, à prendre en compte dans la conception d'interfaces pour affichages publics, ainsi que des guides à suivre pour chacun d'entre eux.

### Contexte

#### La position/localisation

L'emplacement a un impact direct sur le type d'informations et la manière de les présenter. Certains lieux sont adaptés à la lecture d'informations longues alors que d'autres sont des lieux de transit, imposant une attention limitée de la part des passants.

---

<sup>2</sup> Mise en page, disposition

En intérieur, les positions plus fréquentes sont les entrées, salles d'attente, couloirs, vitrines, ou encore comptoirs (de points d'information par exemple).

Ainsi, pour optimiser son efficacité, un dispositif d'affichage public doit être accessible, visible, placé dans un endroit fortement fréquenté et à hauteur des yeux ou au-dessus.

## L'heure

Le contenu peut être différent selon l'heure. Dans un endroit où se déroulent beaucoup d'événements, il peut être amené à changer et à se réadapter régulièrement. Les informations affichées doivent donc s'adapter à l'heure afin d'optimiser leur utilité pour les heures qui suivent.

Prenons l'exemple d'un écran proposant des informations météorologiques. Le matin, il sera utile de connaître le temps qu'il fera sur toute la journée, alors qu'à partir du midi il ne sera intéressant que de connaître la météo du reste de la journée et le soir celle du lendemain.

## Contenu

### Le type d'informations

Choisir le type d'informations affichées et les média utilisés en fonction de l'environnement, des utilisateurs ciblés et du but de l'affichage est très important. Le contenu peut être textuel ou multimédia – image, vidéo, cartes interactives – mais doit rester simple.

La source des différentes données a également son importance. Ainsi certaines parties du contenu peuvent être statiques alors que d'autres seront dynamiques et renouvelées régulièrement. Cependant la crédibilité de ces sources doit être étudiée et adaptée au domaine d'application. Ainsi, l'utilisation des réseaux sociaux sera filtrée, réduite ou même proscrite dans des domaines tels que la santé ou l'administration publique.

Enfin, des QR Codes peuvent être utilisés pour fournir des informations supplémentaires directement sur le smartphone d'un utilisateur.

### Le nombre d'éléments d'informations

Le nombre de sujets traités et de zones différentes affichées doit être choisi pour que l'écran soit utilisé dans son intégralité sans être trop encombré. Il dépend alors du but et de l'emplacement, celui-ci ayant un impact sur le temps consacré à la lecture. En général, un écran doit comporter entre 3 et 6 zones distinctes.

### Le texte

Le texte affiché doit utiliser des expressions courtes et claires, ne comporter que quelques lignes et être aligné à gauche ou au centre. L'usage de listes est également conseillé dans certains cas. Afin de rester lisible, attractif et simple, le message principal doit utiliser peu de mots (2/3 à 5/6), un nombre limité de lignes (1/2 à 6/7).

Les autres éléments textuels quant à eux doivent utiliser 30 mots maximum et favoriser l'usage de verbe à la forme active et de mots-clés. Les titres doivent comporter 22 caractères au maximum.

Tout texte plus long ne doit avoir pour but que de fournir des informations supplémentaires et non primordiales.

## Présentation

### La disposition/layout

La disposition considère les dimensions de l'écran afin de les exploiter au mieux pour optimiser le nombre de zones, leur position, leur taille, etc. Elle doit être pensée dans le but de capturer l'attention du lecteur et d'être assez simple d'interprétation pour guider sa lecture. En général, les vues sont construites autour d'une zone centrale principale, censée attirer l'attention, entourée de 2 à 4 zones de plus faible importance.

### Les couleurs

La distance entre un utilisateur et un affichage public est généralement comprise entre 1 et 3m (comparée à 50cm pour un ordinateur ou une tablette), les couleurs choisies doivent donc permettre de faciliter l'interprétation visuelle, être adaptée au contenu et être visibles dans l'environnement. On comptera  $7 \pm 2$  couleurs. Néanmoins, moins il y a de couleurs, plus simple est la lecture.

### La police

La police d'écriture doit être simple et lisible. Concernant la taille, le ratio entre titre et texte conseillé est de 1:1,5 à 1:2.

### La « dynamicité »

Cet aspect concerne le changement de l'affichage au cours du temps et des transitions entre chaque changement. Il est directement lié à l'attention de l'utilisateur, décomposée en plusieurs phases :

- L'attention complète dure entre 2 et 3 secondes
- La décision de lire attentivement le contenu ou de détourner le regard est prise en 10 à 15 secondes
- Le temps global passé à observer l'écran est compris entre 3 et 7 minutes et dépend du lieu. Par exemple, il sera compris entre 2 et 3 minutes dans une entrée et entre 7 et 8 minutes dans une salle d'attente.

Dans le cas d'une présentation cyclique de vues différentes, chacune d'entre elle doit rester visible pendant 3 à 5 minutes.

C'est donc un véritable modèle autour des interfaces d'affichages publics que proposent ici A. Bendinelli et F. Paternò.

## 3. Modèle

La figure suivante présente une modélisation du guide détaillé précédemment sous forme de diagramme de classes UML [2].

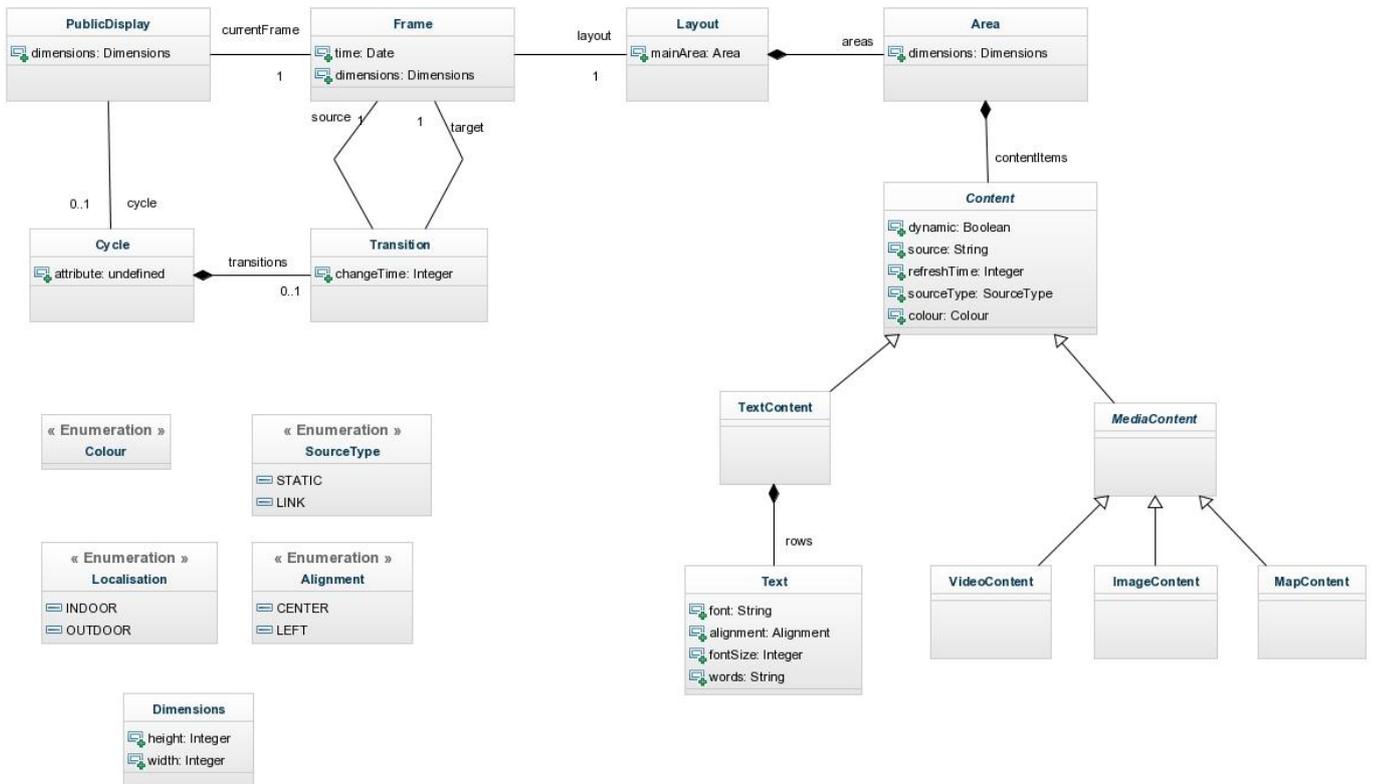


Figure 1 - Modèle

Ce modèle peut être retrouvé ici : [Modèle Public Displays](#)

Afin de le compléter, certaines contraintes doivent être ajoutées. Le langage UML ne permettant pas de les définir, elles sont détaillées ici textuellement :

- L'attribut *dimensions* d'un objet *Frame* est égal à l'attribut *dimensions* de son conteneur *PublicDisplay*
- La multiplicité de l'attribut *areas* de la classe *Layout* est **3..6**
- La multiplicité de l'attribut *rows* de la classe *TextContent* est **1..7**
- La multiplicité de l'attribut *contentItems* de la classe *Area* est **1..3**
- L'attribut *words* de la classe *Text* doit contenir entre **2 et 6** mots
- L'attribut *changeTime* de la classe *Transition* est compris entre **3 et 5** (exprimé en minutes)
- Si l'attribut *cycle* est non nul, toute *Frame* est à la fois *source* et *target* de deux objets *Transition* différents

## 4. Exemple d'application



Cet affichage public est donc conforme au modèle présenté et peut donc être supposé efficace. Il gagnerait cependant à être placé plus bas pour offrir une meilleure visibilité.

## 5. Avantages et inconvénients

Avec l'augmentation importante du nombre d'écrans et autres affichages dans les espaces publics, un tel guide est indispensable. Actuellement, les interfaces sur ces dispositifs sont majoritairement simplistes, ne proposant souvent qu'un affichage vertical d'informations simples. Suivre les pratiques définies ici permet d'optimiser l'efficacité et l'intégration de ces installations en proposant des données pertinentes et adaptées à l'environnement.

Cependant les aspects présentés sont fortement liés à cet environnement. Ainsi chaque interface sera unique et demandera un réel travail d'adaptation à la conception, ce qui peut être lourd à mettre en place. Bien sûr des schémas génériques peuvent être définis mais l'effort d'adaptation sera toujours important en raison du grand nombre de facteurs variables impliqués.

Ce développement récent autour des dispositifs d'affichages dans les espaces publics s'inscrit dans l'informatique ubiquitaire, troisième grande ère de l'informatique. En se projetant légèrement, on peut imaginer une interaction directe entre un passant et le dispositif, via son smartphone par exemple. Ainsi les informations affichées pourraient être générées à partir des données provenant de l'utilisateur afin de s'adapter au mieux à lui. Bien sûr des questions autour de la récolte et l'usage de données ainsi que leur impact sur la vie privée se posent de plus en plus et ce cas, ayant une dimension publique évidente, ne fait pas exception.

## Comparaison technologique

Le *Responsive Web Design (RWD)* est un paradigme de conception des sites web permettant d'adapter l'affichage au dispositif afin d'en optimiser le confort visuel, la lecture ou encore la navigation. De nombreuses solutions sont aujourd'hui disponibles sous forme de *framework*<sup>3</sup>. La partie suivante présente la comparaison de deux d'entre eux, *Bootstrap*[4] et *Gumby*[5].

### Bootstrap vs. Gumby

#### Bootstrap

Lancé en Août 2011 sous licence *Open Source MIT* [6] par Twitter, Bootstrap se définit comme une collection d'outils dédiée à la création de sites et applications web. De par sa simplicité d'utilisation et du grand nombre d'outils qu'il propose, Bootstrap s'est rapidement imposé comme la référence en termes de framework RWD. Sa dernière version, 3.3.0, date du 29 Octobre 2014.

#### Gumby

Gumby est un framework RWD lancé en Mai 2013 dans sa version 2.2.0 sous licence *Open Source MIT* et la dernière version en date est la 2.6.

## Comparaison

### Grille

Ces deux frameworks sont basés sur un système de grille qui permet de définir des affichages adaptatifs. Ainsi les affichages créés grâce à ces deux outils sont découpés en lignes elles-mêmes découpées en colonnes, comme le montre la figure suivante.



Figure 3 - Grille commune (source : <http://gumbyframework.com/docs/grid>)

---

<sup>3</sup> Cadre de travail

Cette grille présente différents critères, détaillés dans le tableau ci-dessous, selon le framework.

	Grille			Colonnes					CSS
	Nombre de colonnes	Imbrication de grilles	Hybride*	Différenciation de résolutions*	Décaler	Ordonner	Centrer	Agencement en tuiles	Classes
<b>Bootstrap</b>	12			 4 préfixes css : {xs, sm, md, lg}					Ex : « .col-md-3 »
<b>Gumby</b>	12 à 16								Ex : « two columns »

*Hybride : différents nombres de colonnes maximum au sein d'une même grille*

*Différenciation de résolutions : précision de la taille en fonction du type de dispositif*

## CSS

Ces frameworks permettent également de styliser un site internet rapidement grâce à des styles prédéfinis. Voici une liste non exhaustive de critères et d'éléments pris en charge.

	Préprocesseur CSS	Templates disponibles	Visibilité selon rés*.	Composants							Média
				Icones	Tables	Formulaires	Boutons	Indicateurs	Navigation	Panels	
<b>Bootstrap</b>	Less, Sass			Glyphicons, 250							Vidéos, audio, Images, tweets, ...
<b>Gumby</b>	Sass			200							Vidéos, images

\*résolution

## JavaScript

Ces outils embarquent également des scripts JavaScript permettant de rendre un site web plus dynamique, là encore de manière simple et rapide.

	Composants								
	jQuery	Directives	Gestion des lecteurs d'écrans	Modals*	Tooltip*	Toggle*	Pagination	Onglets	Validation de formulaires
<b>Bootstrap</b>	+	HTML5 (data-...)	+	+	+	+	+	+	+
<b>Gumby</b>	+	Personnalisées (gumby-)	-	+	+	+	-	+	+

*Modals : boîtes de dialogues se plaçant sur le contenu principal et le rendant inaccessible*

*Tooltip : information supplémentaire au survol d'un élément*

*Toggle : fait apparaître ou disparaître un élément, au click par exemple*

## Généralités

Enfin, terminons cette étude comparative avec des critères plus généraux.

	Navigateurs compatibles	Documentation	Communauté	Installation personnalisable	Extensions
<b>Bootstrap</b>	Chrome, Firefox, IE8-11, Opera desktop, Safari (Mac)	Complète et claire	Très importante et active, grand nombre d'utilisateurs	+	Nombreux plugins, intégration à de nombreux frameworks
<b>Gumby</b>	Chrome, Firefox, IE8-10, Opera	Complète, parfois confuse	Restreinte	+	Liste restreinte

## Conclusion

Bien que ces frameworks soient assez proches, Bootstrap l'emporte largement sur Gumby. Avec un nombre d'utilisateurs nettement supérieur, il dispose d'une communauté active qui lui permet d'évoluer régulièrement en proposant de nouvelles fonctionnalités et extensions. De plus, de nombreux IDE et frameworks populaires, tels que WebStorm ou AngularJs, permettent une intégration de celui-ci via des plugins.

Concernant la personnalisation de sites Internet, tous deux proposent des fonctionnalités similaires, le style demeurant à l'appréciation de chacun. Bootstrap propose tout de même des éléments supplémentaires qui permettront de pousser plus en avant la personnalisation. Bien que plus verbeuses, les classes CSS utilisées par Gumby semble aussi simples à utiliser que celle de son opposant – certaines sont même identiques - et le temps d'apprentissages des deux outils parait équivalent.

Là où Bootstrap est nettement en avance sur Gumby, et c'est le point qui nous intéresse particulièrement, c'est sur la plasticité de l'interface. En effet, bien que Gumby propose une grille hybride, il ne permet pas de spécifier avec précision la taille d'un élément en fonction de la taille du dispositif sur lequel il est affiché, ce qui constitue la grande force de Bootstrap. De plus, Bootstrap embarque depuis peu la gestion des lecteurs d'écrans, ce qui permet d'adapter un site web à un public malvoyant.

Cependant Gumby reste une très bonne alternative à son concurrent. Il propose même certaines fonctionnalités natives qui manquent à Bootstrap. De plus, ce framework est encore jeune et peut offrir de belles surprises dans ses versions futures. A suivre...

## Références

1. A. Bendinelli, F. Paternò : [Design Criteria for Public Display User Interfaces](#)
2. *Unified Modeling Language* : <http://www.uml.org/>
3. *Projet YourCast* : <http://yourcast.fr/>
4. *Bootstrap* : <http://getbootstrap.com/>
5. *Gumby* : <http://gumbyframework.com/>
6. *Licences Open Source* : <http://opensource.org/licenses>