

# Mémoire

Analyse et Conception d'un Modulateur de  
Présentation de l'Information sur des  
Terminaux Mobiles, Multi-Plates-Formes

Présenté par André OSTACHKOV en vue de l'obtention du titre  
de Maître en Sciences Economiques et de Gestion (IAGM)

Directeur : Monsieur Jean VANDERDONCKT



<b>PREFACE : RESUME ET METHODOLOGIE DU TRAVAIL</b> .....	<b>1</b>
<b>I. INTRODUCTION</b> .....	<b>3</b>
1. PRESENTATION ET ANALYSE DU PROBLEME .....	3
1.1. <i>Problèmes dus à la diversité</i> .....	5
1.1.1. Contraintes dues à la viabilité d’une application sur un terminal mobile .....	5
1.1.1.1. Caractéristiques et/ou préférences des utilisateurs finaux .....	5
1.1.1.2. Localisation et/ou contexte dans lequel se trouve l’utilisateur .....	6
1.1.2. Contraintes dues aux capacités techniques.....	6
2. PRESENTATION DES SOLUTIONS.....	8
2.1. <i>A propos des Standards</i> .....	8
2.2. <i>Indépendance du contenu</i> .....	9
<b>II. SOLUTION PROPOSEE SUR BASE DE XI ML</b> .....	<b>13</b>
<b>1. PAGE SOURCE</b> .....	<b>15</b>
1. XI ML: THE EXTENSIBLE USER-INTERFACE MARKUP LANGUAGE.....	15
1.1. <i>Indépendance par rapport aux outils de création d’applications</i> .....	16
1.2. <i>Indépendance par rapport à la plate-forme</i> .....	16
1.3. <i>Mise à jour automatique des composantes de l’IU</i> .....	18
1.4. <i>Génération automatique des IU</i> .....	19
1.5. <i>Conclusion</i> .....	20
<b>2. LE MODULATEUR</b> .....	<b>21</b>
METHODOLOGIE – ETAPES SUIVIES PAR LE MODULATEUR .....	22
POINTS OMIS PAR LE MODULATEUR.....	23
<i>Transformation de la présentation de l’information suivant le mode d’interaction homme - machine</i> .....	23
<i>Transformation de l’information sur base de l’ergonomie</i> .....	24
1. DECOMPOSITION DE L’ « INTERFACE DE BASE » EN NIVEAUX (ÉTAPE 1).....	26
2. ANALYSE DES POSSIBILITES DE TRANSFORMATION DE L’INFORMATION (ÉTAPE 2).....	28
2.1. <i>Analyse de l’information</i> .....	28
2.1.1. Propriétés déclaratives de l’information: Taxonomie des « modalités de sortie » .....	29
2.1.1.1. Définition des propriétés de base.....	30
a) Propriétés de base.....	30
Linguistique (vs. non-linguistique).....	30
Analogique (vs. non-analogique).....	30
Arbitraire (vs. non-arbitraire).....	31
b) Etat de représentation : statique vs. dynamique.....	31
c) Modes ou moyens de représenter l’information.....	31
Mode graphique .....	32
Mode Acoustique .....	32
Mode tactile.....	33
2.1.1.2. Représentation des niveaux.....	33
a) Niveau supérieur .....	33
b) Niveau générique .....	34
c) Niveau atomique .....	36
d) Niveau sub-atomique .....	37
Schéma de synthèse : Taxonomie des modalités de sortie par Bernsen.....	37
2.1.1.3. Tableau de synthèse : Taxonomie des modalités de sortie .....	38
2.1.2. Adaptation du modèle des « modalités de sortie » au modulateur .....	41

2.1.2.1. Le type d'information - d'objet .....	41
a) Mode Graphique.....	42
b) Mode Sonore .....	43
c) Mode tactile.....	44
2.1.2.2. Les caractéristiques de base.....	44
a) Mode Graphique.....	45
Positionnement.....	45
Unités de mesure.....	46
Arrangement.....	46
Caractérisation .....	48
b) Mode sonore.....	52
Positionnement.....	52
Arrangement.....	52
Caractérisation .....	54
c) Mode tactile.....	55
Positionnement - Arrangement .....	55
Caractérisation .....	56
2.1.2.3. Fichier .....	56
a) Format .....	56
b) Mode ou niveau de compression.....	56
c) Taille .....	56
2.1.3. Conclusion .....	57
Tableau de synthèse.....	58
2.2. <i>Etude des terminaux</i> .....	59
2.2.1. Choix des terminaux .....	60
a) Téléphones mobiles.....	61
b) Personal Digital Assistant (PDA) .....	62
La saisie d'informations.....	62
Types d'organiseurs électroniques.....	63
PalmOS™ .....	63
Pocket PC™ .....	64
Psion™ (EPOC) .....	66
c) Hybrides.....	67
Tableau de synthèse reprenant la typologie des terminaux .....	67
2.2.2. Contraintes physiques des interfaces-utilisateurs des terminaux .....	68
2.2.2.1. Définition et caractérisation des interfaces .....	68
a) Caractéristiques de l'écran .....	68
b) Caractéristiques des enceintes.....	69
Tableau de synthèse reprenant les IU et leurs caractéristiques.....	70
2.2.2.2. Répercussions des paramètres physiques des IU des terminaux sur les caractéristiques de base de l'information.....	71
2.2.3. Contraintes-systèmes liées aux terminaux .....	72
2.2.3.1. Définitions des contraintes-systèmes.....	72
a) Protocole.....	72
HTTP (Hypertext Transfer Protocol).....	72
WAP (Wireless Application Protocol).....	73
SMS (Short Message System).....	73
b) Langages de programmation Markup ou langage de marquage .....	73
HTML (Hypertext Markup Language) .....	73
WML (Wireless Markup Language).....	74

XML (EXtensible Markup Language) .....	74
c) Langages de programmation .....	74
d) Systèmes d'exploitation .....	74
e) Microbrowser .....	74
f) Caractéristiques du système.....	74
2.2.3.2. Incidence des contraintes-systèmes sur les différentes caractéristiques de base de l'information .....	75
a-b) Le protocole / langage de marquage .....	75
HTML .....	76
WML.....	77
c) Langages de programmation .....	78
d-e) Systèmes d'exploitation et le Microbrowser.....	78
Tableau de synthèse reprenant les possibilités de scrolling pour les terminaux choisis.....	79
f) Caractéristiques du système.....	79
Tableau de synthèse: Implications des caractéristiques-système .....	81
Tableau de synthèse : Les caractéristiques-système des différents terminaux .....	82
2.2.4. Conclusion .....	83
Tableau de synthèse : Caractérisation des terminaux .....	84
2.3. <i>Adaptation de l'information au terminal (Etape 3)</i> .....	85
2.4. <i>Synthèse graphique des deux premières étapes</i> .....	86
3. RECOMPOSITION DU SITE PAR NIVEAUX .....	87
3.1. <i>Usage standard du terminal</i> .....	87
Tableau de synthèse.....	91
3.2. <i>Réseau</i> .....	92
3.3. <i>Préférences des utilisateurs</i> .....	93
3.3.1. Discussion générale.....	93
3.3.1.1. Hors modulateur .....	93
3.3.1.2. Dans le modulateur .....	93
Illustration de la section (3.3.1).....	94
<b>3. APPLICATION DU MODULATEUR A UN ARBRE DE DECISION .....</b>	<b>95</b>
1. FORMULAIRE : « DEFINITION DE L'INFORMATION » .....	96
2. FORMULAIRE : « DEFINITION DU TERMINAL ».....	98
2.1. « <i>Incidence du Langage de marquage</i> ».....	99
2.2. « <i>Incidence du Microbrowser</i> ».....	101
3. CONCLUSIONS.....	101
<b>TABLE DES REGLES .....</b>	<b>103</b>
<b>III. BIBLIOGRAPHIE .....</b>	<b>105</b>
<b>IV. ANNEXES .....</b>	<b>109</b>
ANNEXE 1 .....	I
ANNEXE 2 .....	II
ANNEXE 3.....	III
ANNEXE 4.....	IV
ANNEXE 5.....	V
ANNEXE 6.....	VI
ANNEXE 7.....	VII
ANNEXE 8.....	VIII

## Préface : Résumé et méthodologie du travail

A cette époque, où les technologies ouvrent de plus en plus de possibilités aux transferts de données par les airs, nous assistons à une véritable effervescence des terminaux mobiles. Dans un premier temps, nous avons vu apparaître une multitude de terminaux employant leur système d'exploitation et leur langage propre. Cette situation était très difficile tant pour les entreprises de logiciels, que pour les consommateurs. Vu l'absence de toute uniformité sur le marché, les constructeurs se sont lancés dans une lutte afin d'imposer leur système. Aujourd'hui, ces entreprises, conscientes du problème, commencent à se réunir en conglomérat et tentent ensemble de créer des standards. C'est à cette période charnière que se place cette recherche. Elle s'inscrit dans un ensemble beaucoup plus vaste de travaux portant sur l'une des solutions les plus prometteuses : l'indépendance des données.

Ce travail propose une méthodologie afin d'adapter une interface programmée en XILM à n'importe quelle interface de terminal mobile. L'objectif final est d'obtenir une présentation de l'information qui tienne compte de l'ensemble des contraintes imposées par l'environnement dans lequel l'utilisateur évolue, y compris de la plate-forme.

Cette méthode est présentée comme une description du fonctionnement d'un programme. Celui-ci, à partir d'une interface utilisateur de base, transforme et adapte les différentes informations pour générer des interfaces-utilisateurs adaptées à des terminaux donnés. Ce « programme » a été appelé « modulateur ». L'ambition de ce mémoire n'est en aucun cas de créer ce programme mais bien de poser les bases d'un raisonnement qui, je l'espère, pourront un jour aboutir sur un tel projet. Pour illustrer cette recherche, vous trouverez, sur la disquette en annexe, le modulateur présenté sous la forme d'un arbre de décision interactif. De plus, pour mieux se rendre compte du genre de contraintes qu'impose une telle approche, le texte a été ponctué de règles<sup>1</sup>.

Le modulateur procède en trois étapes. La première étape consiste à décomposer le site en informations. La deuxième étape, la plus longue, vise à donner pour chaque information du site un ensemble de représentations possibles pour l'interface-utilisateur d'un terminal donné.

---

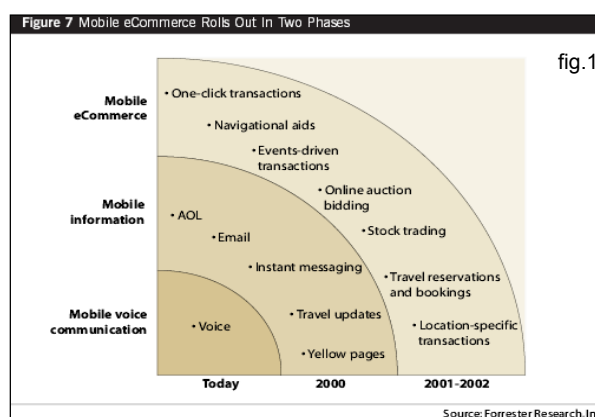
<sup>1</sup> Reprises dans une table en fin de travail.

Durant cette étape, le modulateur définit chaque information suivant un ensemble de caractéristiques. D'un autre côté, il modélise le terminal et finalement, il adapte les deux canevas suivant une méthode d'optimisation. Une fois l'ensemble des représentations délimité pour un terminal, la troisième étape vise à recomposer le site. Il faut donc choisir pour chaque information une représentation parmi celles possibles et réarranger ces informations de la manière la plus adéquate par rapport au terminal et à l'usage que l'on en fait.

# I. Introduction

## 1. Présentation et analyse du problème

Suite à l'explosion des produits et des services par Internet, les entreprises ne pourront plus se passer, dans un avenir proche, de l'e-business. De plus nous arrivons aujourd'hui dans une phase où les avancées technologiques (évolution des réseaux...) nous permettent d'entrer dans e-business mobile (cf. fig.1 : Evolution du développement de eCommerce (Forrester99)).



L'entreprise devra d'un côté répondre à une demande de plus en plus forte de ses clients et d'un autre côté, elle devra supporter, en interne, les coûts de la mobilité grandissante de ses travailleurs. Le service de recherche d'IBM prévoit qu'à partir de 2001, plus de 25% des travailleurs seront mobiles et que les entreprises devront consentir au moins 30% de leurs dépenses à supporter cette mobilité. Et ce, malgré la réduction du coût des télécommunications.

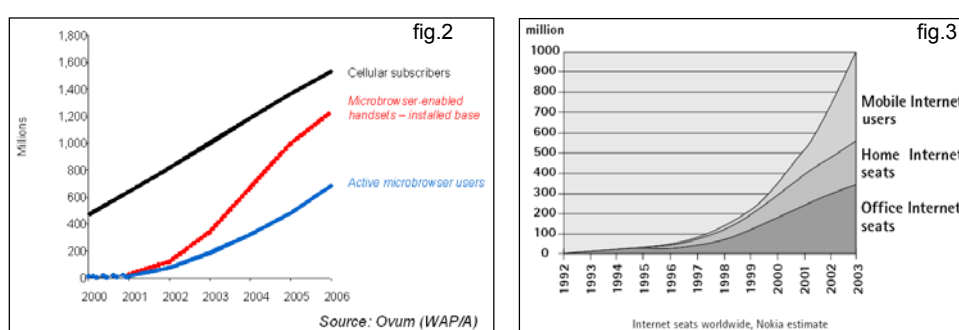
Pour répondre à la demande, les entreprises se devront d'être accessibles par le plus de personnes possible et par conséquent, par le plus de moyens possible. En effet, d'ici 2005, l'utilisateur sera confronté à au moins 5 plates-formes Forrester (99). Le nombre de terminaux à partir desquels ces services pourront être accessibles sera l'un des facteurs clés de leur réussite (KSF).

De plus, d'autres études nous montrent que les terminaux d'accès à Internet (WAP, PDA, UMTS, Web TV...) connaîtront une croissance de plus en plus forte, durant les prochaines années, au détriment des terminaux uniquement « vocaux » :

- *Le marché des appareils mobiles, multi-plates-formes croît 2 fois plus rapidement par an que*

celui des ordinateurs de bureau (Berst,98) ;

- Pour certains appareils tels que les GSM, le taux de pénétration est en forte croissance et peut atteindre jusqu'à 50% des personnes concernées dans le public et le privé (Ciancetta,99) ;
- Le marché des terminaux mobiles, multi-plates-formes d'accès à Internet est estimé à 1,77 billions US\$ en 2002 (Sybase,98) et converge avec les terminaux fixes (Ciancetta,99);
- Voici encore deux estimations mises sous la forme de graphiques. La figure 2 comme la figure 3 nous montre l'évolution du nombre d'utilisateurs de Microbrowser (programme permettant l'accès à Internet à partir d'un terminal mobile). Dans la figure 2, cette évolution est mise en rapport avec l'évolution du nombre des téléphones mobiles et du nombre de téléphones munis de Microbrowser (Ovum). Dans la figure 3, Nokia montre cette évolution en rapport au nombre d'ordinateur connecté à Internet, qu'ils soient privés ou en entreprise. Bien que ces chiffres diffèrent quelque peu, la tendance est claire.



Pour conclure, non seulement, le nombre de terminaux par utilisateur aura augmenté mais aussi leur diversité, de façon à convenir au mieux aux besoins du consommateur.

La situation du marché des appareils mobiles donnant accès à l'Internet, aujourd'hui, est semblable à la situation des PC's il y a un peu moins de vingt ans<sup>2</sup>. On est face à une multitude de matériels utilisant leur langage de programmation (API : Application Programming Interface) et leur *interface-utilisateurs*<sup>3</sup> (IU) propres (écran tactile, stylet, reconnaissance vocale, reconnaissance de l'écriture, voix de synthèse, miniaturisation des écrans...). Aucun standard stable n'existe pour l'ensemble de ces situations.

<sup>2</sup> Forrester Research (p5 Mobile World – Septembre 2000)

<sup>3</sup> Interface Utilisateur (IU) est la traduction littérale de l'anglais User Interface (UI). Notons que dans la littérature en français on retrouve le même concept sous le vocable de « Interface Homme-Machine » (IHM). Dans ce travail nous emploierons indifféremment ces termes.



## **1.1. Problèmes dus à la diversité**

Avant tout, la complexité diffère fortement d'un terminal à l'autre. On distingue, d'une part, les contraintes dues aux capacités techniques inhérentes aux terminaux et, d'autre part, les contraintes dues à la viabilité d'une application sur un type de terminal mobile. La viabilité d'une application relève de la spécification des tâches, c'est-à-dire quelles sont les tâches que l'utilisateur doit effectuer pour obtenir le résultat espéré et la façon dont elles s'organisent. Celle-ci est motivée soit par les caractéristiques et/ou préférences des consommateurs finaux, soit par la localisation et/ou le contexte dans lequel se trouve l'utilisateur.

L'objectif final est d'obtenir une présentation de l'information qui tienne compte de l'ensemble des contraintes imposées par l'environnement dans lequel l'utilisateur évolue, y compris la plate-forme<sup>4</sup>.

### **1.1.1. Contraintes dues à la viabilité d'une application sur un terminal mobile**

#### **1.1.1.1. Caractéristiques et/ou préférences des utilisateurs finaux**

Chaque terminal mobile possède ses caractéristiques qui lui confèrent un usage particulier pour un segment d'utilisateurs cibles. Il est donc nécessaire de considérer la spécification des tâches tant par rapport aux caractéristiques techniques de l'appareil qu'aux exigences de ses utilisateurs. Pour prendre l'exemple du téléphone mobile, l'un de ses atouts majeurs est que son utilisateur l'a toujours sur lui. Cela lui donne un accès opportun et immédiat à une information toujours mise à jour. De plus, on sait que le type d'information<sup>5</sup> consulté est précis, répété et orienté « business » (cours de bourse, annuaire téléphonique...). Le temps que les utilisateurs désirent passer sur un site est faible : il s'étend typiquement de quelques secondes à quelques minutes au maximum. Par conséquent, afin qu'une application réponde à ce type de demande, elle doit être rapide d'accès et fortement personnalisée. Ces critères doivent être considérés pour décider si oui ou non une application a du sens d'exister sur un terminal mobile.

---

<sup>4</sup> une plate-forme est une combinaison logiciel - matériel

<sup>5</sup> Par exemple la cible des téléphones mobiles est, pour l'instant un homme d'affaires. Les applications qu'il recherche sont simple, rapide et locale (localisées géographiquement) ex. bourse, pages jaunes... (Forrester [4])

D'un autre côté, il est nécessaire que l'utilisateur puisse interagir sur les contraintes de présentation du contenu tout en restant dans les limites techniques du terminal qu'il utilise. Cela implique que l'utilisateur puisse adapter le contenu aussi bien à ses aptitudes (capacité visuelle, auditive,...) et aux environnements dans lesquels il se situe (broyant, extérieur, réunion,...) qu'à ses préférences propres. Par exemple, un aveugle utilise un *browser* muni d'un écran tactile de lecture, il ne pourra accéder qu'à une information de type texte; un conducteur pourrait choisir une interaction uniquement vocale... Actuellement, le W3C dans la WAI (Web Accessibility Initiative)<sup>6</sup> fait des recherches (white papers, guidelines, Bobby<sup>6</sup>...) en matière d'accessibilité au Web pour tout type d'utilisateur<sup>7</sup>.

Lorsque l'on combine aux différentes capacités techniques des terminaux les attentes et/ou les besoins des consommateurs à l'égard des applications selon les terminaux, on conclut que beaucoup d'applications devront être entièrement reconçues lors du passage du PC vers des environnements mobiles. Une simple conversion de langages avec une réorganisation du layout suivant les possibilités d'affichage du terminal n'est, en aucun cas, une façon définitive d'obtenir une application « mobile » à partir d'une version PC. Les développeurs doivent donc gérer en plus de la forme, le contenu et la spécification des tâches liées à une application.

#### **1.1.1.2. Localisation et/ou contexte dans lequel se trouve l'utilisateur**

A cause de la grande mobilité des terminaux, il est aussi utile de considérer une adaptation du contenu basée sur la localisation et/ou sur le contexte dans lequel se trouve l'utilisateur. Par exemple, durant une réunion, l'utilisateur peut passer d'un mode vocal (sonore) à un mode visuel où le contenu est délivré de façon silencieuse.

#### **1.1.2. Contraintes dues aux capacités techniques**

On se rend bien compte qu'une même information ne peut être représentée de la même manière sur toutes les interfaces. Par exemple, l'affichage d'une image sur un écran de téléphone mobile requiert des adaptations comme : changement d'échelle, la troncature

---

<sup>6</sup> [www.W3C/WAI](http://www.W3C/WAI)

<sup>7</sup> Bobby est un programme développé en par la WAI. Il permet de vérifier l'accessibilité d'un site Internet par rapport aux standard développés par la WAI.

d'image (cropping), conversion des couleurs, réduction de la taille du fichier, réduction de la qualité de l'image, transformation du format de l'image... (cf. fig.4)

Pour pouvoir générer un « *format/layout final*<sup>8</sup> » valide, l'information sur les caractéristiques et les capacités des *interfaces* et des réseaux est

primordiale. Bien que des travaux de classification existent (ex. CC/PP<sup>9</sup>,...), la question de trouver une manière standard de modéliser les caractéristiques techniques inhérentes aux différents terminaux reste pertinente. Il est clair que déterminer une liste exhaustive de toutes les spécifications techniques possibles paraît difficile. C'est pourquoi, il est indispensable de se concentrer sur les paramètres pertinents, c'est-à-dire, nécessaires pour générer un « format final » en accord avec les possibilités du terminal et les exigences de l'application.

En plus des langages de programmation classiques (e.g., Java, JavaScript, Visual Basic, C++, Active-X...), les terminaux mobiles utilisent leur propre langage de programmation d'accès au Net: Wireless Markup Language (WML) pour les téléphones cellulaires ; Voice Markup Language (VoxML), Java Speech Markup Language (JSML) et SpeechML pour les appareils utilisant un terminal vocal (ex. Téléphones) ; HTML pour les ordinateurs de bureau et TVML pour les télévisions. Les programmeurs doivent donc employer une multitude de codes-source afin de développer différentes interfaces pour une seule information sur différents terminaux. Ce qui signifie qu'il faut programmer autant de fois une application que l'on désire avoir de terminaux à partir desquels cette application pourra être accessible. Il en est de même en ce qui concerne les mises à jour. Par exemple, si l'on a M informations à mettre à jour sur N terminaux (et donc potentiellement langages) différents, cela nécessitera  $M*N$  manipulations. Considérant la croissance prévue du nombre de terminaux dans les prochaines années, on se rend très vite compte de la lourdeur (temps – coût) que pourront prendre la programmation et le retraitement des données sur Internet. Sans compter le risque, en termes de manque à



<sup>8</sup> Le layout ou en français le format est la manière d'agencer les éléments d'une page

<sup>9</sup> Composite Capability/ Preference Profile (voir infra : section Modulateur 2.2.4.)

gagner, si l'on développe une application pour un terminal qui ne serait pas suivi par le marché.

Pour conclure, les programmeurs font face à des choix difficiles : soit ils essayent d'atteindre chaque type de terminal et adaptent les applications de sorte qu'elles soient optimales pour chaque terminal, en termes de caractéristiques techniques et d'usages qu'en font les utilisateurs. Ce qui signifie qu'il y aurait autant de versions de l'application que de terminaux disponibles sur le marché ou du moins souhaités pour l'application. Ce n'est pas très réaliste, compte tenu du facteur temps-coût. Si les programmeurs choisissent de prendre le plus grand commun dénominateur en ce qui concerne l'ensemble des caractéristiques pertinentes (technique, affichage,...), cela aurait pour conséquence des applications qui fonctionneraient de façon minimale sur tous les terminaux mais ne tireraient pas avantage des caractéristiques attractives et nouvelles mais propres à chaque interface. Une autre possibilité serait une solution de compromis qui pourrait se décliner suivant différents modèles: Nokia<sup>10</sup>, AIML<sup>11</sup>, XUL, UIML<sup>12</sup>, XIML<sup>13</sup>, Phone.com<sup>14</sup>... Pour une question de concision nous ne faisons que citer ces modèles alternatifs. Pour de plus amples détails à ce sujet, consulter les références en notes de bas de page.

## **2. Présentation des solutions**

### **2.1. A propos des Standards**

Il existe, bien sûr, la possibilité d'instaurer des standards concernant les caractéristiques des interfaces-utilisateurs pour l'ensemble des terminaux mobiles de sorte que tout le monde les suive, comme c'est, actuellement, le cas pour les ordinateurs de bureau. Mais, la standardisation possède sa propre vitesse d'évolution et d'acceptation au sein des gens du métier. Essayer d'imposer un standard de façon prématurée pourrait étouffer l'évolution et les

---

<sup>10</sup> [27]

<sup>11</sup> [63]

<sup>12</sup> [64-66]

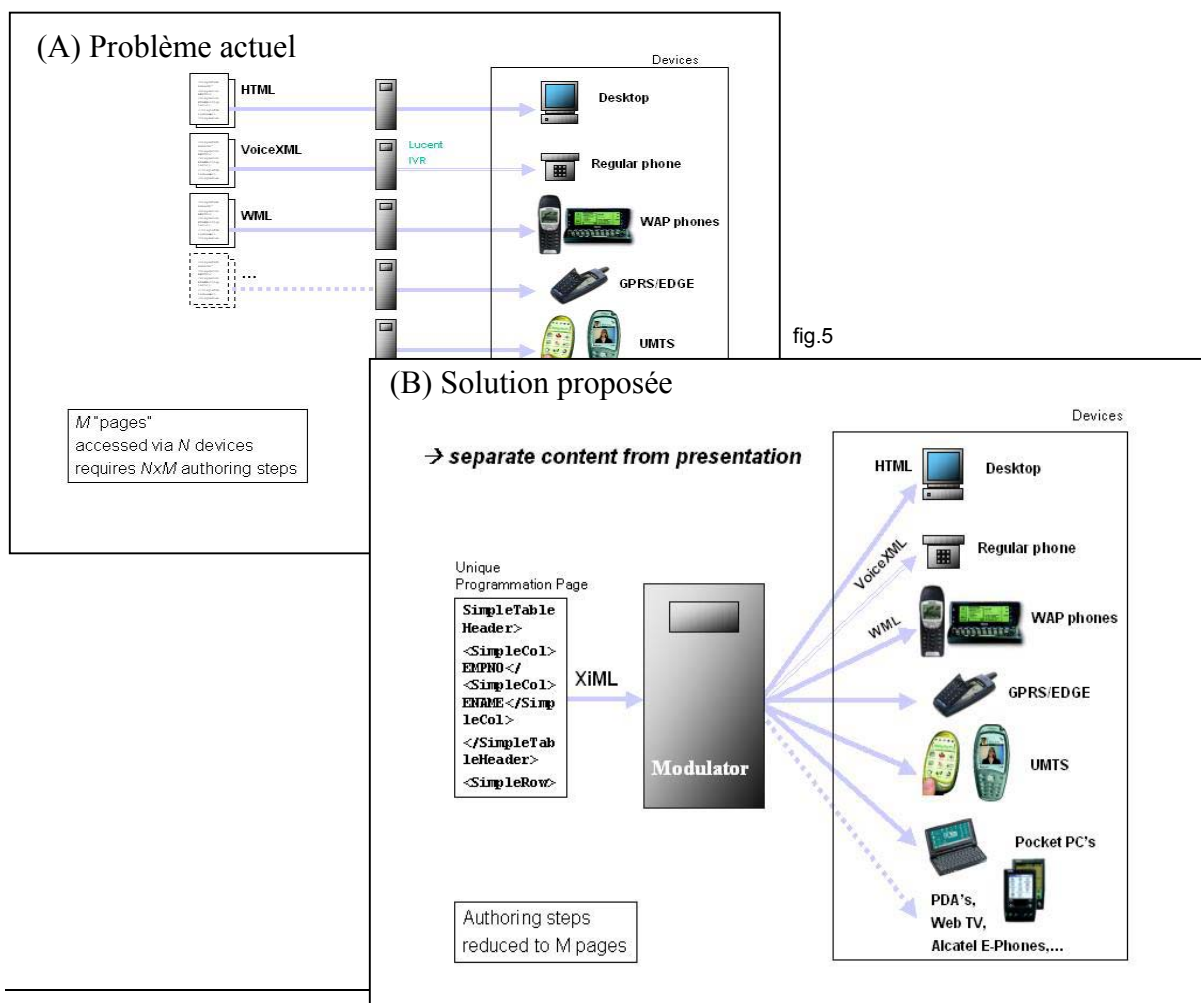
<sup>13</sup> Ce langage sert d'appui à la méthode présentée dans ce travail (cf. infra section Page source.1) [57]

<sup>14</sup> voir [www.phone.com](http://www.phone.com) [67-73]

possibles bénéfices de celle-ci. Le marché des appareils mobiles, tel qu'il se présente aujourd'hui, est un marché en pleine effervescence et en constante quête d'amélioration. Dans de telles conditions d'innovation, si quelqu'un présentait une solution meilleure à celles proposées par le standard celui-ci pourrait être immédiatement abandonné. Imposer un standard aujourd'hui est donc encore difficilement concevable ou du moins prématuré.

## 2.2. Indépendance du contenu

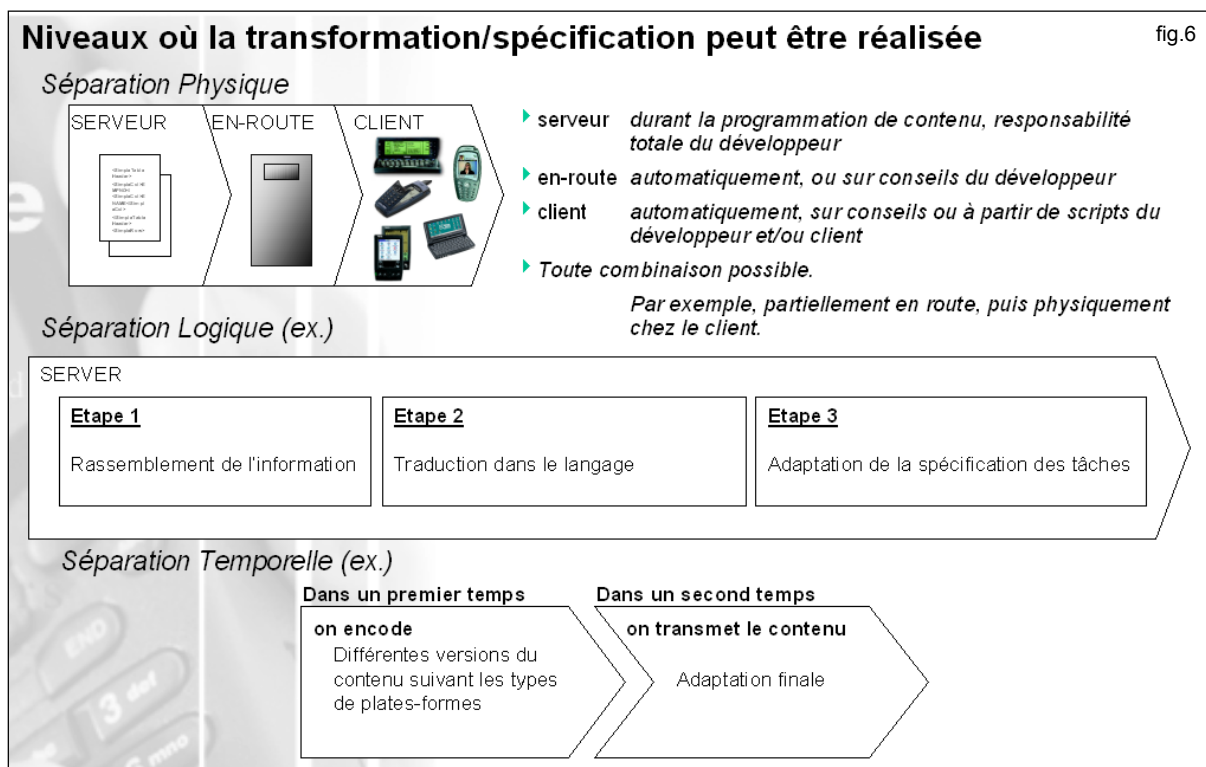
Pour palier à ces aléas, une façon possible de procéder est de définir le contenu de sorte qu'il soit indépendant du terminal (IU, spécificités techniques...), du réseau (différentes générations dans l'évolution 1,2 ou 3G<sup>15</sup>) et du contexte (temps, localisation, type de service...). La figure 5 nous illustre le problème tel qu'il se présente actuellement (5A) et tel qu'il pourrait être résolu avec l'indépendance du contenu par rapport au terminal.



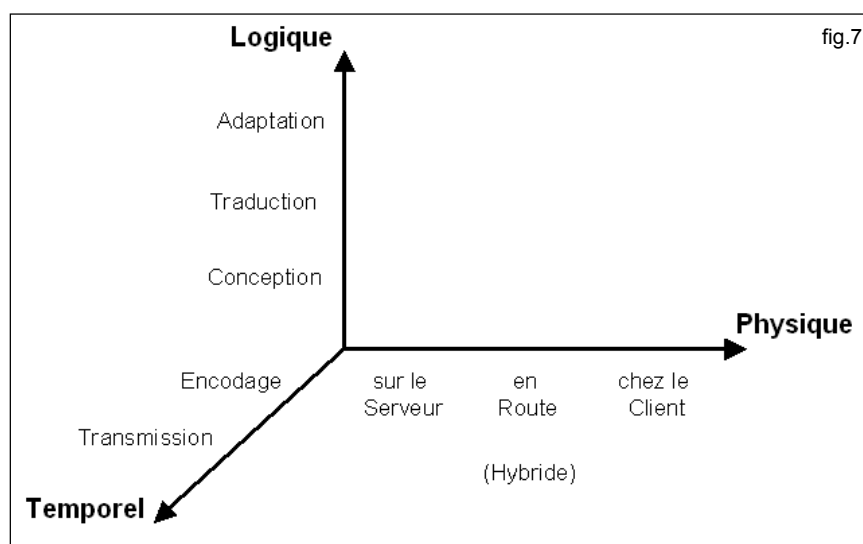
<sup>15</sup> nous reviendrons sur ce point à la section Modulateur 3.2

De cette façon, l'information ne devra plus être codifiée qu'une seule fois et celle-ci sera adaptée automatiquement aux différents terminaux et à leurs interfaces-utilisateurs propre. (cf. *fig5B*). Cette adaptation du contenu au terminal serait effectuée par le « *Modulateur* ». Par modulateur, nous entendons la méthode permettant cette adaptation. L'explication de cette méthode constitue le corps de ce travail.

Cette transformation ou adaptation du contenu, désormais indépendant, peut se faire de différentes manières (séparation physique, logique, temporelle,...) et/ou à différents niveaux dans l'architecture du réseau du service on-line (cf. *fig.6*). La séparation physique représente le niveau et l'endroit où se trouve concrètement le modulateur qui transformera le contenu brut en un contenu interprétable par le terminal visé. Il est noté que cette transformation ne se fait pas obligatoirement en une fois et peut être subdivisée en étapes successives ou simultanées (= critère temporel) suivant un critère logique, technique (réseaux, terminaux...), économique...(exemple *fig.6*). Il reste encore à trouver la structure optimale qui convienne à tous les acteurs (providers de e-services, fabricants d'appareils mobiles, HSP,...) et qui puisse servir de standard. Pour l'instant, on assiste encore à une effervescence de solutions proposées par divers acteurs qui privilégient un certain type de subdivisions suivant leurs intérêts propres ou suivant la technologie dont ils disposent (ex. [www.vizzavi.com](http://www.vizzavi.com)).



Etant donné qu'il est possible de combiner ces différents types de séparation, nous pouvons également les représenter sur un graphique suivant trois dimensions. Il est possible ainsi de créer une matrice des différentes possibilités (cf. fig.7).



Le sujet de l'indépendance entre les terminaux et l'information-source fait actuellement l'objet de nombreux débats et études. L'un des événements les plus marquant à ce propos, a été le *workshop* (octobre 2000) organisé par le *W3C*<sup>16</sup> et qui regroupait les principaux acteurs concernés (ex. : Nokia, Microsoft, Oracle, phone.com, Harmonia (UIML), HP,...). Après l'analyse des principales conclusions, regroupées sous la forme d'une SWOT, nous pouvons voir que les forces et les opportunités sont grandes et que, pour l'instant, les solutions qui existent ne sont que peu nombreuses et ne répondent que partiellement au problème.

En conclusion, nous pouvons dire qu'il y a une grande attente du marché en ce qui concerne un outil puissant et pratique capable de gérer une séparation efficace et intelligente entre le contenu et les terminaux. C'est à ce besoin que la solution présentée dans ce travail essaye de répondre.

<sup>16</sup> [www.W3C.org](http://www.W3C.org) - World Wide Web Consortium, organisme qui régit les standards du web.

## Device Independence SWOT analysis

Accessibility Focus	General Focus	Device Independent Application Model Focus
<b>Strengths</b>		
<ul style="list-style-type: none"> <li>Legislation</li> <li>Cost saving</li> <li>Many devices</li> <li>W3C</li> <li>Fashion</li> </ul>	<ul style="list-style-type: none"> <li>Increases audience</li> <li>Leads to increased content → <i>self-reinforcing spiral</i></li> </ul>	<ul style="list-style-type: none"> <li>Allows for content repurposing</li> <li>Accounts for digital TV concerns</li> <li>Allows for coping strategies</li> <li>Provides state and scope</li> <li>Rich user experience</li> </ul>
<b>Weaknesses</b>		
<ul style="list-style-type: none"> <li>Hard to define</li> <li>Learning curve</li> <li>Legacy</li> <li>Complexity</li> <li>No standards (e.g. TV, phone)</li> <li>Lack of education</li> </ul>	<ul style="list-style-type: none"> <li>Lack of compelling use cases</li> <li>Variable quality of service / experience</li> <li>Fixing legacy content</li> </ul>	<ul style="list-style-type: none"> <li>Doesn't account for 'close to the metal'</li> <li>yacc [<i>Yet Another Compiler Compiler</i>]</li> </ul>
<b>Opportunities</b>		
<ul style="list-style-type: none"> <li>WAI</li> <li>Market</li> <li>Education</li> <li>Internationalisation, translation</li> </ul>	<ul style="list-style-type: none"> <li>Future proofing</li> <li>Cool gadgets - <b>feature creep marketing</b></li> <li>New kinds of services</li> <li>New content markets</li> <li>Optimise user experience</li> </ul>	<ul style="list-style-type: none"> <li>Leverage VoiceXML, XForms etc → XML</li> <li>Tie in existing 'standards' without binding authors to specific schemas</li> <li>Leverage trends in localization</li> </ul>
<b>Threats</b>		
<ul style="list-style-type: none"> <li>Chicken and egg - who will start?</li> <li>Proprietary authoring tools</li> <li>Many guidelines - which one to use?</li> <li>Misconception</li> </ul>	<ul style="list-style-type: none"> <li>Perceived cost</li> <li>Fragmented web</li> <li>Copyright / control of content presentation</li> <li>People can wait</li> <li>May not be (<u>seen as</u>) necessary / useful</li> <li>[<i>Dilbert's manager icon</i>]</li> </ul>	<ul style="list-style-type: none"> <li>Large scope of work</li> <li>Requires mimetic engineering (<i>training, thought patterns</i>)</li> </ul>
<b>Wishlist</b>		
<ul style="list-style-type: none"> <li>Usable tools</li> <li>Tool integration</li> <li>Train users</li> </ul>	<ul style="list-style-type: none"> <li>Phones (etc) with APIs for Voice in/out, GUI /WIMP etc Multimodal preview in tools</li> </ul>	<ul style="list-style-type: none"> <li>Empirical data</li> </ul>
<p><b>Focus</b></p> <ul style="list-style-type: none"> <li>→ Education (ex W3C Certification)</li> <li>→ Tool development</li> <li>→ Involvement of industry</li> <li>→ Find user cases and business cases</li> <li>→ Focus on channels adapting authors intent</li> </ul>		





## II. Solution proposée sur base de XIIML

La solution présentée dans ce travail est basée sur le principe d'indépendance entre le contenu informationnel et l'interface homme-machine (IHM) d'un terminal donné. Il s'agit de construire un modèle tel que, à partir d'une unique page de codes-source, une application puisse être accessible et adaptée à n'importe quel terminal (*cf. fig.9*). Il y a donc deux grandes parties au modèle. Premièrement, il faut une page source qui soit programmée dans un langage suffisamment robuste d'un point de vue conceptuel et qui puisse être reconnu ou facilement adapté aux terminaux. Deuxièmement, il faut un modulateur qui puisse adapter les informations aux différents types de terminaux. Cette adaptation est faite sur base des caractéristiques physiques de l'interface-utilisateur, des contraintes-système du terminal (réseau, système d'exploitation, caractéristiques physiques du système...), des contraintes fonctionnelles du terminal dues à un usage normal (spécification des tâches), et des contraintes dues à l'utilisateur. Dans ce dernier type de contraintes on distingue trois sous-ensembles: les problèmes d'accessibilité (l'utilisateur est mal-voyant, mal-entendant...), l'adaptation nécessaire suite à un changement d'environnement (environnement bruyant, réunion, voiture,...) et, finalement, les préférences personnelles de l'utilisateur.

Dans le raisonnement qui suit, on a pris, par convention, comme hypothèse que le modulateur se trouve « en route », c'est-à-dire qu'il n'est ni à la source, ni dans le terminal. De cette manière le modulateur est totalement indépendant. Cependant, il serait possible d'imaginer d'autres cas. Par exemple, le modulateur pourrait être intégré dans la phase de programmation, afin de laisser au concepteur du site le contrôle des outputs générés, ou dans le terminal. Ces différentes possibilités seront analysées plus loin dans le texte.

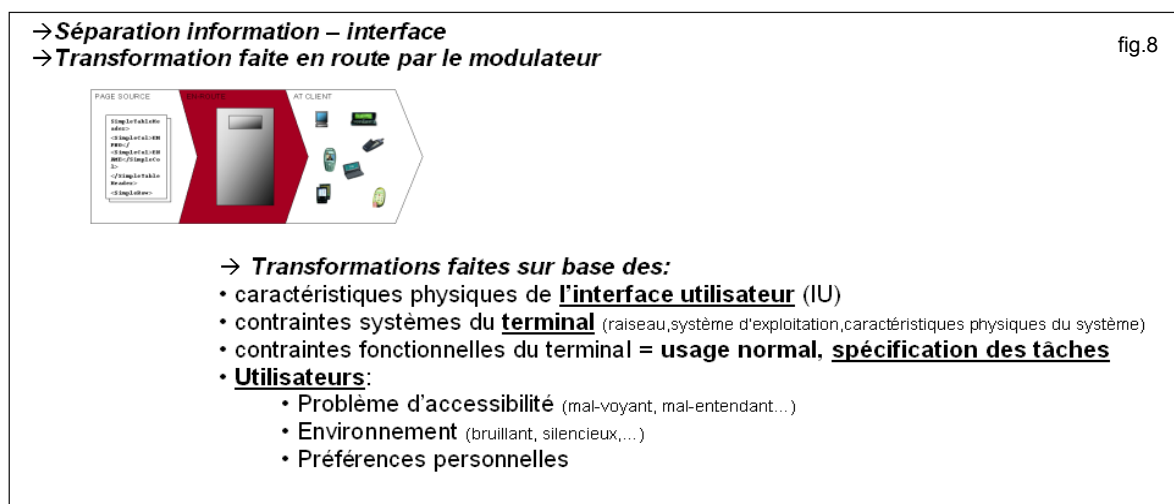
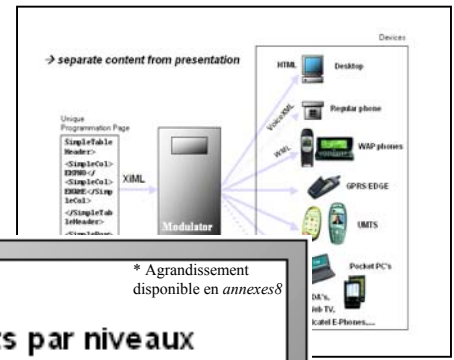


fig.9

# Détail du Modulateur

PAGE SOURCE EN XIML « Interface de base »  
 MODULATEUR (détail)



**iveaux\*** \* Agrandissement disponible en annexes8

**NGEMENT des objets par niveaux**

**Information**

Vueur d'informations

- objet graphique statique aux propriétés fixes (Image - dessin)
- objet graphique statique aux propriétés fixes (Texte)
- élément graphique statique aux propriétés fixes (tableaux (3 lignes et 2 colonnes))

## 2. Analyse des possibilités de transformation de l'information

**A. Informations/Données**

Caractéristiques  
Types de base Fichiers

Grafiques		
Sonores		
Tactiles		

Contenu de l'IU (output): ensemble d'informations de types différents organisés suivant des caractéristiques de base et possédant un format propre Ex.:

**B. Type de TERMINAL visé**

**Contraintes PHYSIQUES des IU**

**Contraintes SYSTEMES**

Protocole	Langage de programmation	Microbrowser
a.	b.	c.
d.	e.	f.

Langage de programmation Mark up      Systèmes d'exploitation      Caractéristiques du Système

**C. Adaptation séquentielle\* de l'information au terminal**

1. Contraintes Systèmes  
 2. Contraintes physiques de l'IU

**Dégradations Modifications**

	Caractéristiques de base	Fichiers
	Grafiques	
	Sonores	
	Tactiles	

➔ Intervalle [min;MAX] des possibilités de représenter l'information [2]

## 3. Recomposition du site par niveaux

[2]+

**Contrainte RESEAU**  
**Contraintes dues à l'USAGE standard du TERMINAL**  
**Contraintes dues aux préférences UTILISATEUR**  
 (Personnelles, d'Accessibilité et Environnement)

**Dégradations Modifications**

**POSITIONNEMENT ARRANGEMENT**

### IU adaptées aux TERMINAUX



Dans les parties qui vont suivre, on présente successivement les deux parties du modèle, à savoir : la « page source » et le « modulateur ». Ajoutons que la contribution de ce travail porte sur le développement du modulateur. C'est pourquoi, dans un esprit de concision, nous n'entrerons pas dans l'argumentation des différents langages possibles pour la page source.

## 1. Page source

Afin de pouvoir transférer de l'information, il est nécessaire que celle-ci soit codifiée dans un langage de programmation défini et compréhensible par le terminal vers lequel on veut l'envoyer. Cette page programmée et également appelée « *code-source* ». Dans le cas de ce travail, nous avons choisi le XIML comme langage pour la page source. Ce choix se justifie par le fait que seul ce langage permet de tenir compte de la spécification des tâches<sup>17</sup>.

### 1. XIML<sup>18</sup> : *The eXtensible user-Interface Markup Language*<sup>19</sup>

Le XIML est un langage qui permet de décrire une interface dans son entièreté sans se soucier de la façon dont cela va être implémenté. Il peut également être utilisé et échangé entre terminaux via Internet. De plus, le XIML est supposé être un langage de marquage très flexible pour ce qui est de la description d'IHM, c'est ce qui le rend peu sensible à l'évolution des terminaux. Ce sont principalement ces raisons qui placent le XIML en tête des candidats pour devenir un standard en matière de description d'IHM et c'est ce qui nous a poussés à le choisir comme code-source. Ce langage a été développé par un consortium de sociétés et d'unités de recherches universitaires dirigées par RedWhale Software<sup>20</sup>.

Comme tout langage dérivé du XML, le XIML est basé sur un contenu textuel dont la structure est donnée par des balises<sup>21</sup> de marquage. Ces balises sont définies par le schéma

---

<sup>17</sup> Le tableau comparatif des caractéristiques des différents langages candidats pour définir une IU se trouve en *annexes 1*

<sup>18</sup> La traduction française serait : *Langage Augmenté d'Interface-utilisateur Extensible*.

<sup>19</sup> Cette partie explicative de l'XIML est faite sur base de la traduction du document de Jacob Eisenstein (11/99), RedWhale Software Internal Papers on XIML.

<sup>20</sup> Voir [www.ximl.org](http://www.ximl.org)

<sup>21</sup> « Balise de marquage » est de terme français pour « tag » en anglais. Etant donné l'omniprésence de l'anglais dans le langage informatique, il nous arrivera d'utiliser un terme pour l'autre.

XML qui définit le langage. Un schéma XML doit être disponible sur le web et doit être partagé par quiconque utilise le langage. Le schéma indique aux applications qui lisent le XIML, ce qu'elles peuvent trouver dans un document XIML. Les applications qui éditent en XIML doivent s'assurer que le document créé soit conforme au schéma, de sorte que ce document soit compréhensible par les autres outils utilisant ce langage.

Les points qui suivent reprennent les principaux avantages de l'XILM.

### **1.1. Indépendance par rapport aux outils de création d'applications**

Actuellement, la conception des IU est faite à partir d'*interface builders* qui sont généralement intégrés dans des kits de développement plus larges, comme Microsoft Visual Studio. Cette approche ignore le fait que le développement d'IU est une approche cyclique avec des étapes discrètes qui doivent être réalisées de façon séquentielle. De même, cela rend impossible aux développeurs de dessiner l'IU dans un autre système que celui qu'ils utilisent pour le développement d'application. Cela oblige les développeurs à sacrifier un support utile et efficace dans la création d'IU en faveur d'un outil puissant quant à la création d'application, ou inversement. Etant donné que le XIML est un langage standard de description d'IU qui peut être utilisé par n'importe quel système de développement d'IU, il est désormais possible de créer un outil indépendant de création d'IU qui interagisse avec les développeurs d'applications. Cette innovation a un avantage : il n'est plus nécessaire de choisir entre une « belle » interface (de qualité) et une « bonne » application (performante) puisqu'il peut y avoir une certaine séparation entre la conception de l'application et le design de l'interface. Ajoutons que les qualités nécessaires à un programmeur ne sont pas celles d'un styliste (ou d'un ergonomiste) pour ce qui est de dessiner une interface. A présent, il est possible de sous-traiter cet aspect, donnant ainsi des applications aussi ergonomiques que puissantes.

### **1.2. Indépendance par rapport à la plate-forme**

Un autre point fort est qu'avec le XIML il est possible de rendre la conception de l'interface indépendante de la plate-forme, contrairement à ce qui se fait aujourd'hui. En effet, comme vu précédemment, l'IU est dessinée à partir d'un "package" de programmes destinés à la

programmation (ex. Microsoft Visual Studio, Symantec Visual Café, Metrowerk CodeWarrior) et pour un système d'exploitation<sup>22</sup> ou d'exploitation défini (ex. Windows98, Macintosh OS 9, Linux). Toute tentative de transposer l'IU sur une nouvelle plate-forme nécessite, à peu de choses près, de recommencer le travail. C'est dû au fait que le code de l'IU interagit avec le système d'exploitation. En fait, le code de l'IU est en général le code le moins transposable (ou portable) de l'application. Le XIML, quant à lui, se veut indépendant non seulement par rapport au système d'exploitation mais aussi aux environnements de développement et même aux langages de programmation. Une fois le programme écrit, cela convertit le XIML en code de programmation pour une plate-forme donnée. De cette sorte, toute IU écrite en XIML peut instantanément être transposée sur cette plate-forme. Cela signifie que le développeur d'applications sera capable d'écrire le code XIML une seule fois, pour fonctionner ensuite sur n'importe quel système opérationnel, sur n'importe quel terminal (ex. téléphone portable, palm pilot, PC,...) et être capable d'interagir avec des codes programmes quel que soit le langage dans lequel ils sont écrits. Eventuellement il pourra le faire moyennant des adaptations légères.

On peut noter que le même commentaire a été fait à propos du Java. La différence est que le Java est un langage de programmation et que pour obtenir une indépendance vis-à-vis de la plate-forme, il est nécessaire d'écrire le programme en Java. Avec le XIML, on peut employer le langage que l'on veut (ex. C++, Java, Lisp), pour autant que l'on décrive l'IU en XIML.

Une application de ces propriétés pourrait être une UI sur mesure (« customisation »). Le programmeur d'une application peut, s'il le désire, exporter le document XIML pour l'IU ainsi que l'application. Cela permet à l'utilisateur de modifier le document XIML, et par là-même de personnaliser son application. Il est, en effet, démontré que la plupart des utilisateurs n'emploie que dix pour cent des fonctions offertes par un programme. Le problème est que ces dix pour cent sont différents suivant les utilisateurs. Cette idée d'une IU modifiable donnerait à l'utilisateur l'opportunité de rendre "son" application plus accessible. Cependant, le créateur du programme garde le contrôle sur ce qui pourrait ou non être modifié par l'utilisateur. Les modifications concernant l'IU se feraient au travers d'une fenêtre de

---

<sup>22</sup> Système d'exploitation est la traduction française de Operationg System (OS) néanmoins une traduction plus littérale pourrait être « Système Opérationnel ». Désormais, nous utiliserons indifféremment ces notations.

dialogue plus que sous la forme d'un code "ouvert". De cette sorte, le code-source ne serait pas divulgué au public. N'oublions pas que l'utilisateur n'aurait accès qu'au code XIML et non au code du programme, vu l'indépendance du code (voir supra).

### 1.3. Mise à jour automatique des composants de l'IU

Il est important de souligner que la mise à jour peut s'effectuer soit à l'interprétation du code par le programme, c'est-à-dire à l'exécution (at run time), soit à la programmation, c'est-à-dire à la conception de l'application (at design time).

Le XIML, quant à lui, construit l'IHM à chaque fois que l'application est chargée et lancée par l'utilisateur (at run time). C'est ce qui permet de modifier l'interface après la compilation du programme, comme expliqué antérieurement. On appelle cette avancée technologique "l'IU interprétée"<sup>23</sup> car le document XIML est réinterprété à chaque fois que le programme est lancé et non compilé une fois pour toutes. Cette avancée offre la possibilité d'une mise à jour automatique des composants de l'IU. Le mécanisme est simple, il suffit de définir un *composant de l'IU* comme une partie de l'IU qui n'est pas créée par le dessinateur de l'interface, mais bien achetée, ou sous patente, par le vendeur du programme. Le composant de l'IU est traité comme une "boîte noire" par le programmeur du software<sup>24</sup>; celui-ci dépend d'une série de procédures pour le manipuler.

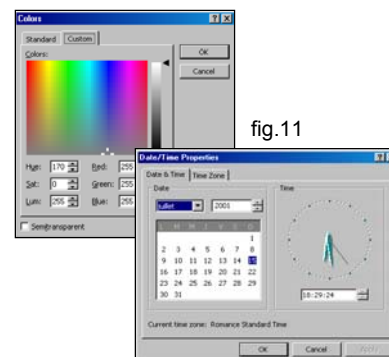
L'emploi de composants d'IU devient, de plus en plus, chose courante. Les outils



traditionnels de design d'IU, offrent généralement une palette de composants IU (ex. *check boxes, radio button, sliders...* (cf. fig.10).

Le designer n'a plus qu'à tirer ces composants de la palette vers la

fenêtre de dialogue préalablement dessinée. Mais les composants peuvent être de plus en plus complexes, avec l'apparition de contrôles *ActiveX* (ex. de composants complexes: contrôle du calendrier Windows (cf. fig.11), outil de sélection des couleurs sur Apple).



<sup>23</sup> "The Interpreted User-Interface technology"

<sup>24</sup> mot anglais pour dire application.

le programme code, mais *la technologie de l'IU interprétée* permet de récupérer ces composants en temps réel sur le net. Supposons que Microsoft sorte un nouveau contrôleur de calendrier. Sans XIIML, il faudrait recompiler l'application avec le nouveau contrôleur et émettre un "update"<sup>25</sup> ou un "patch". Avec XIIML, la mise à jour serait faite automatiquement, pour autant que Microsoft mette le nouveau contrôle sur l'URL correcte. Naturellement, XIIML permet la possibilité d'une sauvegarde (*backup*), qui peut être locale, de sorte que l'IHM puisse fonctionner correctement dans le cas d'un problème de réseau. Par exemple : indisponibilité ou temps de réponse en baisse.

#### 1.4. Génération automatique des IU

Contrairement à d'autres langages (ex. UIML), XIIML tient compte de la spécification formelle de l'information abstraite. Une telle information comporte des hiérarchies de tâches, de l'information de domaine et la modélisation de l'utilisateur. Dans la perspective de la recherche, la conception de l'IU est vue comme un processus de mouvement partant d'une telle spécification abstraite à une spécification abstraite puis concrète de l'IU. Cette vue du design de l'IU peut potentiellement mener à certaines applications qui valent la peine d'être explorées.

Il a été montré que pour certaines applications, cela peut être fait automatiquement. C'est-à-dire que, à partir d'une spécification de la tâche et d'une information du domaine étant données, une IU peut être générée automatiquement par l'ordinateur. Cela fonctionne bien pour les applications de gestion et les système d'information, telles que les formulaires conçus pour le Web<sup>26</sup> et la visualisation de données. On peut pour cela se référer aux premiers travaux de Puerta<sup>27</sup> qui traitent de la génération automatique d'applications sous forme de formulaires à partir de spécifications abstraites. Un designer de Web pourrait utiliser XIIML et spécifier l'information qu'il veut obtenir des visiteurs de sa page Web. Il ne devrait rien dire, en réalité, sur la manière de construire l'interface ou le design des pages ; aucun encodage

---

<sup>25</sup> fichier ou programme de mise à jour

<sup>26</sup> Web-based forms

<sup>27</sup> [voir A.R.Puerta et al., Model-Based Automated Generation of User Interfaces, in Maybury, M. and Wahlster, W., eds. *Readings in Intelligent User Interfaces*. San Francisco: Morgan Kaufmann 1998] [www.arpuerta.com](http://www.arpuerta.com)

dans un langage particulier (ex. HTML, Javascript ou CGI) ne serait requis. A la place, le designer de Web alimenterait le document XIML partiellement complété dans un software qui générerait automatiquement toutes les informations concrètes de l'IU. Le document XIML maintenant complété, pourrait alors être utilisé pour créer l'IHM.

Similairement, un programmeur de base de données pourrait utiliser XIML pour spécifier un nombre de « slots<sup>28</sup> » qui pourraient être remplis avec l'information en provenance de la base de données. Ce programmeur a besoin seulement de spécifier un squelette général décrivant la structure que l'information prendra. L'utilisateur devrait alors soumettre une requête, et ces « slots » seraient remplis avec l'information venant de la base de données. Une fois de plus, un software interviendrait et générerait automatiquement l'information concrète de l'IU. De cette manière, une présentation graphique peut être générée automatiquement pour visualiser l'information de la requête dans la base de données. Une recherche a déjà été faite pour montrer que la génération automatique de telles présentations graphiques est faisable<sup>29</sup>.

## 1.5. Conclusion

Le XIML est un langage de programmation d'IU puissant et flexible qui pourrait, sans conteste, devenir un standard en la matière. Il permet de décrire une interface dans son entièreté et de façon indépendante au reste, qu'il s'agisse du langage de programmation de l'application dont il organise l'interface ou du système d'exploitation sur lequel l'interface va être implémentée. Ces caractéristiques et le fait que l'XIML construit l'IU à chaque lancement ou chargement de l'application permettent de modifier celle-ci après que le programme ait été compilé. Cela permet une série d'applications fort utiles comme, par exemple, la possibilité pour l'utilisateur d'avoir une interface sur mesure, la mise à jour automatique des composants de l'IU ou bien même la génération automatique des IU. C'est à cette dernière que nous allons nous intéresser car elle est la plus utile parce qu'elle permet le plus grand gain de temps et de coûts. En effet, cette propriété permet à une entreprise qui veut

---

<sup>28</sup> créneaux

<sup>29</sup> voir S. K. Feiner and K. R. McKeown, Automating the Generation of Coordinated Multimedia Explanations, in Maybury, M. and Wahlster, W., eds. *Readings in Intelligent User Interfaces*. San Francisco: Morgan Kaufmann 1998



être accessible sur le Web, une fois son application programmée dans un langage, de l'être automatiquement sur l'importe quel type de terminal sans devoir reprogrammer l'application en fonction de chacun de ceux-ci. Il faudrait pour cela, qu'il y ait un outil qui puisse modéliser tant l'information que les terminaux et qui, pour chaque type de terminal, génère une IU adaptée à celui-ci. Cet outil sera décrit et analysé dans les chapitres suivants sous le nom de « modulateur ».

## 2. Le modulateur

Le modulateur se veut un outil capable de générer, au moyen de l'XIML, des interfaces-utilisateurs adaptées à n'importe quel type de terminal. Vu la complexité du problème et afin de rester pratique, cette présentation sera ponctuée de règles ou de conseils pour permettre à l'utilisateur de cette méthode de correspondre aux hypothèses du modèle et d'avoir, au final, le meilleur résultat possible.

Pour parvenir à cet objectif, il est nécessaire de modéliser tant l'information et la façon dont elle s'organise que les terminaux. En ce qui concerne l'organisation de l'information sur une « page » de l'IU, il y a un côté subjectif dû au style que le designer veut donner au site<sup>30</sup>. En effet, chaque site se différencie, en plus de l'information ou des services qu'il propose, par l'aspect et le style de présentation de cette l'information. Afin de respecter cette diversité de styles, et pour ne pas générer des interfaces toutes semblables, le modulateur ne génère pas d'interface directement à partir de l'application mais prend pour point de départ une *interface de base*, c'est-à-dire, une interface dessinée par un ergonomiste. Cette interface de base peut être soit conceptuelle, soit adaptée à un terminal. Dans le premier cas, le site serait construit de manière générale avec l'aide d'un éditeur et reprendrait l'information dans le format le plus large pour qu'il soit le plus apte possible aux dégradations. Dans le deuxième cas, pour la même raison, il est nécessaire que le créateur du site le conçoive pour un terminal qui offre le plus de possibilités. L'ordinateur de bureau pourrait être un bon terminal « de départ ». Néanmoins, la première solution est préférable mais nécessite un nouvel outil d'édition,

---

<sup>30</sup> site = ensemble de pages inter reliées dans lequel l'utilisateur peut se déplacer au moyen d'un browser

spécialement conçu pour l'XIML.

Afin de donner un aspect pratique à étude, on ponctuera le travail de règles ou plutôt de conseils qui permettront, si on les respecte, de correspondre au mieux aux hypothèses posées. De cette façon, on pourra obtenir un résultat optimal (dans les limites fixées). L'énoncé de ces conseils précédés du mot « Règle ». Elles sont numérotées et reprises dans la table des règles suivant l'ordre d'apparition. De plus, pour rendre ce travail plus interactif, il est possible de visualiser le raisonnement développé dans la troisième étape du modulateur (II.2.2.3) et d'interagir avec celui-ci au moyen d'un arbre de décision contenu dans la disquette en annexe<sup>31</sup>.

La première règle et peut-être la plus importante est la suivante :

***Règle 1. Lors de la création de l' « interface de base » prendre la forme d'information la plus large possible (ex. image en bmp...). Il est toujours possible de dégrader un format, l'inverse ne l'est pas.***

## **Méthodologie – Etapes suivies par le modulateur**

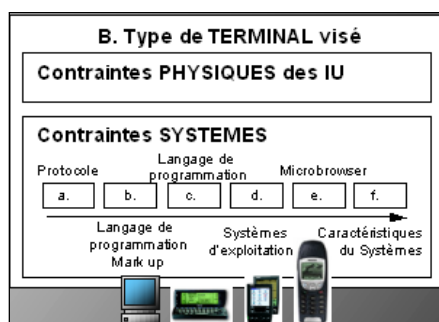
Le modulateur procède à la décomposition, l'analyse et à la recombinaison de l'interface de base. Les étapes suivies par le modulateur visent à limiter vers le haut les caractéristiques des informations de façon séquentielle. Bien que cette hypothèse soit discutable et peut se montrer non-optimale pour de nombreux cas, dans un premier temps, nous l'acceptons comme telle. La première étape est de décomposer l' « interface de base » en objets et en informations tout en analysant le positionnement et l'arrangement par niveaux (Chapitre 1). Dans un deuxième temps, le modulateur caractérise les données ou informations initiales (chap.2.1) et analyse les possibilités de transformation de l'information selon le type de terminal visé (Chapitre 2). Cette caractérisation détermine pour chaque information de la page un panier d'attributs ou caractéristiques de base. Comme vu précédemment, ils seront pris comme des maximums. Bien que l'augmentation des attributs soit possible (ex. : agrandir la taille d'une image), elle entraîne une perte de qualité. C'est pourquoi, pour ne pas détériorer l'image de l'application, le modulateur ne pourra pas augmenter la définition des caractéristiques de base. Le reste du

---

<sup>31</sup> Pour plus de détails se référer au chapitre II.3, Application du Modulateur à un arbre de décision.

processus se présente comme une suite de maximisations séquentielles des attributs de base sous les contraintes des terminaux (Chap.2.3). Ce qui signifie qu'à chaque étape, pour chaque type d'informations, un nouvel ensemble de caractéristiques maximales est établi. Au final cette étape nous donne un ensemble (borné par le haut) de possibilités de présenter l'information, Les contraintes liées au terminal (chap.2.2) sont dans l'ordre :

1. Les caractéristiques physiques de l'IU du terminal (Chap.2.2.2)
2. Les contraintes-système du terminal (Chap.2.2.3)



- a. Protocole (http, WAP...)
- b. Langages de programmation Markup (WML, HTML...)
- c. Langages de programmation (Java, flash, php...)
- d. Systèmes d'exploitation (WindowsCE, Epec, PalmOS...)
- e. Microbrowser (Nokia WAP browser, Phone.com...)
- f. Caractéristiques du système (carte son, ROM/RAM...)

La dernière étape du modulateur est la recombinaison des objets sur base les informations, retravaillées par le modulateur et leur repositionnement sur l'interface (Chapitre 3). L'arrangement et le repositionnement se font à partir des

1. Contraintes **réseaux**
2. Contraintes dues à l'**usage** standard du terminal.
3. Contraintes dues aux préférences de l'**utilisateur** (personnelles, accessibilité, environnement)

### **Points omis par le modulateur**

On reprend dans ce paragraphe les points qui, bien que nécessaires pour une adaptation optimale de l'information par le modulateur, ne seront pas repris dans ce travail. Néanmoins, le modulateur reste pertinent du fait qu'il serait possible de compléter le modèle par la suite. On pourrait traiter ces « lacunes » en ajoutant des « modules » au modèle existant.

### **Transformation de la présentation de l'information suivant le mode d'interaction homme - machine**

On peut noter que la façon dont l'utilisateur d'un terminal introduit (*input modalities*) l'information ou interagit avec la machine, influence aussi la façon de représenter

l'information sur l'interface. Par exemple, sur un téléphone mobile, qui ne dispose de manière standard que d'un clavier neuf touches, l'introduction de texte (text box) avec le clavier est une procédure lente et laborieuse. Il est donc préférable de présenter l'information de sorte que l'utilisateur n'ait qu'à la sélectionner avec les touches de direction (ex. option button, check button, ou liste). Cette problématique, bien que fort intéressante, sera seulement évoquée dans le chapitre concernant l'usage standard du terminal. Pour de plus amples informations à ce sujet se référer aux travaux de Buxton (1983), Foley (1984), Lenorovitz (1984), Greenstein & Arnaut (1988), Bleser and Sibert (1990)<sup>32</sup>.

### **Transformation de l'information sur base de l'ergonomie<sup>33</sup>**

De nombreuses études ont été menées pour voir quelle est la « meilleure » façon de présenter un certain type d'information, afin de l'adapter au mieux à son utilisateur. Par exemple les travaux des chercheurs de l'université de Cleveland<sup>34</sup> ont porté sur les capacités du système perceptif humain à extraire de l'information à partir de graphiques suivant leur présentation (angle de vue, aires, longueur, volume,...), ainsi que l'importance de la perspective dans la présentation des objets sur une IU (synthèse de leur travaux en annexe 3). D'autres exemples intéressants pourraient être la catégorisation des techniques d'attraction de l'attention de l'utilisateur de Vibhu O.Mittal<sup>35</sup>, l'étude de l'ergonomie des formulaires,<sup>36</sup> ou encore la comparaison entre les méthodes d'introduction directe contrairement aux méthodes de sélection<sup>37</sup>.

Les résultats de ces études et bien d'autres pourraient être synthétisés sous la forme de règles ou de conseils à donner au designer de l'IU lors de la création de celle-ci. Ils pourraient aussi être intégrés dans un outil d'édition en XIML (cf. supra) qui pourrait, à partir des données fournies, générer l'ergonomie adéquate et même, dans certains cas, organiser les données en

---

<sup>32</sup> Vous trouverez un récapitulatifs de ces recherches sur les « input modalités » en annexes 2.

<sup>33</sup> Recherche d'une meilleure adaptation entre une fonction, un matériel et son utilisateur (dictionnaire Larousse)

<sup>34</sup> [39]

<sup>35</sup> [34]

<sup>36</sup> [36]

<sup>37</sup> [37] et [41]

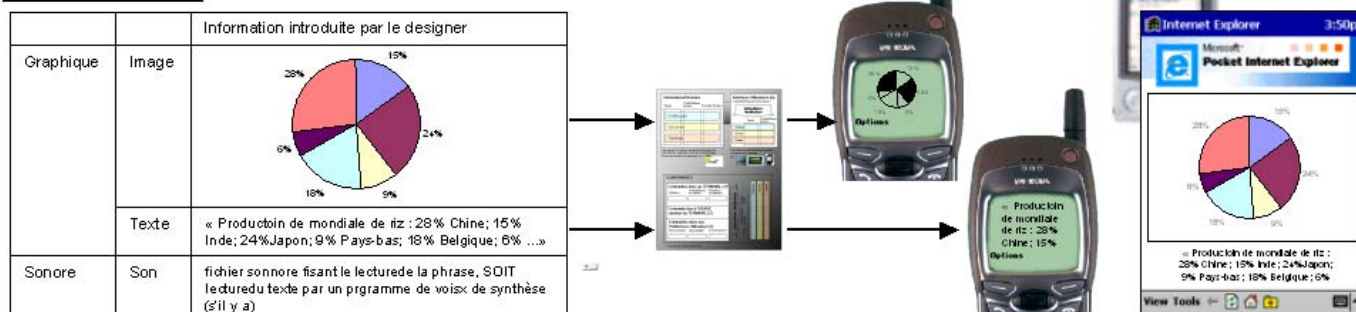
niveaux (ex. : la représentation des graphiques (cf. fig.12)). Comme le modulateur part d'une interface de base, il considère que l'ergonomie a déjà été optimisée. Par conséquent, il n'en tient pas compte et considère les objets comme tels. Néanmoins, il serait possible de revoir l'ergonomie des objets suivant leur dégradation ou de l'intégrer dans l'outil d'édition en l'XIML de l'IU de base (comme évoqué précédemment). Etant donné les limites fixées à ce travail, nous laissons cela comme pistes à des recherches futures.

Le modulateur, ne tenant pas compte de ces aspects, gardera, dans un premier temps, le type d'interaction défini dans l'interface source. Il en sera de même pour les différents objets, le modulateur les considérera tels que dans la page source, sans générer aucune transformation interne au type. Les seules transformations possibles sont des dégradations de caractéristiques propres (voir infra : section 2.1.1, Analyse de l'information) ou des changements de modes parmi ceux définis dans l'interface de base.

## Exemple du fonctionnement du modulateur vs. existence d'un outil d'édition de l'Interface de base. (Cas d'un graphique)

fig.12

**Modulateur:** l'information est introduite par le designer suivant les différents modes



→ Information adaptée au terminal sans aucune modification du «contenu»

**Outil d'édition en XIML:** la présentation de l'information est générée automatiquement par l'outil sur base des données fournies par le designer et suivant l'ergonomie et le mode d'interaction avec le terminal

*Données intraduites*

Production mondiale de riz en milliards de tonnes	Pays-bas	Japon	USA	Inde	Belgique	Chine
	5	8	3	6	2	9










*ex. de textes générés à partir des données*



→ Possibilités d'exploiter les données en générant des sous-niveaux.

ex.: on pourrait en choisissant la chine avoir un graphique la concernant

*Graphiques générés*

	Simple	Aire	Surface 3D	Volume 3D
Points				
Lignes*				
Barres				
Parts**				

## 1. Décomposition de l' « interface de base » en niveaux (Etape 1)

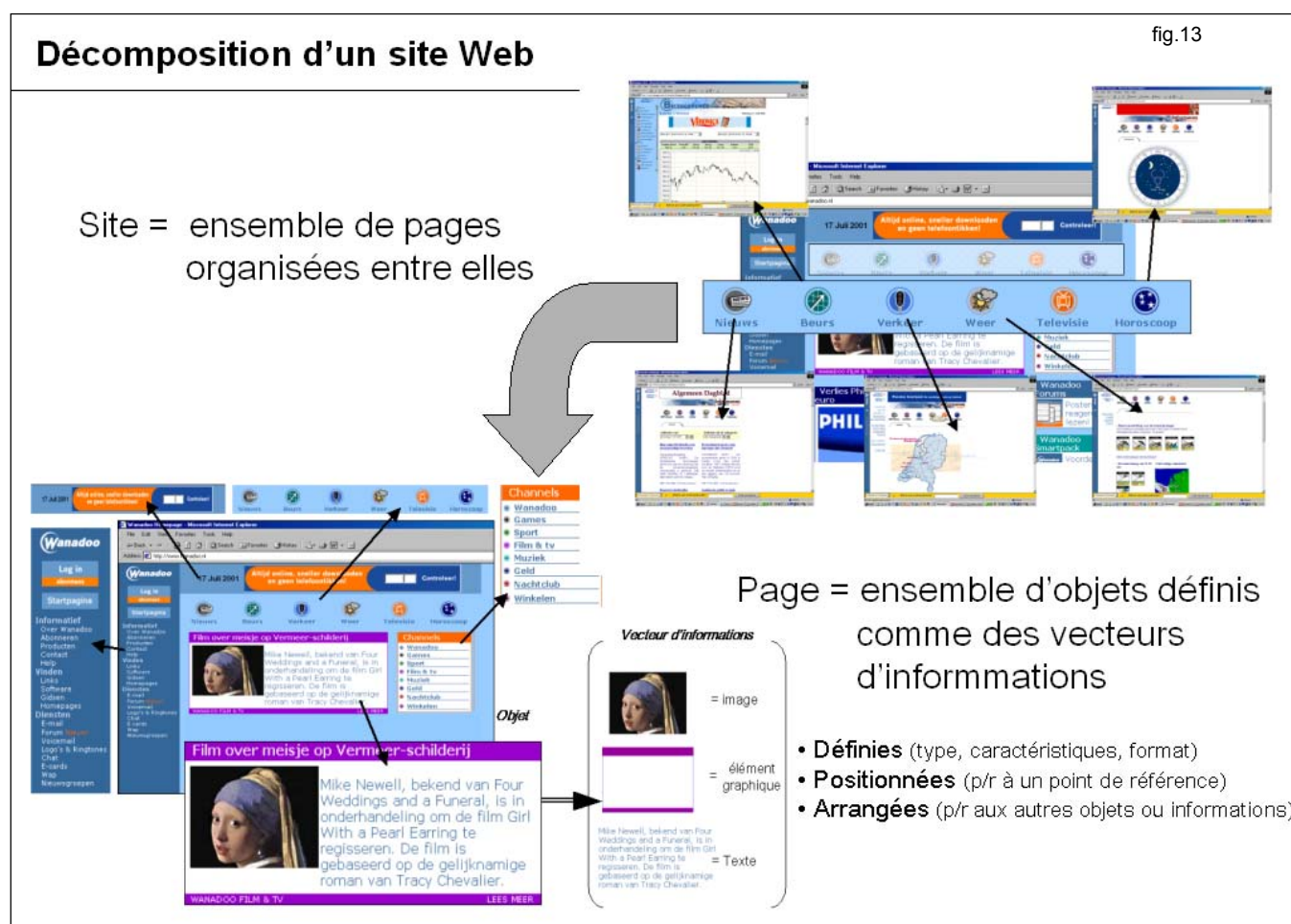
L'interface d'une application est un ensemble d'informations organisées de façon complexe qui interagissent entre elles. Afin de pouvoir l'analyser au mieux, il faut la décomposer en éléments. Ces éléments ou objets sont à leur tour décomposés et ce jusqu'à l'obtention d'éléments indivisibles ou purs. Une fois que l'on ne dispose plus que d'objets purs, appelés informations, il faut analyser ces objets en commençant par le niveau le plus faible et les recomposer par élément, suivant les niveaux, et de façon cohérente avec l'original. Cette méthode par niveau, permet d'être sûr qu'aucun élément n'a échappé à l'analyse. Chaque objet se définit comme étant :

1. Défini ou caractérisé = défini en tant que tel (type, caractéristiques, format (cf. section suivante)). Notons que pour les objets « non-purs » seules certaines caractéristiques sont pertinentes.
2. Positionné = par rapport à un point de référence physique (ex. bords ou centre de l'écran) ou au browser.
3. Arrangé = par rapport aux autres objets de la même classe.

Dans le cas d'une application mobile, l'interface se présente sous la forme d'un site. Suivant la méthode de décomposition, on peut définir un site comme étant un ensemble de pages organisées entre elles. Les pages forment ainsi le premier niveau de décomposition. Elles peuvent, à leur tour, être subdivisées en objets (niveau 2) qui eux-mêmes sont composés d'informations pures (niveau 3) (cf. fig.13). Etant donné que la plupart des sites organisent l'information dans des tableaux, le modulateur définira l'arrangement entre les objets par rapport à l'emplacement qu'ils occupent dans le tableau. Ensuite, dans chaque cellule on voit l'alignement de l'objet (vertical et horizontal). Cette étape est primordiale car, c'est sur base de cette décomposition que le modulateur va réarranger et repositionner l'information attribuant à chaque objet un degré d'importance relatif à son niveau. Etant donné qu'au sein d'un même niveau, certaines informations peuvent être plus importantes que d'autres, il est intéressant que le créateur du site organise lui même la hiérarchisation des données. Cette hiérarchisation pourrait ce faire par un système d'annotations ou de tags mais ceci reste encore à définir.


**Règle 2.** Lors de l'édition d'un site, organiser l'information suivant un ordre d'importance. Cette hiérarchisation est faite sur base d'un système d'annotations ou de tags.

Si aucun renseignement n'est donné à ce sujet, le modulateur prendra comme ordre d'importance l'ordre des niveaux. Par exemple, pour le mode graphique, au sein de chaque niveau, l'ordre d'importance des objets est fixé arbitrairement sur un critère d'usage. Le modulateur découpe la page en zones (ou suivant les éléments graphiques comme des tableaux) et considère comme la plus importante la plage centrale. Ensuite, il classe les objets restants en partant du coin supérieur gauche de l'écran en essayant de distinguer des objets de navigation (ex. menu bouton...). Au dernier niveau, les objets graphiques et le texte sont classés sur le critère de la taille.



## 2. Analyse des possibilités de transformation de l'information (Etape 2)

### 2.1. Analyse de l'information

	Section	Type de terminaux visés
	<b>Objectif</b>	1. Modéliser les différents types d'information 2. Caractériser l'information de manière pertinente pour un affichage sur une IU
	<b>Entrée</b>	Site « brute »
	<b>Sortie</b>	1. Pour chaque information du site avoir un tableau reprenant l'ensemble des caractéristiques la caractérisant (cf. Tableaux de synthèse)
	<b>Connaissances</b>	1 Taxonomie des modalités de sortie (Bernsen) 2. Connaissances générales des IU

Nous essayerons de modéliser l'information (dernier niveau de décomposition) de la façon la plus complète possible tout en orientant cette taxonomie vers notre but premier qui est d'en faire un outil utile et pratique lors de la conception d'une interface-utilisateurs (IU). C'est pourquoi, nous ne nous intéresserons pas au contenu de l'information mais bien à la forme, autrement dit, à la façon de la représenter (*output*). Les modalités de représentation de l'information ont deux grands types de propriétés : les propriétés déclaratives et fonctionnelles. Les **propriétés déclaratives** sont les propriétés qui définissent les modalités parce qu'elles précisent la façon dont l'information va être représentée. Cette définition se fait suivant trois dimensions : la définition de l'objet en tant que tel (*caractérisation*), sa situation par rapport aux autres objets (*arrangement*) et par rapport à un référent physique (*positionnement*). Les **propriétés fonctionnelles**, quant à elles, caractérisent les modalités et montrent si elles sont adaptées ou non à un certain type d'information et, le cas échéant, posent les conditions pour qu'elles le soient. C'est ce que l'on appelle le mapping de l'information (*mapping information*). Par exemple un « bip » (modalité arbitraire acoustique) peu être adéquat pour signaler une fonction dans un environnement calme, ce qui n'est pas le cas dans un milieu bruyant où un vibreur serait plus approprié (modalité arbitraire tactile). Nous y reviendrons dans le chapitre concernant les contraintes dues à l'usage du terminal.



### 2.1.1. Propriétés déclaratives de l'information: Taxonomie des « modalités de sortie »<sup>38</sup>

La taxonomie des modalités (unimodale) de sortie est directement inspirée des travaux de Bernsen<sup>39</sup> (1994) basés sur les caractéristiques permanentes des signes des ordinateurs. Cela entre dans le schéma que Bernsen appelle « Modality Theory » (Bernsen 1993) et qui inclut une taxonomie des modalités d'entrée et une méthode pour appliquer cette taxonomie aux interfaces-utilisateurs (IU) actuelles (Mapping Method). Il est clair que cette théorie utilise une approche représentative et centrée sur les interactions homme-ordinateur (Human Computer Interaction = HCI). Elle est duale parce qu'elle fait une distinction entre les modalités d'entrée et de sortie.

Etablir des bases conceptuelles et taxonomiques pour analyser tout type particulier de *représentation de sortie* (output représentation), uni- ou multimodale soulève deux problèmes : le domaine de complexité et la terminologie. Etant donné le nombre presque illimité de combinaisons possibles entre les différentes modalités de sortie à disposition des programmeurs d'interface, la théorie ne peut répertorier chacune d'elles. C'est pourquoi, la théorie adopte une approche générative basée sur les éléments simples, ce qui permet d'avoir des principes par lesquels n'importe quelle combinaison peut être analysée en cas de nécessité. Deuxièmement, la terminologie dans ce domaine étant confuse, il est indispensable d'établir une terminologie robuste, conceptuellement claire et intuitivement acceptable. Si nous sommes d'accord que, par exemple, des tables, des bips ou un langage (naturel) écrit ou parlé, peuvent être pris comme « modalités », alors l'acceptabilité intuitive de l'approche présentée ci-dessous semble être assurée. Il est à noter que les « modalités » sont prises au sens de « moyen de représentation » et non comme en psychologie, au sens de « modalité sensitive ».

La taxonomie des modalités de sortie unimodales (Bernsen 1994) a été générée à partir d'un ensemble de *caractéristiques de base* (déclaratives). De fait, chaque modalité générée est dite

---

<sup>38</sup> «Taxonomy of Output Modalities» – Bernsen (1993-1994)

<sup>39</sup> Multimodal Systems Group at the Center of Cognitive Science, Roskilde University

« pure » ou unimodale et est définie par l'ensemble restreint et unique de ses « *caractéristiques de base* ». Il est donc possible de définir un ensemble limité de modalités à partir duquel n'importe quelle modalité de sortie ou combinaison de modalités de sortie puisse être générée et analysée. Ce qui définit une représentation multimodale par un ensemble de représentations unimodales.

### 2.1.1.1. Définition des propriétés de base

#### a) Propriétés de base

##### *Linguistique (vs. non-linguistique)*

Les représentations linguistiques sont basées sur un système de significations qui est à la fois syntactique, sémantique et pragmatique (ex. les langues). Il est à remarquer que les représentations linguistiques présentent un manque de spécificité (caractéristique des représentations analogues). En effet, le type linguistique présente un certain degré d'abstraction qui a pour résultat de permettre beaucoup d'interprétations quant au contenu transmis de cette façon. C'est pour cette raison que pour les « interactions homme-machine », où l'on ne peut laisser une trop grande part à l'interprétation, il est conseillé de

Ex.: *Portrait d'une jeune fille portant un ruban bleu dans les cheveux. Tableau de Vermeer* → ???



fig.14

**Règle 3. Toujours illustrer une représentation linguistique par une représentation analogique (ex. donner la description d'un tableau, et en montrer une reprographie) et inversement.** (cf. infra)

##### *Analogique (vs. non-analogique)*

Les représentations analogiques sont caractérisées par une forte similitude avec ce qu'elles représentent. Elles sont aussi appelées des représentations « isomorphiques » ou « iconiques ». Elles peuvent être graphiques (ex. photo, film), aussi bien qu'acoustiques (ex. enregistrement de bruit), tactiles (ex. illustration en relief, masque), statiques ou dynamiques. Bien que les représentations analogiques soient très spécifiques, elles manquent de pouvoir d'interprétation. En d'autres mots, si l'on voit une image, aussi fidèle qu'elle soit, on ne sait rien sur les circonstances de la



→ *Qui est-ce ?*  
→ *Le Peintre ?*  
→ *Quelle époque ?*

Tableau de Vermeer (1632-1675) – « La laitière »

fig.15

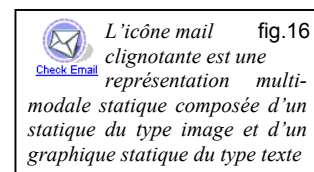
situation qu'elle représente. On peut en conclure que l'un des usages de base du langage est *d'annoter* les représentations analogiques, et l'une des fonctions de base des représentations analogiques est *d'illustrer* le texte linguistique (*exemple, fig.15*).

### ***Arbitraire (vs. non-arbitraire)***

La distinction entre représentation arbitraire et non-arbitraire marque la différence entre les représentations externes qui, pour remplir leur fonction représentative, se réfèrent à des systèmes de signes ou de représentations existantes et celles qui ne le font pas. On peut remarquer que cette distinction tend à être négligée car souvent elle correspond avec la distinction entre analogique et non-analogique. Néanmoins, cela ne change en rien la typologie, il y a des représentations qui, à l'origine, étaient arbitraires et qui, par la suite, ont pris un usage commun et, par-là même, sont devenues non-arbitraires (ex. panneaux routiers).

### **b) Etat de représentation : statique vs. dynamique**

Toute modalité unimodale est soit statique, soit dynamique, ce qui se définit, pour l'utilisateur, en fonction d'une liberté d'examen ou d'analyse dans le temps. Ce qui signifie que l'information est constante dans le temps et que l'utilisateur peut à sa guise y avoir accès, et ce de façon aléatoire. Il faut donc déterminer s'il y a ou non un



L'icône mail clignotante est une représentation multimodale statique composée d'un statique du type image et d'un graphique statique du type texte

changement physique perceptible et définitif de l'information dans le temps. On en conclut que, dans cette typologie, une icône qui clignote est considérée comme statique (*ex. fig.16*).

### **c) Modes ou moyens de représenter l'information**

Entre tous les moyens de représenter une information, les modes graphiques, acoustiques et tactiles (du toucher) ont semblé les plus pertinents en ce qui concerne le design d'IU ; la technologie ne nous permettant pas encore de considérer les modes relatifs à l'odorat et au goût. Le mode d'une information unimodale est d'une importance fondamentale pour son adaptation d'une information à une interface donnée. Le fait que les modes disposent de différents canaux d'informations (*information channels*) est primordial pour l'usage de l'information, de l'interface et par conséquent du terminal. Chaque mode de sortie (*output medium*) est défini de façon unique par un certain nombre de canaux d'informations. Un canal d'information est un aspect humainement perceptible du mode qui sert à transmettre

l'information.

### ***Mode graphique***<sup>40</sup>

Depuis toujours, l'homme s'est servi du mode graphique ou visuel pour représenter l'information (dessins des cavernes, hiéroglyphes,...). Il n'est dès lors pas étonnant que les modes graphiques (linguistiques, analogues, arbitraires et explicites) de type statique aient été étudiés de façon plus détaillée. Les canaux d'informations sont : le trait, la taille (longueur, largeur, profondeur), la texture, la résolution, le contraste, la couleur ou l'échelle de gris, la luminosité, la saturation, la teinte, la position, l'orientation, la perspective, les propriétés spatiales, les changements de propriétés répétés ou non, le mouvement, le déplacement (relatif à l'observateur) et l'ordre d'apparition.

### ***Mode Acoustique***<sup>41</sup>

Jusqu'aux récentes avancées technologiques, le mode acoustique n'avait qu'un rôle limité dans l'HCI<sup>42</sup>, contrairement aux interactions homme-homme où ce mode de transmission d'information est omniprésent et tient une place centrale. Dans beaucoup de cas, la modalité acoustique donne une alternative ou un supplément à la modalité graphique. Par exemple, l'ajout d'une image sonore à une image graphique augmente fortement la réalité (virtuelle) de l'événement représenté (ex. film d'un lancement de fusée (cf. fig.17)). De plus, le mode sonore présente deux avantages sur les modes graphiques et tactiles. Premièrement, ce mode permet à l'utilisateur d'identifier et de surveiller la source d'information quel que soit l'endroit où elle est placée. En plus, il permet de distinguer, de contrôler et d'être attentif simultanément à plusieurs sources sonores, à condition qu'elles aient des paramètres différents. Bien que le potentiel du mode acoustique soit loin d'être pleinement exploré, le langage acoustique dynamique (parole) apparaît déjà comme un sérieux rival de l'interface-



fig.17

<sup>40</sup> Graphic Modalities

<sup>41</sup> Acoustic Modalities

<sup>42</sup> Human Computer Interaction = interaction homme-machine

utilisateur graphique (ex. voicehtml, IBM voice speech, Lernhout&Houspie...).

Canaux d'informations : intensité, timbre, rythme, durée, hauteur (pitch), ordre des séquences dans le temps, localisation de la source. La modalité acoustique linguistique a comme canaux additionnels : la qualité de la voix, l'intonation, accent, le dialecte, la personnalité, le stress.

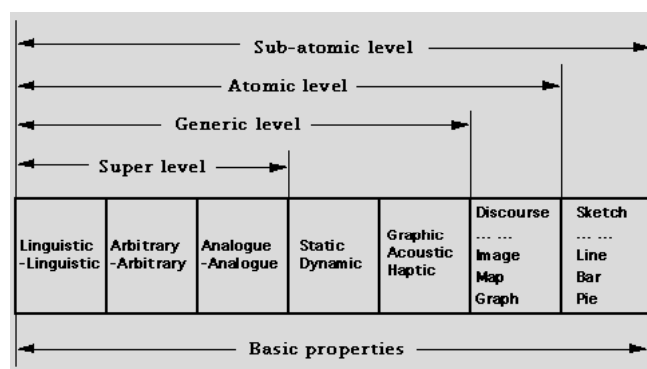
### ***Mode tactile***<sup>43</sup>

Ce mode d'interaction HCI, bien qu'il soit fréquemment utilisé comme moyen d'entrer ou d'encoder l'information (input modalities (ex. clavier, écran tactile, stylet...), il reste faible, voir inexistant dans les interfaces-utilisateurs (*output modalities* (ex. écran braille, combinaison virtuelle, joystick vibrant...)). Ceci est dû au fait que, d'un côté, la technique n'est pas encore suffisamment avancée, et que, de l'autre, les représentations tactiles sont inférieures aux représentations graphiques, aussi bien en termes de densité de l'information qu'en termes de vitesse d'acquisition. Néanmoins, ce type de mode est d'une grande utilité pour augmenter l'accessibilité à l'information des personnes malvoyantes et, de manière générale, apporte beaucoup de réalisme et de complétude à l'information.

Canaux d'informations : les bords, la taille (longueur, largeur, profondeur), la texture (surface, qualité), la position, l'orientation, la température, la pression, le voltage, l'arrangement dans l'espace, le mouvement, la répétitivité ou non dans le changement des propriétés, l'ordre temporel.

#### **2.1.1.2. Représentation des niveaux**

Afin de résoudre le problème de complexité, la taxonomie est organisée en quatre niveaux de hiérarchie, correspondant chacun à un niveau d'abstraction plus ou moins élevé. On distingue les niveaux *supérieur*, *génériques*, *atomique* et *sub-atomique*.



##### **a) Niveau supérieur**

C'est le niveau qui a le degré d'abstraction le plus élevé. Les modalités à ce niveau sont

<sup>43</sup> Haptic modalities

définies par la combinaison de trois ensembles de propriétés de base : *linguistiques/non-linguistiques*, *analogiques/non-analogiques*, *arbitraires/non-arbitraires*. Ce qui amène quatre combinaisons de propriétés de base qui définissent le niveau supérieur et qui sont : les modalités linguistiques, les modalités de structure explicite, les modalités arbitraires et les modalités analogiques. Par exemple, les modalités linguistiques sont définies par le panier suivant : linguistique, non-arbitraire et non-analogique. Les modalités linguistiques sont non-arbitraires parce qu'elles sont basées sur un système de significations existant et elles sont non-analogiques car le langage n'a pas de relation primaire avec ce qu'il représente. Dans une notation plus simplifiée, le profil des modalités linguistiques s'écrit <li,-ar,-an>. Les profils pour la structure explicite, l'arbitraire et l'analogique sont respectivement : <-li,-ar,-an>, <-li,ar,-an> et <-li,-ar,an>.

### **b) Niveau générique**

Du niveau supérieur au niveau générique, la taxonomie se différencie en distinguant les modes et les états de représentation. Le *mode* détermine la façon dont une modalité se traduit physiquement, alors que les *états de représentation* spécifient si oui ou non la modalité permet une liberté dans le temps dans la façon de percevoir l'information. La théorie modale de Bernsen fait une distinction entre les *modes d'expression* graphiques (graphics), sonores (sounds) et tactiles (harpics), chacun caractérisé par des ensembles distincts de qualités de perception (respectivement : visuelles, sonores et tactiles). On différencie les états selon qu'ils soient *statiques* ou *dynamiques*. Ces modes définissent l'étendue de la taxonomie. Une modalité de représentation générique pure est définie, en conséquence, par un ensemble complexe de propriétés caractérisées par un mode d'expression spécifique et un *profil* de caractéristiques de base. Par exemple, une même information linguistique peut être représentée de façon graphique, sonore ou tactile mais le choix du moyen influence fortement l'adéquation de la représentation pour une interface ou un usage donné.


Une matrice des modalités génériques pures tenant compte des propriétés de base mentionnées ci-dessus donnerait 48 combinaisons (2x2x2x2x3). Cette matrice est générée de façon mécanique, sans tenir compte de la cohérence de l'output. Si l'on tient compte des non-sens ou des redondances de certains résultats on peut procéder, sans altérer la pertinence des résultats, à une réduction pragmatique et passer de 48 à 28 modalités génériques (*cf. fig.18*).

Il est, par exemple, incohérent d'avoir une modalité qui serait analogique et arbitraire (ex. utiliser la représentation graphique d'une banane pour représenter une voiture). Tout comme d'avoir un langage arbitraire ou un élément sonore ou tactile qui serait statique. Ensuite, nous pouvons, pour une raison synthétique regrouper les modalités dynamiques et statiques pour les médias sonores et tactiles, ce qui ramène le tableau à vingt modalités (*cf. tableau de synthèse (2.1.1.3)*).

### Generic modalities

fig.18

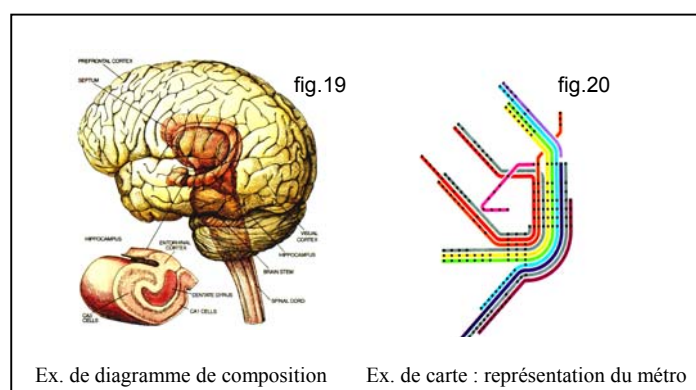
	+ Li								- Li							
	+ An				- An				+ An				- An			
	+ Ar		- Ar		+ Ar		- Ar		+ Ar		- Ar		+ Ar		- Ar	
	+ Dy	- Dy	+ Dy	- Dy	+ Dy	- Dy	+ Dy	- Dy	+ Dy	- Dy	+ Dy	- Dy	+ Dy	- Dy	+ Dy	- Dy
Gra			2	1			6	5			12-14	9-11	22	21	26	25
Sou			3				7				15-17		23		27	
Tou			4				8				18-20		24		28	

 = 1) + Linguistic <math>\diamond</math> + Arbitrary  
 2) + Analogue <math>\diamond</math> + Arbitrary  
 3) - Dynamic <math>\diamond</math> Sound/Touch

Li = linguistic  
 An = Analogue  
 Ar = arbitrary  
 Dy = dynamic vs. static  
 Gra = graphic  
 Sou = sound (acoustic)  
 Tou = touch (haptic)

### c) Niveau atomique

De façon générale, c'est le niveau le plus bas dans la taxonomie. A ce niveau, les modalités unimodales sont prévues pour être utilisées comme des « briques de construction » par les designers d'interface. Par conséquent, les propriétés ajoutées par le niveau atomique ne sont pas « transversales » dans l'arbre de représentation de la taxonomie, mais bien propres à une modalité générique. Pour la famille des *modalités linguistiques* les propriétés sont : le geste, l'écrit, le parlé, le texte, le discours, le label/mot clé et la notation. Pour les *modalités analogiques* on retrouve : l'image, le graphique, la carte (=diagramme de composition qui donne des informations géométriques de la réalité (cf. fig.19)), le diagramme de composition (=image analytique) (cf. fig.20) et le diagramme conceptuel (=représentation analogue de la décomposition analytique of abstraite d'une entité). Pour les *modalités arbitraires* et de *structure explicite* on a respectivement les caractéristiques « ad hoc » et « séparateur ».

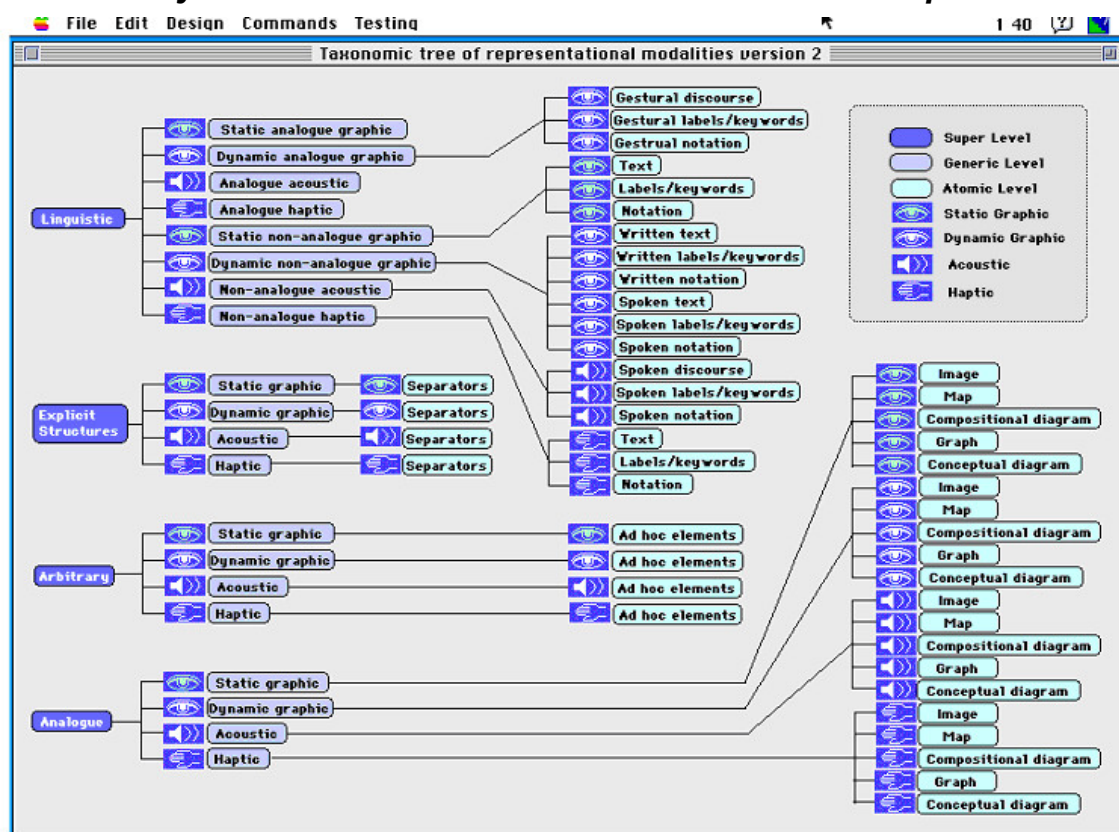




### d) Niveau sub-atomique

Bien que normalement, le niveau antérieur soit le niveau de complexité le plus avancé dans la taxonomie, il a été nécessaire d'introduire un niveau supplémentaire dans certaines parties pour une question de complétude de la théorie et un meilleur usage pratique. Cette distinction a notamment été introduite pour les graphiques statiques (<-li,-an,ar,sta,gra>) ou l'on peut rencontrer trois types de base (graphiques en ligne, en barres et en tarte (« pie »)) qui à leur tour, comme toute représentation graphique, peuvent être en une, deux ou trois dimensions (1D, 2D ou 3D)<sup>44</sup>. Ce niveau est très utile dans l'étude de l'ergonomie de l'information, mais pour les raisons expliquées antérieurement, cette problématique ne sera pas traitée dans ce travail.

### Schéma de synthèse : Taxonomie des modalités de sortie par Bensen



Ce schéma est une copie d'écran du programme de Bensen mettant en œuvre sa taxonomie.

<sup>44</sup> voir Levy, E. Schiano, D.J., Zacks, J., Tversky, B. (1995). Representing information in graphs, *Paper presented at the meeting of the Psychonomics Society* Los Angeles, CA (synthèse et représentation graphique en annexes)

### 2.1.1.3. Tableau de synthèse : Taxonomie des modalités de sortie

(Afin de respecter la nomenclature de Bernsen, la terminologie anglaise est maintenue)

SUPER LEVEL CLASSES	GENERIC UNIMODALE LEVEL		ATOMIC UNIMODALE LEVEL	SUB-ATOMIC UNIMODALE LEVEL	
		NOTATION	Well-known types		
<b>I. Linguistic modalities</b> <li,-an,-ar>	1. Static analogue sign graphic language	<li,an,-ar,sta,gra>	Hieroglyphs. Rarely used.	Included in 4 a-c and 5 a-c.	
	2.a. Static analogue sign acoustic language	<li,an,-ar,sta/dyn,aco>	Part of everyday spoken language.	Included in 6 a-c.	
	b. Dynamic analogue sign acoustic language				
	3.a. Static analogue sign haptic language	<li,an,-ar,sta/dyn,hap>	Apparently none.	Included in 7 a-c.	
	b. Dynamic analogue sign haptic language				
	4. Dynamic analogue sign graphic language	<li,an,-ar,dyn,gra>	Gestural language. Dynamic hieroglyphs would appear anachronistic.	4a. Static/dynamic gestural discourse 4b. Static/dynamic gestural labels/keywords 4c. Static/dynamic gestural notation	
	5. Static non-analogue sign graphic language	<li,-an,-ar,sta,gra>	Letters, words, numerals, other written language signs, text, logograms (e.g. arrows), special-purpose notations (e.g., programming languages, formal logic, music). Sequential, list and tabular orderings.	5a. Static graphic written text	5a1. Static graphic typed text 5a2. Static graphic hand-written text
				5b. Static graphic written labels/keywords	5b1. Static graphic typed labels/keywords 5b2. Static graphic hand-written labels/keywords
				5c. Static graphic written notation	5c1. Static graphic typed notation 5c2. Static graphic hand-written notation
	6.a. Static non-analogue sign acoustic language	<li,-an,-ar,sta/dyn,aco>	Spoken letters, words, numerals, other spoken language related sounds, discourse, list orderings.	6a. Static/dynamic spoken discourse	
	b. Dynamic non-analogue sign acoustic language			6b. Static/dynamic spoken labels/keywords 6c. Static/dynamic spoken notation	
	7.a. Static non-analogue sign haptic language	<li,-an,-ar,sta/dyn,hap>	Touch letters, numerals, words, other touch language related signs, text, list and table orderings. Example: Braille.	7a. Static/dynamic haptic text	
	b. Dynamic non-analogue sign haptic language			7b. Static/dynamic haptic labels/keywords 7c. Static/dynamic haptic notation	
	8. Dynamic non-analogue sign graphic language	<li,-an,-ar,dyn,gra>	Dynamizations of (5). Moving text, running numerical counters, digital clocks. Graphically viewed spoken language discourse (lip reading).	8a. Dynamic graphic written text	8a1. Dynamic graphic typed text 8a2. Dynamic graphic hand-written text
8b. Dynamic graphic written labels/keywords				8b1. Dynamic graphic typed labels/keywords 8b2. Dynamic graphic hand-written labels/keywords	
8c. Dynamic graphic written notation				8c1. Dynamic graphic typed notation 8c2. Dynamic graphic hand-written notation	
8d. Graphic spoken text or discourse					
8e. Graphic spoken labels/keywords					
8f. Graphic spoken notation					

<b>II.</b> <b>Analogue</b>  <-li,an,-ar>	9. Static analogue graphics	<-li,an,-ar,sta,gra>	Pure photographs, naturalistic drawings, holograms	9a. Static graphic images	9a1. real-world picture
			Pure diagrams, maps, cartoons. Sequential, list and tabular orderings.	9b. Static graphic maps	9a2. non-real-world picture
				9c. Static graphic compositional diagrams	
				9e. Static graphic conceptual diagrams	
	1D, 2D or 3D graph space containing geometrical forms. Pure charts (dot charts, bar charts, pie charts, etc.). Non- iconic use difficult due to the absence of linguistic annotation.	9d. Static graphic graphs	9d1. Line graphs		
			9d2. Bar graphs		
			9d3. Pie graphs		
	12. Dynamic analogue graphics	<-li,an,-ar,dyn,gra>	Pure movies, videos, realistic animations. cartoons	12a. Dynamic graphic images	cf. 9a1 ant 9a2
			Pure animated maps,	12b. Dynamic graphic maps	
			Pure graphs (see 11) evolving in graph space. Sequential, list and table orderings possible. 1D, 2D or 3D spatial.	12d. Dynamic graphic graphs	
			Pure animated diagrams, sequential, list and table orderings. Pure standard animations.	12c. Dynamic graphic compositional diagrams	
		12e. Dynamic graphic conceptual diagrams			
10.a. Static analogue acoustics	<-li,an,-ar,sta/dyn,aco>	Sound 'pictures'. Single sounds, sound sequences. List ordering possible.	10a. Static/dynamic acoustic images		
b. Dynamic analogue acoustics		Apparently none, but many possibilities, synthetic or manipulated, exist. Music?	10b. Static/dynamic acoustic maps		
		E.g. Geiger counters.	10c. Static/dynamic acoustic compositional diagrams		
		10e. Static/dynamic acoustic conceptual diagrams			
		10d. Static/dynamic acoustic graphs			
11.a. Static analogue haptics	<-li,an,-ar,sta/dyn,hap>	Single touch representations, touch sequences.	11a. Static/dynamic haptic images		
b. Dynamic analogue haptics		Apparently none, but many possibilities exist.	11b. Static/dynamic haptic maps		
			11c. Static/dynamic haptic compositional diagrams		
			11e. Static/dynamic haptic conceptual diagrams		
	1D, 2D or 3D graph space containing geometrical forms. Pure charts (dot charts, bar charts, pie charts, etc.).	11d. Static/dynamic haptic graphs			

<b>III. Arbitrary</b> <-li,-an,ar>	13. Arbitrary static graphics	<-li,-an,ar,sta,gra>	Diagrams consisting of geometrical elements. Sequential, list and tabular orderings. Non-iconic use difficult due to the absence of linguistic annotation.		
	16. Dynamic arbitrary graphics	<-li,-an,ar,dyn,gra>	Diagrams consisting of geometrical elements. Non-iconic use difficult due to the absence of linguistic annotation.		
	14.a. Arbitrary static acoustics	<-li,-an,ar,sta/dyn,aco>	Sound signal icons.		
	b. Dynamic arbitrary acoustics				
	15.a. Arbitrary static haptics	<-li,-an,ar,sta/dyn,hap>	Touch signals of different sorts.		
	b. Dynamic arbitrary haptics				
<b>IV. Explicit structures</b> <-li,-an,-ar>	17. Static graphic structures	<-li,-an,-ar,sta,gra>	Form fields, frames, table grids, line separations, trees, windows, bars. Sequential, list and tabular orderings. Non-iconic use difficult due to the absence of linguistic annotation.		
	18.a. Static acoustic structures	<-li,-an,-ar,sta/dyn,aco>	Apparently none.		
	b. Dynamic acoustic structures				
	19.a. Static haptic structures	<-li,-an,-ar,sta/dyn,hap>	Form fields, frames, grids, line separations, trees.		
	b. Dynamic haptic structures				
20. Dynamic graphic structures	<-li,-an,-ar,dyn,gra>	Dynamic frames, windows, scroll bars. Non-iconic use difficult due to the absence of linguistic annotation.			
		<b>NOTATION</b>	<b>Well-known types</b>		
<b>SUPER LEVEL CLASSES</b>	<b>GENERIC UNIMODALE LEVEL</b>		<b>ATOMIC UNIMODALE LEVEL</b>		<b>SUB-ATOMIC UNIMODALE LEVEL</b>

### **2.1.2. Adaptation du modèle des « modalités de sortie » au modulateur**

Rappelons-nous que le but de cette première phase est de modéliser au mieux l'information « pure ». Pour ce faire, le modulateur distingue trois étapes dont les deux premières sont une application directe, à peu de choses près, du modèle de Bernsen. Il s'agit du *type* et des *caractéristiques de base* qui définissent l'information. Etant donné que les IU sur lesquelles porte notre étude sont dépendantes de terminaux « informatiques », une information, pour y être représentée, doit être traduite dans un langage qui soit compréhensible par ceux-ci. D'une manière simplifiée, on peut définir ce langage comme une traduction de l'information en code ordinateur. Dans le cas où l'on veut pouvoir distinguer une information de manière indépendante, on regroupe ce code en fichiers. Chaque fichier procède une extension ou un format qui permet au système d'exploitation du terminal de savoir quel type d'information contient le fichier et comment la décoder correctement. Etant donné qu'à chaque type d'information correspond un certain nombre de formats de fichiers, et que tous les terminaux ne supportent pas tous les formats, le *format de codage* initial est un facteur primordial dans l'analyse descriptive d'une information. Par analogie au monde réel, on pourrait comparer le format fichier au support de l'information comme par exemple une toile pour une peinture ou une feuille de papier pour un dessin. De même comme il est possible de « recopier » une oeuvre d'une toile sur une feuille de papier, il est possible de convertir un format fichier en un autre. Dans ce cas aussi, chaque « support » a ces caractéristiques et ces limites propres. Comme dans le monde réel, tout support a une taille proportionnelle à la taille ou au type d'information qui y est représentée, ce qui influence directement la commodité avec laquelle le support va être transporté. Par exemple, il est plus facile de transporter un « pins » représentant une peinture que la toile elle-même. Pour les fichiers, c'est le même principe. Mais étant donné que la vitesse est un facteur clé pour les applications on-line, la *taille du fichier* est particulièrement importante.

#### **2.1.2.1. Le type d'information - d'objet**

La taxonomie proposée par Bernsen se veut robuste, conceptuellement claire et orientée vers le designer d'IU. Néanmoins, lorsqu'il s'agit de la mettre en pratique pour des interfaces « informatiques », ce n'est plus si évident. La terminologie peut paraître lourde ou pas assez

explicite et la hiérarchisation pas toujours adéquate. C'est pourquoi nous allons réorganiser cette taxonomie en essayant quelle soit non seulement plus compréhensible mais aussi plus adaptée aux interfaces étudiées. Désormais, la principale division ne se fait plus sur la distinction entre linguistique, analogue, arbitraire et explicite mais en fonction des modes de sortie que l'on peut trouver sur un terminal. Par analogie aux cinq sens qui ont servi de base à la « modality theory », on peut représenter un même objet sur une interface graphique (l'écran), sonore (les enceintes) ou tactile (ex. vibreur, gant tactile, écran braille...). La suite de la classification à comme ligne directrice de regrouper les informations suivant les types de formats de fichiers selon lesquels elles peuvent être codées. Ce qui revient à mettre ensemble les informations interprétées de la même manière par le système d'exploitation et/ou le browser et qui, par conséquent, sont considérées, aux yeux de ceux-ci, comme identiques. Pour ce faire, nous regrouperons les éléments à identifier dans la taxonomie de Bernsen qui reste un instrument conceptuellement solide. Comme chaque mode suit sa propre logique, ils vont être analysés séparément.

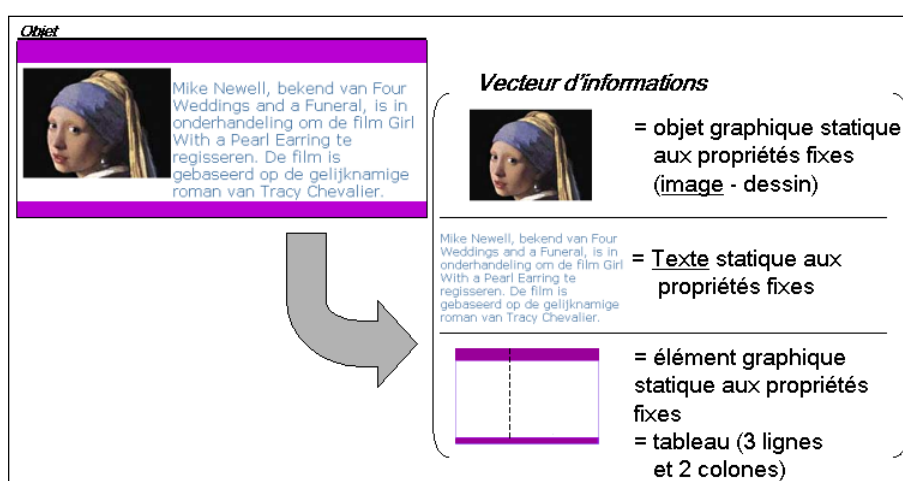
### **a) Mode Graphique**

Le mode graphique a été le premier supporté par les interfaces informatiques et reste celui dont les possibilités ont été les plus exploitées. Par conséquent, il est le plus fourni en terme de modélisation. La première distinction se fait entre les informations suivant qu'elles soient statiques ou dynamiques (pris dans le sens défini par Bernsen). Il est à noter que, du point de vue du format des fichiers supportant ce type de données, le mode statique ayant des propriétés fixes répétitives peut être aisément assimilé au mode dynamique. En effet, mis à part les fichiers propres au mode statique répétitif (ex. gif animated), cette catégorie peut être matérialisée par des fichiers de petite taille dont le format est adapté aux films (ex. avi) et qui seraient passés en boucle. Pour chacune de ces catégories, on distingue trois types de données propres au mode graphique : les objets graphiques (images, graphiques, cartes, diagrammes), le texte (texte, label/mot clé, notation) et les éléments graphiques (lignes, tableaux, fenêtres, cadres, formulaire...). Ceux-ci sont obtenus en faisant des regroupements dans la taxonomie<sup>45</sup>. Attention, dans cette taxonomie n'est considéré comme texte que ce qui a été

---

<sup>45</sup> Le détail du regroupement est repris en annexes 4

défini comme tel. Un texte qui serait défini comme une image (ex. photo d'un texte) et non comme du texte, sera considéré par le modulateur comme une image, à moins d'être reconverti par un OCR (programme de reconnaissance de caractères)<sup>46</sup>. Il est à noter que les sous-catégories qui ont été présentées entre parenthèses dans le texte, ne sont données qu'à titre d'exemple, par souci de complétude. Elles n'interviennent en rien dans le modulateur tel que présenté dans ce travail ; par contre, elles seraient pertinentes dans le cas où l'on tiendrait compte de l'ergonomie dans l'adaptation des données aux terminaux.



## b) Mode Sonore

Ici, la classification est faite suivant le mode dans lequel le son a été généré. Il peut être soit dû à un enregistrement de sons *réels*, soit généré de façon *synthétique* par l'ordinateur. Le langage de synthèse bien que généré par l'ordinateur se distingue du fait qu'il soit produit de façon indépendante par un programme qui convertit du texte en sons. Nous le considérerons donc indépendamment car sa gestion est faite de façon externe au modulateur. Précisons que ces types de sons ont leur format propre de fichier. De manière générale, les fichiers-sons résultant d'un enregistrement (ex. WAV, AU, MP3...) sont plus volumineux que les autres (ex. MIDI). Par exemple un son en WAV enregistré en 44.100kHz (fréquence), 16bit (taille d'un mot-mémoire, défini le taux d'échantillonnage d'un son), stéréo et codé en PCM, ce qui correspond à une qualité CD, prenant 172kb par seconde d'enregistrement. Un son MIDI

<sup>46</sup> Possibilités pour de futures améliorations du modèle ; ce point ne sera pas abordé ici

(synthétique) en 16 pistes (nombre de sons distinguables simultanément) ne prend quant à lui que 0,31 kb/s, soit presque 400 fois moins. Par contre, contrairement au WAV, le rendu d'un son en MIDI dépend directement des capacités de la carte son à générer des sons. Il se pourrait que dans un concerto pour piano enregistré en MIDI, si la carte ne peut générer le son du piano, celui-ci soit remplacé par de la clarinette.

### **c) Mode tactile**

Ce mode sera le moins approfondi des trois car l'interface à laquelle il fait référence (ex. écran braille, gant virtuel...) est généralement externe au terminal et dépend en principe d'un gestionnaire de périphérique qui le calibre et adapte l'information à celui-ci. Le traitement de ce type d'information se fait donc en dehors du modulateur. Il serait néanmoins possible d'en tenir compte dans des améliorations futures du modèle. On pourrait par exemple communiquer au gestionnaire du périphérique les niveaux d'importance des informations ou la façon exacte de les interpréter.

Pour ce mode on peut faire une distinction suivant le type d'interface de sortie. Pour l'instant on, a d'un côté, les interfaces tactiles « graphiques » car l'information y apparaît en relief comme sur un écran (ex. écran braille) et, de l'autre coté, les interfaces tactile « sonores » (ex. vibreur de gsm) qui émettent un type de signal défini (ex. vibration, chaleur...). Ces propriétés font que les interfaces dites « graphiques » offrent plus de possibilités en ce sens qu'elles permettent des représentations linguistiques (ex. braille) et analogues (ex. images, graphiques ou diagrammes tactiles) tandis que les interfaces « sonores » seront plus adéquates pour des signaux arbitraires (ex. alarme).

#### **2.1.2.2. Les caractéristiques de base**

Une fois le type de l'objet déterminé, il faut en décrire les caractéristiques. Les *caractéristiques de base* sont des paramètres qui définissent l'objet et la façon dont il est représenté sur l'interface. Chaque mode a des caractéristiques qui lui sont propres. De manière générale, les caractéristiques sont divisées suivant qu'elles touchent au positionnement<sup>47</sup>, à l'arrangement<sup>48</sup> ou à la caractérisation de l'objet<sup>49</sup>. Voyons en détail ces

---

<sup>47</sup> Positionnement = définition de l'objet par rapport à des repères physiques ou au browser



caractéristiques.

## a) Mode Graphique

### *Positionnement*

Etant donné que l'interface graphique des terminaux est un écran, l'information y est positionnée suivant deux dimensions : la largeur et la hauteur. Il faut désormais définir un point de repère pour le positionnement et discuter de l'unité de mesure à prendre.

Points de repère : On peut définir un objet soit par rapport aux bords de l'écran, soit par rapport à un point particulier de celui-ci, de manière générale on prend le centre de l'écran (*cf. fig.21*). Ici, nous définirons arbitrairement un objet à la distance qui le sépare des bords. Cette distance sera positive si la distance de l'objet au bord est comprise

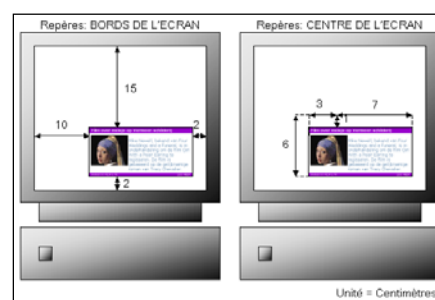


fig.21

dans l'écran et négative dans le cas contraire (*cf. fig.22*). De même, la distance entre un repère de l'interface peut se mesurer soit par rapport aux bords de l'objet, soit par rapport au centre de celui-ci. Dans cette méthode nous prendrons les mesures à partir des bords de l'objet car cela donne plus de précision quant à taille de l'objet relative à l'écran et permet de voir directement si l'objet dépasse ou non du cadre. Ayant



fig.22

défini le positionnement de la sorte, il est possible de définir des alignements de façon absolue (ex. centrage vertical, horizontal ou milieux absolus). Ces alignements sont la résultante de distances particulières par rapport aux bords. Par exemple un objet qui serait aligné au milieu absolu (centrage horizontal et vertical) serait défini par des distances identiques aux quatre bords de l'écran, et ce quelle que soient la taille de l'objet, celle de l'écran ou la distance qui sépare l'objet des bords.

<sup>48</sup> Arrangement = définition de l'objet par rapport aux autres objets

<sup>49</sup> Caractérisation = définition de l'objet en tant que tel

### **Unités de mesure**

Il existe plusieurs standards en ce qui concerne les unités de mesure:

- Pouce (Inch) = Unité de mesure anglaise qui équivaut à 2,54 cm
- Centimètre (cm) = 1/100 de mètre<sup>50</sup>
- Millimètre (mm) = 1/1000 de mètre
- Pixel (pxl) ou Point = plus petit élément de teinte homogène d'une image<sup>51</sup> ou la plus petite entité affichable sur un écran. Unité servant à la mesure de la résolution (mesure = dpi = Point/Pouce)<sup>52</sup>.
- Pica = unité de mesure typographique qui correspond à 4,217 mm
- Point Pica = 1/12 de Pica

Etant donné que le pixel est l'unité la plus précise pour mesurer une image, nous mesurerons désormais la distance en pixels.

### ***Arrangement***

C'est la façon dont les objets s'organisent les uns par rapport aux autres au sein d'un même niveau de décomposition ou avec les niveaux supérieurs. En d'autres termes, il s'agit de la distance qui sépare les objets. Cette organisation dépend fortement de la façon donc le langage de programmation organise l'information et surtout de la façon dont le browser interprète le langage. De manière générale, les éléments s'organisent en fonction du niveau de décomposition N+1. La page (par définition le niveau principal) est divisée verticalement en lignes et chaque ligne est divisée horizontalement en caractères. Sur cette trame s'organisent des objets définis par rapport à la page (ex. tableaux...). A l'intérieur de ces objets s'organisent d'autres objets (cellules) et ainsi de suite jusqu'au niveau atomique (image). La *figure 24* montre le rendu écran par browser Microsoft Explorer de l'objet, qui nous servira d'exemple tout au long du travail, et le code en HTML qu'il a été nécessaire de générer. Le but de cet exemple est de montrer comment les informations s'arrangent en HTML. A l'aide de la *figure 23* qui nous donne les fonctions des différents tags utiles à notre exemple, nous

---

<sup>50</sup> Unité SI de longueur (symb. m), égale à la longueur du trajet parcouru dans le vide par la lumière pendant une durée de 1/299 792 458 de seconde.

<sup>51</sup> Dictionnaire Larousse

<sup>52</sup> Arts Graph. En photocomposition et en P.A.O., nombre de points formés par le balayage d'un faisceau laser, sur une hauteur de 25,4 mm (1 pouce anglais), pour assurer la restitution des textes et des images = dpi (Dots Per Inch) ou ppp (Points Par Pouce) (Dictionnaire Larousse)

fig.23

pouvons conclure que les informations (une image et du texte) sont dans la ligne centrale d'un tableau, chacune dans une cellule. A l'intérieur de ces cellules, l'image est alignée à gauche tandis que le texte est aligné par défaut (à gauche).

En HTML est fait de tags qui marque le début et la fin de chaque objet.		
	Tag de but	Tag de fin d'une section
Table	<Table>	</Table>
Ligne	<TR>	</TR>
Colonne	<TD>	</TD>
Image	<img nom du fichier de l'image >	
Texte	<FONT>	</FONT>

```

...
<TABLE bgColor=white border=0 cellpadding=0 cellspacing=0 width=431>
  <TR>
    <TD bgColor=#9900cc colspan=3 height=24></TD></TR>
  <TR>
    <TD>
      <IMG align=left border=0 height=115 hspace=2 src="12591.jpg" vspace=6 width=155></TD>
    <TD bgColor=white width=421>
      <FONT color=#336699 face=arial size=2>
        <BR>Mike Newell, bekend van Four Weddings and a Funeral, is in onderhandeling om de film Girl With a Pearl Earring te regisseren. De film is gebaseerd op de gelijknamige roman van Tracy Chevalier.</BR></FONT></TD></TR>
  <TR>
    <TD bgColor=#9900cc colspan=3 height=24></TD></TR>
</TABLE>

```

2 colonnes


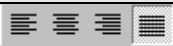




3 lignes

fig.24

En résumé, si l'on fait abstraction du langage de programmation, les objets peuvent être alignés horizontalement et/ou verticalement par rapport au niveau N+1 de la même façon que celle, vue dans le positionnement. Sachant cela, il ne reste plus qu'à positionner « physiquement » la page (le niveau hiérarchique le plus élevé). Les autres niveaux, étant donné qu'ils sont définis en fonction du niveau qui leur est supérieur, seront automatiquement adaptés. Cette adaptation peut être faite de différentes manières. Par exemple en adaptant proportionnellement l'espace entre les objets, en adaptant leur taille, en adaptant l'espace et la taille, en changeant le type de l'objet (ex. l'image est remplacée par sa description en texte), en réorganisant les objets... Cette problématique sera abordée dans les chapitres suivants.

Le cas des objets mobiles, c'est-à-dire ceux dont l'arrangement avec les autres objets bouge dans le temps, est traité de la même façon que le cas des objets statiques. En effet, il est toujours possible de délimiter le mouvement de l'objet dynamique à l'intérieur d'un cadre statique. C'est ce nouvel objet qui sera arrangé et considéré comme le plus petit niveau. Le fait qu'il soit dynamique sera repris comme une caractéristique.

Pour le texte, on distingue en plus:


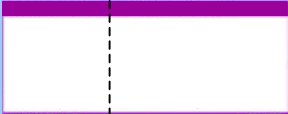

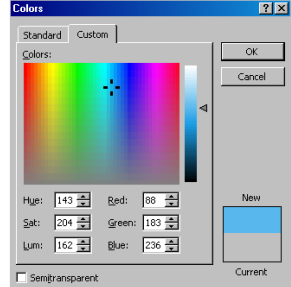
Arrangement	Possibilité	Exemple	Illustration
alignement des mots sur la ligne			
horizontal	droite, centre, gauche, justifier		gauche
vertical	distances en points		0
espacement entre			
les lettres	distances en points		0
les lignes	distances en interligne relatif à la taille de la police. Si 0,1,2= sans, simple ou double		1
les paragraphes	distances en points avant et après le §		0 et 0

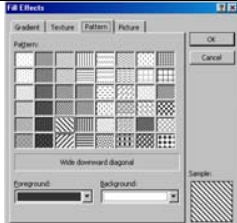
### *Caractérisation*

Les caractéristiques sont tous les facteurs qui permettent de paramétrer un objet (dans ce cas graphique) dans le but de pouvoir l'« afficher » sur une interface. Dans la mesure où c'est sur ces facteurs que l'on joue pour dégrader ou adapter un objet à une interface, il est primordial que ceux-ci soient bien définis. Les caractéristiques s'organisent suivant le type d'objet (objets graphiques ou éléments graphiques du système ou texte) et leur état (statique ou dynamique). Les caractéristiques sont définies pour un état statique. Dans le cas d'un objet dynamique, il faut rajouter aux caractéristiques statiques, les caractéristiques propres au mode dynamique. La liste des caractéristiques présentées ci-dessous se veut la plus exhaustive possible. Les caractéristiques ont été sélectionnées sur base des « canaux d'information » de la taxonomie de Bernsen, et suivant la manière dont les programmes<sup>53</sup> calibrent généralement

<sup>53</sup> Les outils de calibrage utilisés pour définir ces caractéristiques ont été principalement ceux de l'office de

l'information graphique. Les caractéristiques en italiques sont celles qui ont été définies par Bernsen comme étant importantes mais qui ont trait à l'ergonomie et interviennent lors de l'édition de l'image; le modulateur, tel que présenté dans cette version, n'en tient donc pas compte.

Caractéristiques	Définitions			Illustrations		
<b>Statique</b>						
Objet/Elément graphique						Mike Newell, bekend van Four Weddings and a Funeral, is in onderhandeling om de film Girl With a Pearl Earring te regisseren. De film is gebaseerd op de gelijknamige roman van Tracy Chevalier.
Taille Source	Définit la taille de l'objet tel qu'il est dans le fichier source (Largeur x Hauteur)	Pxls x Pxls		155 x 115	428 x 284,5	
Taille Objet	Définit la taille de l'objet tel qu'il est présenté dans l'interface de base	En Pxls ou De manière relative (%)		67% = 104 x 77	72 % = 308 x 204	
Résolution	Détermine la <u>qualité</u> du grain de l'image	LxH Dpi (pixels par pouce)		300 x 300	96 x 96	
Orientation	Disposition de l'image	Paysage : L>H Portrait : H>L		paysage	paysage	paysage
Perspective		1,2ou3D				
<b>Couleurs</b>						
Palette	Nombre de couleur	Bit = 2 <sup>X</sup>	>16bit (couleurs réelles) 8 bit (256c) 4bit (16gris / 16 couleurs) 1bit (N&B)	24 bit	2 bit	24 bit
Couleurs	La définition des couleurs dépend du codage employé (RGB(Red Green Blue)...). De façon générale on emploie le RGB, la couleur est alors définie soit par la combinaison des trois couleurs (Rouge-Vert-Bleu) soit par trois facteurs qui sont la teinte, la saturation et la luminosité. Pour des couleurs en 24bit ces facteurs varient de 0 à 255 <b>→ type de représentation des couleurs</b>			True color RGB	Palettized En RGB cela donnerait: Blanc (255;255;255) Noir (0;0;0) Mauve (153;0;153)	RGB

<i>Texture</i>	<i>Remplissage des zones planes</i>					
<i>Contraste</i>	<i>Accentuation de certaines couleurs afin de montrer l'opposition entre des formes qui sont mises en valeur par leur juxtaposition. Notion très utile pour l'ergonomie et l'accessibilité des objets graphiques.</i>					
Texte (les caractéristiques du texte peuvent être utiles pour hiérarchiser le contenu de l'information mais aussi pour l'accessibilité ou l'ergonomie)						
Police	Type de calligraphie utilisée	Nom de la police	Arial Bauhous Freestyle script Courier, ...			Arial
Taille	Taille de la police.	points	7, 10, <b>16</b> , ...			14 en points (Taille 2 en HTML)
Style		Nom du style	Normal, <b>Gras</b> <i>italique</i> , <b>gras italique</b>			normal
Effets		Nom de l'effet	<u>Souligné</u> , <u>Barre</u> , <sup>Exposant</sup> , outline, double souligné, ...			aucun
Couleur	cf. supra.	Définie par rapport à un type de représentation	cf. supra			R : 102 V : 153 B : 255
<b>Dynamique</b>						
Vitesse de cadencement ou de déplacement des images	Le rendu du mouvement est créé par une suite d'images	Images par seconde	Pour avoir la fluidité de l'image (ex. video-téléphonie), il faut 24 images par seconde			
<i>Ordre d'apparition</i>	Ordre dans lequel les images sont agencées					
<i>Déplacement (relatif à l'observateur)</i>			Fixe (=film) Disparition, Défilement, ...			

## b) Mode sonore

### *Positionnement*

Lorsqu'on parle d'interface sonore, le positionnement tel qu'il a été défini pour le mode graphique est beaucoup plus difficile à concevoir. Etant donné que le son<sup>54</sup> est une onde et que le support est l'air, on se rend bien compte que définir le positionnement par rapport au support est peu concevable. Il faut donc se tourner vers d'autres notions comme le positionnement par rapport à sa source. A partir du moment où l'on a plus de deux sources distinctes, il est possible, en jouant sur des différences de signaux, de créer un environnement sonore qui donne l'impression d'un positionnement du son dans l'espace. Différentes techniques de cryptage existent, allant de la plus simple stéréophonie (droite – gauche) à des techniques beaucoup plus complexe comme le son THX. Retenons simplement que l'environnement sonore apporte beaucoup d'informations et est indéniablement complémentaire au mode graphique surtout en ce qui concerne le réalisme de l'information ou la mise en condition de l'utilisateur. Par exemple, le simple fait de doubler un objet « éloigné » par un bruit de faible volume renforce l'impression de la distance.

### *Arrangement*

Il s'agit, ici, de la façon dont les différents sons d'un même fichier ou d'une même séquence sont organisés les uns par rapport aux autres. Cette organisation est un arrangement temporel et/ou spatial (voir positionnement) fait sur base des propriétés du son (cf supra) : le timbre, la hauteur<sup>55</sup>, l'intensité et la durée (= le rythme).

Pour conclure ces deux premiers aspects, on peut se rendre compte que manipuler des sons

---

<sup>54</sup> **Définition** (Larousse, Biblioplus) : Emis par les corps animés d'un mouvement vibratoire, le son se propage sous forme d'ondes mécaniques susceptibles de subir des réflexions (phénomène d'écho), des réfractions (transmission à travers une paroi) et des interférences (renforcement ou annulation de l'intensité sonore entre deux sources identiques émettant en phase et à la même fréquence).

La vitesse de propagation des sons audibles, d'une fréquence comprise entre 15 Hz (son grave) et 15 kHz (son aigu), est d'environ 340 m/s dans l'air, de 1 425 m/s dans l'eau et de 5 000 m/s dans l'acier. Un son est essentiellement **caractérisé par** sa **hauteur** (liée à sa fréquence), son **intensité** (liée à l'amplitude des vibrations sonores) et son **timbre**, qui dépend des intensités relatives des différents sons harmoniques qui le composent.

<sup>55</sup> La hauteur est fonction de la fréquence de l'onde. Par exemple, le « LA » standard est une vibration dont la fréquence est de 440Hz



n'est pas une chose simple. Modifier l'arrangement ou le positionnement de séquences sonores déjà enregistrées nécessite de tenir en compte d'énormément de facteurs. Cela rend difficile toute automatisation, demande des logiciels puissants et ne pourra, enfin, qu'être approximatif quant au résultat. Sans oublier que le retraitement des fichiers de sons analogiques, par opposition aux sons synthétiques, est limité au travail sur les paramètres de l'onde: la fréquence (vitesse/hauteur) et l'intensité (volume). De plus, le rendu sonore, y compris les possibilités de positionnements simulés (ex. Cristal<sup>®</sup>SoundFusion<sup>™</sup> interactive 3D audio), est fonction de la carte-son du terminal et des logiciels qui la gèrent. C'est pourquoi nous n'allons pas opérer de transformations sur les éléments sonores. La seule chose que nous pouvons faire est de voir si le fait d'ajouter une information sonore est pertinent en fonction des différentes contraintes (terminal, utilisateur,...) et si, au vu de ces contraintes, cela est possible. De plus, le modulateur pourrait, pour certains fichiers procéder à des conversions ou des compressions simples pour gagner de la vitesse lors du transfert. Par exemple, convertir un fichier wav en MP3 permet de réduire par cinq la taille du fichier. Toutefois, cela nécessite de s'assurer que le terminal puisse lire ce format.

Pour finir, rappelons que doubler l'information graphique par une information sonore apporte un complément indéniable et est particulièrement utile aux utilisateurs souffrant de problèmes d'accessibilité. Mais, en ce qui concerne la façon juste de le faire, nous devons nous référer à des travaux portant sur l'ergonomie ou l'accessibilité de l'information<sup>56</sup>. La seule recommandation que nous pouvons faire est de

---

<sup>56</sup> voir à ce sujet :

Guillaume Calvet, Julien Kahn, Moustapha Zouinar, Pascal Salembier, Jean-Christophe Briois, Laurence Nigay, Gaëtan Rey, Laurence Pasqualetti, *Etude empirique de l'usage de la multi-modalité sur un ordinateur de poche in "Interaction sans frontières"*, Actes de la Conférence conjointe AFIHM-BCS sur l'Interaction Homme-Machine IHM-HCI'2001 (Lille, 10-14 septembre 2001), Volume II, Jean Vanderdonckt, Ann Blandford & Alain Derycke (éds.), Cépaduès-Editions, Toulouse, 2001. pp. 5-8

Marcus Nyberg (Ericsson Research), Staffan Bjork (PLAY, Interactive Institute), Mikael Goldstein (Ericsson Research) and Johan Redstrom (PLAY, Interactive Institute), *Handheld Applications Design: Merging Information Appliances Without Affecting Usability*

Eija Kaasinen, Juha Kolari and Timo Laakko, *Mobile-Transparent Access to Web Services*

Ces deux dernière reference sont des exposés de la : in EIGHTH IFIP TC.13 CONFERENCE ON HUMAN-COMPUTER INTERACTION Tokyo, JAPAN July 9-13, 2001, Waseda University Conference Centre, Shinjuku

***Règle 4. Organiser et doubler l'information de façon hiérarchique suivant le degré d'importance et, si possible, par plusieurs types de fichiers (sons réels (wav, au,...) et synthétiques (midi). De cette façon, le modulateur pourra donner des préférences quant au fichier à utiliser pour représenter une information suivant le terminal visé.***

Par exemple, on pourrait imaginer que le texte d'un discours puisse être « lu » soit en utilisant la bande originale (volumineux) soit par une voix de synthèse (rapide mais moins réaliste). Dans ce dernier cas, le modulateur pourrait soit donner le fichier soit le faire lire à un programme de lecture, situé au niveau du terminal, en lui fournissant des informations caractérisant la voix à employer pour rendre la lecture la plus réaliste possible.

***Règle 5. Pour tout texte donner des informations de lecture pour la voix de synthèse. Dans le cas où il n'y a pas d'information, c'est la voix « standard » définie par l'utilisateur (si le programme de lecture est au niveau du terminal) ou le designer de l'interface (si la lecture se fait en amont).***

### ***Caractérisation***

Comme expliqué dans la conclusion des deux points précédents, le modulateur ne retire pas, ou très peu, les sons et les considère plutôt comme des fichiers dont il peut, dans certains cas, transformer le format ou pour lesquels il peut donner des indications de lecture au browser du terminal. C'est pourquoi la majorité des caractéristiques présentées ici sont données à titre indicatif. Seule celles qui ne sont pas en italiques sont considérées par le modulateur. Dans cette analyse nous distinguons les caractéristiques propres au son de celles de la voix pour le langage de synthèse. Bien que les caractéristiques de la voix puissent être définies en fonction des caractéristiques du son, elles ont des dénominations qui lui sont propres. Par exemple, le timbre de voix (homme-femme/grave-aigu) peut être vu comme une modulation de la fréquence.

Le son se définit selon sa durée (= rythme) *son timbre* et *les caractéristiques de l'onde*, à savoir : *la fréquence* (= *timbre ou hauteur*) et *l'intensité* (= *volume*).

Le langage de synthèse se définit par :

- La langue employée : les mots ne sont pas prononcés de la même façon suivant les langues. Par exemple, « potato »<sup>57</sup> se prononce en anglais [pe/teite†] tandis qu'en français il serait prononcé [potato].
- Le timbre ou la personnalité: joue sur la hauteur de la voix, permet de définir une échelle de voix allant d'une voix d'homme (grave) à celle d'une femme (aiguë).
- Intonation: Mouvement mélodique de la parole, caractérisé par des variations de hauteur des voyelles et qui joue un rôle important dans l'organisation de l'énoncé oral<sup>58</sup>. Se définit sur une échelle allant de fort à faible.
- Vitesse : Rapidité avec laquelle les mots sont prononcés et espacés dans le temps. Cela se traduit par l'impression que la prononciation est fort liée si la vitesse est rapide ou hachée dans le cas contraire.

### **c) Mode tactile**

Comme mentionné auparavant, les interfaces tactiles (ex. écran braille, gant virtuel...) sont généralement externes au terminal et dépendent en principe d'un gestionnaire de périphérique qui le calibre et adapte l'information. Le traitement de ce type d'information se fait donc en dehors du modulateur. C'est pourquoi nous n'en tiendrons pas compte par la suite.

#### ***Positionnement - Arrangement***

Pour ce mode, les paramètres sont les plus difficiles à définir. On pourrait rappeler la distinction faite entre les interfaces tactiles « graphiques » où l'information apparaît en relief (ex. écran braille), et les interfaces tactiles « sonores » (ex. vibreur de gsm) qui émettent un type de signal défini (ex. vibrations, chaleur...). Il pourrait donc être possible de transposer les positionnements et arrangements des modes graphiques et sonores aux types d'interfaces

---

<sup>57</sup> signifie pomme de terre en anglais

<sup>58</sup> définition du dictionnaire Larousse

tactiles correspondantes. Mais nous limiterons là dans cette analyse.

### ***Caractérisation***

A titre indicatif, voici les caractéristiques de base évoquées par Bernsen.

Pour les interfaces tactiles « graphiques » : longueur, largeur, profondeur/relief, texture (=qualité de la surface), orientation et vitesse d'apparition des signaux.

Pour les interfaces tactiles « sonores » : température, pression, voltage, vitesse, changement répétitif ou non des propriétés.

#### **2.1.2.3. Fichier**

##### **a) Format**

Chaque format de fichier correspond à un type d'information. Mais une information peut être représentée par différents types de fichiers. La liste présentée dans de tableau récapitulatif n'est pas exhaustive, elle reprend les principaux formats suivant les types d'information et est à considérer à titre d'exemple.

##### **b) Mode ou niveau de compression**

Au sein d'un même format, l'information peut être codifiée différemment en fonction de la taille ou de la qualité désirée. Si l'on veut retravailler un fichier ou traduire son format, c'est une information qu'il est nécessaire de connaître. Il existe des modes de compression pour presque tout type de fichier qu'il soit graphique (gif, JPEG...) ou sonore. Par exemple, il est possible d'enregistrer un fichier wav avec les codages suivants : ACELP.net, CITT A-low, CITT u-low, IMA ADPCM, Lernout&Hauspie CELP ou SBC, Microsoft ADPCM, Microsoft G.723.1, MPEG layer-3, PCM...


##### **c) Taille**

Etant donné que, dans le transfert de données, la vitesse est un facteur clé, nous nous devons d'en tenir compte lors de l'analyse des informations de base.


Dans l'exemple traité, l'image de la fille est un fichier appelé *fille* de type *.jpeg* et ayant une taille de *16Kbits*. Ce fichier n'a subi aucune compression.

### 2.1.3. Conclusion

*Règle 6. Lors de la programmation de la page source, toute information doit être présentée suivant les trois modes (graphique, sonore et tactile (si possible)) en la dédoublant de sorte que toute information linguistique soit annotée/illustrée par une information analogique ou arbitraire, et inversement. (ex.:annoter/expliquer une image avec du texte, pour le cas d'une alarme il faudrait la représenter par une image, un son, un signal tactile (vibrations) et l'accompagner d'une explication écrite et « parlée »)*

Taxonomie de l'information	Caractéristiques de base*	Fichier		
<i>(Toute info. peut être représentée suivant ces trois modes)</i>	<i>(information cannels)</i>			
Tableau de synthèse		Format	Compression	Taille
			Mode/Niveau	(kbit)
<p><u>Statique</u></p> <p><i>propriétés fixes</i></p> <ul style="list-style-type: none"> <li>▪ Objets graphiques                             <ul style="list-style-type: none"> <li>- Image (réelle (photo) ou non (dessin))</li> <li>- graphique (1,2,3D - points;lignes;barres;parts)</li> <li>- carte</li> <li>- diagramme                                     <ul style="list-style-type: none"> <li>» de composition</li> <li>» conceptuel</li> </ul> </li> </ul> </li> <li>▪ Texte                             <ul style="list-style-type: none"> <li>- texte</li> <li>- label / mot clé</li> <li>- notation</li> </ul> </li> <li>• Eléments graphiques (ex ligne, traits, tableau, fenêtre,...)</li> </ul>	<p>Position (p/r à un repère physique)</p> <p>Arrangement (p/r aux autres objets du même niveau)</p> <p>Caractérisation – définition de l'objet en tant que tel</p> <p><u>Général</u></p> <p>Taille source</p> <p>Taille objet</p> <p>Résolution</p> <p><i>orientation (portrait/paysage,...)</i></p> <p><i>perspective (1,2ou3D)</i></p> <p><u>Couleur</u></p> <p>Palette (&gt;16bit (couleurs réelles); 8 bit (256c); 4bit (16gris); 1bit (N&amp;B))</p> <p>Type de représentation des couleurs</p> <p><i>contraste</i></p> <p><i>texture</i></p> <p><u>Élément graphique</u></p> <p>Couleur (suivant le type de représentation)</p> <p><u>Texte</u></p> <p>Police</p> <p>taille</p> <p>style (gras, italique, gras italique, normal)</p> <p>effets (souligné, barré,...)</p> <p>Couleur</p>	<p>CompuServe GIF (=Graphics Interchange Format)</p> <p>Computer Graphics Metafile CGM</p> <p>Bitmap MBM</p> <p>Lotus Ami Draw SDW</p> <p>Paintbrush PCX, DCX</p> <p>Portable Network Graphics PNG</p> <p>Tagged Image File Format (TIFF, JPEG) TIF, JPG, JIF</p> <p>Card VCF</p> <p>Visio VSD</p> <p>Windows Bitmap BMP, RLE, ICO, CUR</p> <p>Windows Metafile WMF, EM</p>	<p>Lempel-Ziv</p>	
<p><i>propriétés fixes répétitives</i></p> <ul style="list-style-type: none"> <li>• Objets graphiques (ex.: )</li> <li>• Texte</li> <li>• Eléments graphiques</li> </ul>		<p>Animated gif, flash,</p>		
<p><u>Dynamique</u> (<i>propriétés changeantes, non-répétitives</i>)</p> <ul style="list-style-type: none"> <li>• Objets graphiques (ex film, vidéotéléphonie...)</li> <li>• Texte (ex défilement de cours boursier)</li> <li>• Eléments graphiques</li> </ul>	<p>vitesse de cadencement ou de déplacement des images</p> <p><i>Déplacement (relatif à l'observateur)</i></p> <p><i>Ordre d'apparition</i></p>	<p>avimpeg6,...</p>		
<b>SONORE</b>				
<p>Sons (analogues/arbitraires) - Langages réels</p>	<p>Position</p> <p>Arrangement (spatial et/ou temporel)</p> <p>Caractérisation</p> <p>Durée (rythme)</p> <p>Timbre</p> <p>Intensité (volume)</p> <p>Fréquence (Hauteur/vitesse)</p>	<p>Audio File</p> <p>WAV,</p> <p>AU,</p> <p>WVE;</p> <p>MP3,</p> <p>Real Audio (RA)</p>	<p>ACELP.net, CITT A-law, CITT u-law, IMA ADPCM, Lemout8Hauspie CELP ou SBC, Microsoft ADPCM, Microsoft G.723.1, MPEG layer-3, PCM...</p>	
<p>Sons synthétiques</p>		<p>Midi</p>		
<p>Langage de synthèse</p>	<p>Langue</p> <p>Timbre - Personnalité</p> <p>Intonation</p> <p>Vitesse</p>	<p>Suivant le programme du périphérique</p>		
<b>TACTILE</b>				
<p>langage et signe ="graphique" (ex. écran braille,...)</p>	<p>Position - Arrangement</p> <p>Caractérisation</p> <p>longueur – largeur, profondeur/relief, texture (qualité de la surface)</p> <p>orientation, vitesse</p>			
<p>Arbitraire ="signal sonore" (ex.vibreur)</p>	<p>température, pression, voltage, changement répétitif des propriétés</p> <p>changement non-répétitif des propriétés, vitesse</p>			

## 2.2. Etude des terminaux

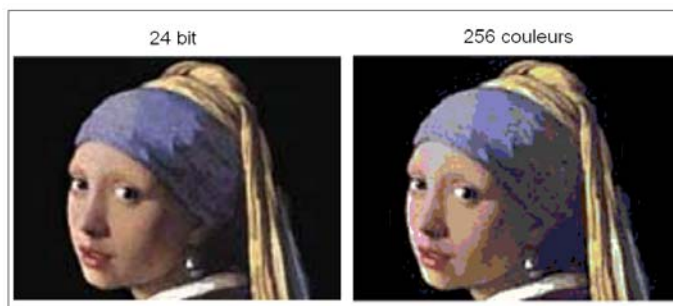
	Section	Type de terminaux visés
	<b>Objectif</b>	<ol style="list-style-type: none"> <li>1. Définir les terminaux visés</li> <li>2. Etablir les caractéristiques des terminaux influençant l'output.</li> <li>3. Déterminer l'influence de ces caractéristique sur l'output</li> </ol>
	<b>Entrée</b>	---
	<b>Sortie</b>	<ol style="list-style-type: none"> <li>1. Décomposition des terminaux en caractéristiques physiques des IU et en caractéristiques systèmes (cf. Tableaux de synthèse)</li> <li>2. Influence des caractéristique des terminaux sur l'output (cf. Tableaux de synthèse)</li> </ol>
	<b>Connaissances</b>	<ol style="list-style-type: none"> <li>1. Tableau général caractérisant les principaux modèles de terminaux mobiles disponibles sur le marché.</li> <li>2. Etudes sur l'influence des caractéristiques des terminaux sur les outputs</li> <li>3. Protocole CC/PP</li> </ol>

Il s'agit de caractériser les terminaux de manière à pouvoir déterminer les possibilités de ceux-ci à représenter les différents niveaux d'information. Ce qui signifie que nous ne prendrons en compte que les caractéristiques qui influencent l'output. Parmi ces dernières, on distinguera deux grandes catégories prises comme contraintes aux caractéristiques de base de l'information: les contraintes physiques des interfaces-utilisateurs et les contraintes-système. Les *contraintes physiques* des IU limitent la manière dont le signal de l'information est présenté à l'utilisateur sans se soucier de l'origine du signal (ex. type de fichier, browser, l'ordinateur qu'il y a « derrière »...). Par exemple, le signal d'une image en couleurs (24bit) de 200x120pxls sera visualisé différemment suivant qu'il soit interprété par un écran d'ordinateur (couleur, 1024x768) ou un écran de gsm (noir et blanc, 96x60). Sur un écran d'ordinateur, l'image sera en couleurs et semblera petite tandis que, sur un écran de gsm, elle sera en noir et blanc et on n'en verra que la moitié.



Les *contraintes-système* sont toutes les contraintes qui limitent le signal à l'interface, à tous les niveaux du système. On distingue dans cette catégorie les contraintes liées aux caractéristiques physiques du système et celles de la programmation. Dans les contraintes liées à la programmation, on a suivant les niveaux d'abstraction, le protocole, le langage de programmation « Markup », le langage de programmation, le système d'exploitation de « Microbrowser ». Les caractéristiques-système sont, quant à elles, les composantes de

l'ordinateur qui peuvent limiter le format d'une information. Par exemple, une carte graphique VGA ne permettra jamais d'afficher plus de 256 couleurs (8bit). Si l'image a été définie en 24bit, la carte divisera le spectre des couleurs en 256 et affichera comme une couleur les couleurs faisant partie d'une même division.



Après avoir modélisé les terminaux et identifié l'incidence des différentes composantes sur la représentativité de l'information sur l'IU, nous allons examiner la faisabilité d'une telle méthode.

### **2.2.1. Choix des terminaux**

Bien que la méthode soit valable pour tout type de terminal, nous allons nous concentrer sur les terminaux mobiles, car c'est parmi ceux-ci que l'on trouve la plus grande diversité de formats et ce sont eux qui promettent le plus grand développement dans les prochaines années<sup>59</sup>.

Etant donné qu'on ne peut analyser tous les terminaux mobiles, il est nécessaire de les regrouper par catégories, ce qui nous permettra de simplifier la suite de la méthode. Cette classification a été faite sur base d'un échantillonnage de 27 terminaux proposés par les fabricants leaders sur le marché<sup>60</sup> (Nokia, Ericsson, Palm, Handspring, Sony, Compaq, Hewlett Packard, Casio, Psion...) et sur base de l'évolution générale des terminaux mobiles. Pour une question de facilité, nous illustrerons la suite de la méthode par un exemple représentatif de chaque catégorie.

---

<sup>59</sup> Cf. (Introduction fig.1 et 2)

<sup>60</sup> voir tableau général reprenant les fiches techniques des terminaux mobiles, classés par catégories, proposés par les principaux fabricants. (Annexes 5)



Lorsqu'on regarde les terminaux mobiles par rapport à leur évolution, il est possible de distinguer les terminaux selon leur fonction première. D'un côté, il y avait les terminaux destinés à la téléphonie (téléphones mobiles), et de l'autre, ceux prévus pour servir d'assistant digital ou d'organisateur électronique (PDA ou Personal Digital Assistant). Les fonctions de ces derniers étaient généralement de type agenda, carnet d'adresse, calculatrice,... Suite à l'évolution des réseaux et des systèmes, ces deux types de terminaux ont eu accès au transfert de données et par là-même au net. Les technologies employées pour ces terminaux étant proches et les possibilités des réseaux de plus en plus grandes, la différence entre ces deux types tente de se réduire. On voit, d'ailleurs, apparaître des terminaux hybrides (appelés aussi « Smartphones ») qui essayent de combiner les fonctions du téléphone avec celles de l'assistant digital.

Au sein de ces trois catégories, la principale distinction se fait suivant les systèmes d'exploitation (OS<sup>61</sup>). En ce qui concerne les PDA, il est encore possible de les classifier selon leur « taille ». On distingue, en effet, les terminaux de type Palm (écran tactile et pas de clavier; ex. Palm, Compaq iPAQ...) des terminaux munis d'un clavier (ex. Psion, HP Jordana...).

### **a) Téléphones mobiles**

Les téléphones mobiles sont la catégorie où les caractéristiques sont les plus variées. Chaque fabricant, voire chaque modèle d'appareil, a son propre système opérationnel et sa façon d'orchestrer les touches qui permettent la navigation ou l'entrée de données dans le système. Celui-ci est généralement peu ou pas modifiable et les caractéristiques qui lui sont propres essaient d'apparaître comme les plus attrayantes en vue de gagner des parts de marché. A cause de cette diversité, il est nécessaire de considérer les appareils au cas par cas. Pour pouvoir, tout de même, prendre en compte cette catégorie, afin d'illustrer le modèle, nous devons choisir un appareil qui soit assez représentatif par ses caractéristiques et sa pénétration dans le marché. Il pourrait, ainsi, servir de référence. Le modèle retenu est le Nokia 6210 car c'est le milieu de gamme de Nokia, leader sur le marché des téléphones mobiles. De plus, les systèmes opérationnels des appareils développés par Nokia sont assez semblables à travers la

---

<sup>61</sup> OS = Operating System ou Système Opérationnel

gamme (3310, 6210, 7110, 8210, 8810), ce qui nous permet d'augmenter la représentativité de notre choix.

### **b) Personal Digital Assistant (PDA)**<sup>62</sup>



Les organisateurs de poche ont pour vocation initiale de remplacer à la fois l'agenda, le carnet d'adresses et le bloc-notes. Ou du moins, ils l'avaient car nombre d'entre eux méritent, aujourd'hui, pleinement l'appellation de PC de poche, tant leurs caractéristiques et leurs fonctions tendent à les rapprocher des ordinateurs de travail. Bien que très différents les uns des autres à la fois en terme de puissance et de fonctions, tous proposent aujourd'hui des logithèques étendues venant compléter les tâches de base des organisateurs : jeux, dessins, plans de villes, et même pour certains lecteurs, MP3 et vidéos.

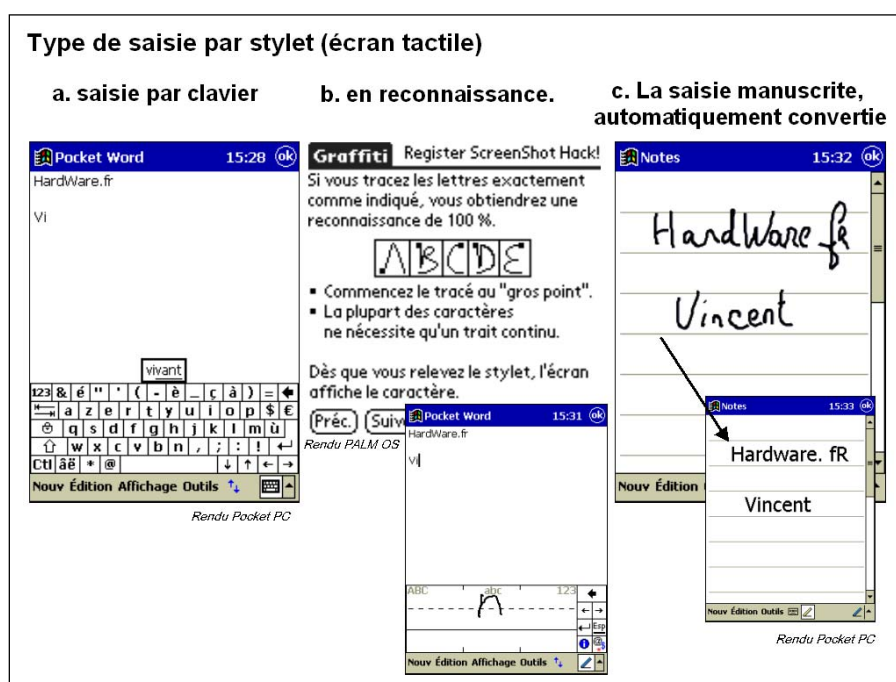
#### ***La saisie d'informations***

Tous les organisateurs électroniques suivent les mêmes principes de saisie d'informations. Il existe trois moyens d'interagir avec le terminal.

1. par stylet sur un écran tactile
  - a. clavier virtuel affiché sur écran
  - b. reconnaissance d'écriture (impose un apprentissage car chaque lettre doit être écrite selon un certain modèle)
  - c. saisie manuscrite automatiquement convertie par un logiciel de reconnaissance d'écriture cursive (utilisable dans les notes du Pocket PC).

---

<sup>62</sup> Sur base du tableau récapitulatif et de [9]



2. par le clavier si l'agenda s'en trouve pourvu, (Psion Revo et Serie 5,HPJornada 720)
3. Par commande vocale (uniquement sur Pocket PC)
4. par la synchronisation des informations avec l'ordinateur.

### *Types d'organiseurs électroniques*

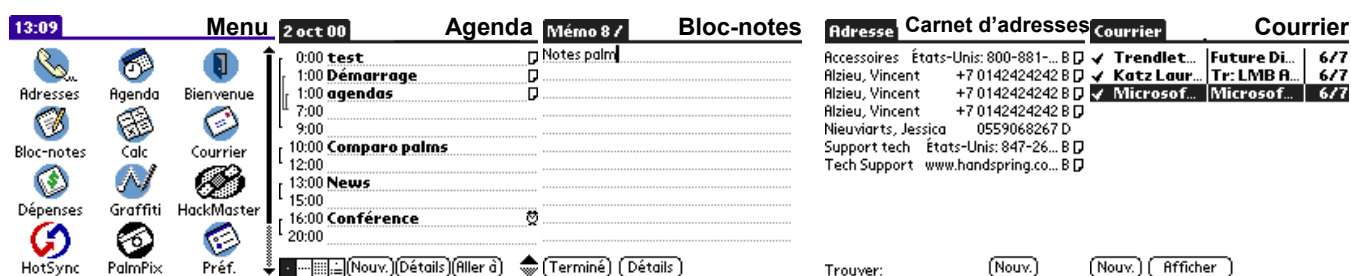
Aujourd'hui, trois grandes familles de systèmes opérationnels (OS) se partagent un marché pour le moins concurrentiel: Palm OS (représenté par Palm Computing, Handspring, IBM, Sony), PSION avec son système d'exploitation EPOC, les « PocketPCs », estampillés Microsoft WindowsCE.



Les Palm sont les premiers à être arrivés sur le marché et représentent environ 70% du marché mondial. Néanmoins, cette position quasi monopolistique tend à faiblir au profit du Visor d'Handspring qui représente déjà 25% du marché aux Etats-Unis. Cette position pour la moins dominante leur vient à la fois de la politique d'accord de licence mise en place dès le départ et de leur important réseau de développeurs évalué par Palm Computing à plus de 100 000 ingénieurs et techniciens pour plus de 5 000 logiciels destinés à cette plate-forme.

Le Palm est avant tout fonctionnel et ne comprend pas de fonctions superflues. Il est basé sur quatre applications de base : l'agenda, le répertoire téléphonique, le bloc-notes et le courrier électronique. L'usage de l'appareil est simple et intuitif (icônes claires, sauvegarde automatique de documents...) et la configuration semi-automatique.

PalmOS est généralement reconnu comme étant le système d'exploitation le plus pratique et le moins gourmand, donnée influant directement sur l'autonomie de l'appareil qui se comptabilise en jours et non en heures comme pour les Pocket PC.



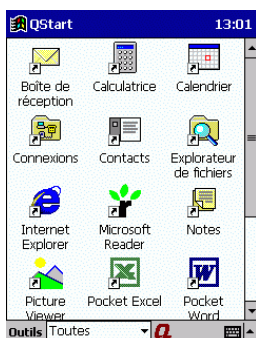
Tous les appareils sous PalmOS offrent une définition de 160 x 160 pixels en monochrome, à l'exception du Palm IIIc qui bénéficie d'un affichage en couleurs. Il dispose, à l'exception du palm m100, de 8 Mo de mémoire (RAM)

## **Pocket PC** **Pocket PC™**

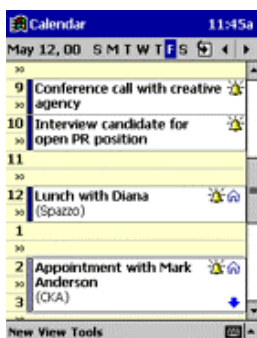
Les Pocket PC sont les terminaux qui fonctionnent avec Microsoft Windows CE comme système d'exploitation. Ils se présentent soit sous la forme d'un Palm (Compac iPac, HP Jordana) ou avec un clavier (HP Jordana 720, Compac Aero 8000H/PC Pro).

Le WindowsCE est une version simplifiée du Windows et s'utilise d'une manière plus ou moins semblable (menu « démarrer », explorateur de fichier...). En plus de fonctions de base du Palm, de nombreux programmes Windows y sont disponibles en version allégée (ex. Pocket Office, jeux,...).

Menu (compaq iPac)



Agenda



Bloc-notes



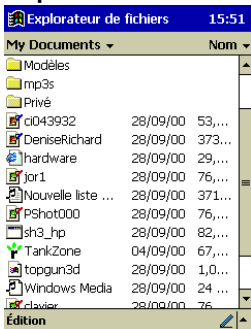
Carnet d'adresses



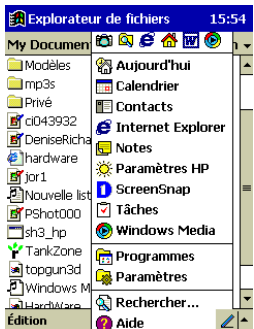
Courrier



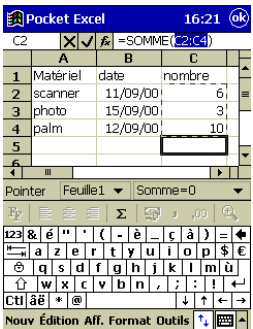
Explorateur



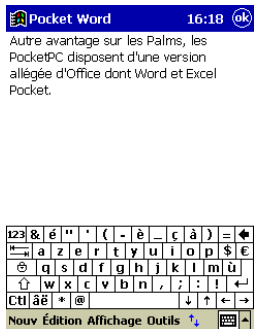
Menu Démarrer



Pocket Excel



Pocket Word



Jeu de golf



Contrairement au Palm, le Pocket PC a été conçu pour le net. Il supporte, en effet, un grand nombre de protocoles, de langages (HTML, WML, XML) et de types de fichier généralement utilisés pour l'échange des données (Flash, Java, Gif, Jpeg...). De plus, la navigation est optimisée grâce au browser. Par exemple, pour assurer la meilleure lisibilité possible des pages, le texte en navigation est conservé à sa taille initiale alors que les graphiques sont réduits. Il est également possible de montrer une même page en portrait ou en paysage



Lecture à la verticale

ou, à l'aide d'un utilitaire, à l'horizontale



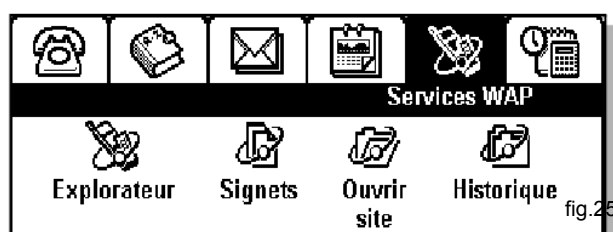
En conclusion, un des attraits majeurs des PocketPC réside dans leurs capacités multimédia. Il est à noter que les nombreux apports n'auraient pas pu être envisageables sur des configurations similaires à celles des Palm. Les modèles des différents constructeurs, Hewlett-Packard, Casio et Compaq, s'appuient aussi sur des processeurs cadencés de 130 à

200 MHz, disposent 16 à 32 Mo de mémoire et affichent jusqu'à 65 000 couleurs. Revers de la médaille, l'autonomie en pâtit : elle est de 6 à 8 heures.

### **Psion™ (EPOC)**

Si l'on devait comparer les Psion aux Palm et aux Pocket PC, on dirait qu'il a, d'un côté, l'aspect fonctionnel et léger du Palm et, de l'autre côté, un système d'exploitation offrant des possibilités semblables au Pocket PC, au multimédia près. Les Psion, au contraire des Palm, sont munis d'un clavier.

Psion a été l'un des pionniers dans le domaine et son handheld<sup>63</sup> était un classique bien avant l'apparition de CE. EPOC, le dernier système d'exploitation du Psion a été développé par Symbian. Cette entreprise est une joint venture regroupant Psion, Nokia, Ericsson Matsushita et Motorola. Son objectif est d'élaborer des standards en matière de communication mobile. Suite à EPOC 5, Symbian a sorti la « Symbian Platte-forme 6.0 » qui inclut les références de design Quartz et Cristal. Quartz est l'OS prévu pour des terminaux de type Palm (opérations entrées au stylet, écran 240x320 et de type tablette) tandis que Crystal est une version étoffée pour les terminaux munis d'un clavier et d'une résolution d'écran de 640x200 minimum. Bien que le remplacement d'EPOC semble être amorcé, Quartz n'est pas encore disponible sur le marché. Comme Palm et CE, EPOC est aussi implanté dans d'autres appareils (ex.:Ericsson R320 (cf. fig.25)).



<sup>63</sup> Handheld = petit ordinateur qui a un clavier

### c) Hybrides

Ces appareils, à mi-chemin entre le téléphone et le PDA, sont la résultante des améliorations techniques des appareils et des possibilités offertes par les nouveaux réseaux<sup>64</sup> (GPRS, UMTS...). En effet, les débits de transfert de données étant de plus en plus élevés, les terminaux ont dû s'adapter afin de satisfaire au mieux les utilisateurs. C'est ainsi qu'après les SMS (Short Message System) désormais disponibles sur tous les téléphones mobiles, on voit apparaître des appareils qui supportent des protocoles plus lourds comme le WAP ou le http. Actuellement, l'arrivée de l'UMTS et des réseaux de troisième génération motivent beaucoup de projets de ce type. Ces projets sont initiés autant par les fabricants spécialisés dans la téléphonie (Nokia), que par les fabricants de PDA<sup>65</sup>. On assiste donc à une véritable mutation qui risque de fort influencer le paysage des terminaux mobiles des prochaines années.



Etant donné que peu de terminaux sont déjà sur le marché, nous allons prendre, pour illustrer cette catégorie, les appareils proposées par Nokia et Ericsson (leader du marché) : Nokia 9210 Communicator et l'Ericsson R380.

#### **Tableau de synthèse reprenant la typologie des terminaux**

Voici le tableau représentant les différentes catégories identifiées de terminaux. Le modèle qui illustre chaque catégorie est considéré comme représentatif et servira d'exemple aux différentes étapes de la méthodologie.

Différents TERMINAUX								
Type	Téléphones	Hybrides			PDA (Personal Digital Assistant)			
				3G	Palm Sized		Handheld PC	
OS	Nokia	EPOC	EPOC	???	Palm OS	Microsoft Windows CE 3.0	EPOC (v.5.01)	Microsoft Windows CE (3.0)
illustration								
Modèle	Nokia 6210	Nokia 9210	Ericsson R320	???	Palm V1x	Compaq iPaq H3650	Psion 5MX	HP Jornada 720

<sup>64</sup> Cf. section II.3.2

<sup>65</sup> Palm a signé des accords avec Motorola et Kyocera pour développer des téléphones en commun. Tout comme Intel avec Ericsson et Samsung. (<http://www.hardware.fr/html/articles/lire.php3?article=262&page=1>)

## **2.2.2. Contraintes physiques des interfaces-utilisateurs des terminaux**

Nous allons voir, dans un premier temps, comment il est possible de caractériser une interface indépendamment de l'ordinateur qui l'anime. Dans un second temps, nous examinerons les répercussions qu'elles ont sur les caractéristiques de base définies dans le chapitre 2.2.3.2.

### **2.2.2.1. Définition et caractérisation des interfaces**

De façon logique, à chaque mode de représentation (graphique, sonore ou tactile) correspondent une ou plusieurs interfaces spécifiques qui permettent de le représenter sur un terminal mobile. Par conséquent, on trouve l'écran pour le mode graphique et les enceintes pour le mode sonore. Ces dernières peuvent être soit intégrées au terminal soit externes (baffles, oreillette, écouteurs...). Pour le mode tactile, les interfaces sont plus nombreuses et inégales quant à l'information délivrée. On peut y distinguer, d'abord, les interfaces capables de transmettre n'importe quel type d'information (une information complète), ce sont celles que l'on a précédemment appelées « graphiques » (ex. écran braille). Ensuite, on peut y percevoir les interfaces dites « sonores » qui ne sont capables de transmettre qu'une information partielle sous la forme d'un signal, ce sont celles que l'on nommait « sonores ». Pour la transmission par voie interne, on cite comme exemple un vibreur ou un signal thermique tandis que pour voie externe, on pense à un gant tactile ou un joystick dynamique (qui dans une simulation de voiture se durcit dans les tournants ou vibre lors d'une sortie de piste). Cependant, puisque aucun terminal ne dispose d'interface tactile intégrée, nous ne tiendrons pas compte de ce mode, d'autant plus que ce type de périphérique est normalement géré par son propre gestionnaire.

#### **a) Caractéristiques de l'écran**

Pour définir un écran, les dimensions couramment utilisées sont la taille, la résolution et le fait que l'écran soit en couleur ou non.

La **taille** est donnée en pouces et représente, dans le système standard, la longueur de la diagonale de l'écran. Cela permet de n'avoir qu'une seule mesure. Les écrans étant un rectangle construit en standard dans un rapport de 4:3 (largeur:hauteur), la diagonale est l'hypoténuse d'un triangle rectangle dont le rapport des cotés est connu. Par exemple, un



écran d'ordinateur de 15 pouces est un écran qui a 9 pouces de haut sur 12 pouces de large<sup>66</sup> (soit 22,86cm sur 30,48cm). Mais la majorité des terminaux ne respecte pas ce standard. Il faut alors se référer au rapport donné par la résolution. En effet, le rapport de la hauteur sur la largeur de l'écran est proportionnel au rapport du nombre de points affichables sur ces deux dimensions. Par exemple, les points affichables sur l'écran d'un Palm sont au nombre de 160 sur la hauteur et de 160 sur la largeur. Quelle que soit la taille (donnée par la diagonale) de l'écran, celui-ci sera un carré (rapport « largeur:hauteur » étant de 1 (160:160)).

La **résolution** de l'écran donne le nombre maximal de points affichables sur la largeur et la hauteur de l'écran. Nous distinguons la résolution de la taille, car la grosseur des points peut varier. Pour se rendre compte de la taille et de la densité des points et pouvoir comparer les écrans entre eux, nous rapportons le nombre de points affichables à une largeur et une hauteur fixes. Celles-ci sont mesurées, par convention, en pouces. Cette mesure, le « point par pouce » (ou DPI, Dots Per Inch) nous renseigne en pratique sur la qualité du « grain » de l'image.

La troisième dimension utilisée pour caractériser un écran est sa capacité à afficher de la **couleur**. On peut soit avoir un écran en couleur soit un écran monochrome.

### **b) Caractéristiques des enceintes**

Le haut-parleur est un appareil qui convertit en ondes acoustiques les courants électriques correspondant aux sons de la parole ou de la musique<sup>67</sup>. C'est une membrane qui engendre des ondes sonores en vibrant sous l'impulsion d'un dispositif électromagnétique ou électrostatique. L'enceinte se caractérise par ses capacités à générer des sons. Vu les caractéristiques du son, celles du baffle définissent une capacité à produire une puissance sonore (mesurée en Watt) et une certaine fréquence de son. On peut donc définir une bande de fréquences<sup>68</sup> parmi la bande de fréquences audibles<sup>69</sup> qui serait accessible à l'enceinte. Les sons normalement émis lors de l'usage des terminaux mobiles (musique, voix, alarmes) restent dans des fréquences qui sont couvertes par la très grande majorité des enceintes. C'est

---

<sup>66</sup> système de deux équations à deux inconnues  $15^2 = X^2 + Y^2$  et  $X/Y = 3/4$

<sup>67</sup> Larousse

<sup>68</sup> ensemble de fréquences comprises dans un intervalle donné (Larousse)

<sup>69</sup> entre 15Hz (son grave) et 15kHz (son aigu)

pourquoi, nous n'accordons à cette caractéristique qu'une importance secondaire.

### Tableau de synthèse reprenant les IU et leurs caractéristiques

Type de IU		Caractéristiques de IU				Différents TERMINAUX									
		Type				Hybrides				PDA (Personal Digital Assistant)					
		OS				Palm Sized				Handheld PC					
		Protocole				Palm OS				Microsoft Windows CE 3.0					
		WAP 1.1				WAP, HTTP, JAVA				WAP, HTTP					
illustration															
Modèle		Nokia 6210		Nokia 9210		Ericsson R320		Général		Palm Vllx		Général		Compaq iPaq H3690	
<b>OUTPUT GRAPHIQUE</b>															
<b>ECRAN</b>		Taille (rayon en pouce)		0	0	0	0	0	0	0	0	0	0	0	
		Rapport LxH		1,57	NA	2,9	NA	variable	3,3	variable	3,9	NA	NA	NA	NA
		Résolution en PXL (LxH)		16:15	ouvert 16: 5	ouvert 3:1	NA	1:1		3:4		8:3	8:3	4:3 ou 8:3	8:3
		Résolution en DPI		96 X 90	600x200	360x116	NA	160 X 160		240 X 320		Variable 480x180 à 640x240	640 x240	variable 640 X 480 (voir plus)	640x240
		Couleur				58,51	NA	Variable	68,56		84,577				
		Couleur			X		X	X		X					X
		Monochrome		X		X		X	X			X	X		
<b>OUTPUT SONORE</b>															
<b>BAFFLE</b>		Interne		0		0		0	0	0	0	0	0	0	0
		Puissance													
		mWatt				250		Variable		Variable		Variable		Variable	
		ohm				32		Variable		Variable		Variable		Variable	
		Décibel (dB)Max						Variable		Variable		Variable		Variable	
		Fréquences (min-max)						Variable		Variable		Variable		Variable	
<b>Externe</b>		OPTION		oreillette	OPTION	oreillette; baffles	OPTION	oreillette	OPTION	écouteurs; baffles	OPTION	écouteurs; baffles	OPTION	écouteurs; baffle	OPTION
<b>OUTPUT TACTILE</b>															
<b>INFO COMPLETE</b>		Interne		N	N	N/A	N	N	N	N	N	N	N	N	N
		Externe		N	N	N	N	N	N	N	N	N	N	N	N
<b>ECRAN TACTILE</b>		(langage+analogue)													
		• prédéfini													
		• « graphique » (points)													
<b>INFO PARTIELLE</b>		Interne		0	0	0	0	???	???	???	???	???	???		N
		Vibreur		X	X	X	X	???	???	???	???	???	???		
<b>Externe</b>		Gant tactile		N	N	N	N	N	N	???		N	N	OPTION	
		Joystick dynamique		N	N	N	N	N	N			N	N		

O = oui / N = non /  ou N/A = Non Available = information non disponible auprès de la documentation fabricant

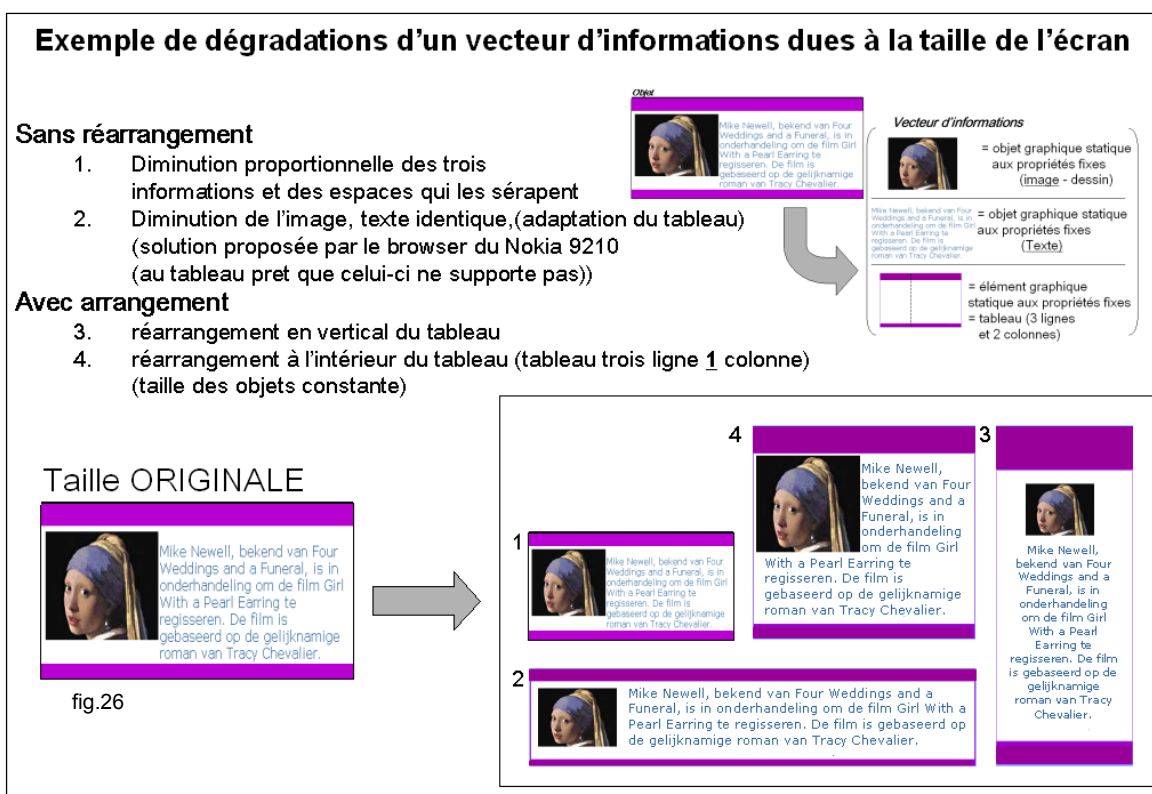
### 2.2.2.2. Répercussions des paramètres physiques des IU des terminaux sur les caractéristiques de base de l'information

Tout d'abord l'analyse des terminaux nous permet de nous renseigner sur l'existence des interfaces, et par là nous renseigne sur les modes par lesquels une information peut être représentée sur un terminal visé. Les caractéristiques physiques des IU des terminaux ne nous donnent, quant à eux, qu'une information relative en ce qui concerne la façon d'ajuster les caractéristiques de base de l'information. En effet, l'affichage dépend surtout des contraintes-systèmes comme, par exemple, la carte graphique ou le browser. Les caractéristiques physiques nous donnent principalement des informations sur le rendu de l'output mais ne limitent en rien l'information puisqu'une interface peut en principe tout afficher. Par exemple une image de 30x50cm en couleur peut être affichable sur un écran de GSM mais on n'en verra qu'une très petite partie et en noir et blanc. Il suffirait pour cela que le GSM ait la mémoire et les programmes capables de supporter le format et la taille de l'image. Pour voir l'entièreté de l'image il faudrait aussi que le browser permette un défilement horizontal et vertical. Il est néanmoins possible de définir des règles indicatives, soit pour ne pas tomber dans des cas absurdes comme celui qui vient d'être énoncé, soit pour optimiser la vitesse de transmission. Par exemple :

***Règle 7. Sauf usage particulier (zoom), ne jamais afficher une image avec une taille supérieure à celle de l'écran. En cas d'absence de possibilité de scrolling cette règle devient impérative.***

***Règle 8. Dans les cas où la vitesse d'accès à l'information est un facteur prépondérant dégrader les caractéristiques de l'information de manière à ce qu'elles soient inférieures ou égales aux capacités de l'IU du terminal.***

De plus, les caractéristiques physiques du terminal sont d'une importance cruciale pour ce qui est du positionnement ou de l'arrangement des différents objets. Pour le cas envisagé, la *figure 26* montre les conséquences possibles qu'une diminution de l'écran sur l'arrangement des informations au sein d'un objet de niveau supérieur.



### 2.2.3. Contraintes-systèmes liées aux terminaux

Comme nous l'avons annoncé auparavant, nous avons pu répertorier, cinq types de contraintes-système majeures. Nous allons les définir et en analyser l'incidence sur les différentes caractéristiques de base de l'information, terminant ainsi l'étape portant sur la modélisation des terminaux.

#### 2.2.3.1. Définitions des contraintes-systèmes

##### a) Protocole

Le protocole est un ensemble de messages et de procédures qui régissent la communication entre les entités. Dans le domaine des transferts de données par Internet, on utilisera en général http et WAP, SMS ou i-Mode (Japon) pour les terminaux mobiles.

##### *HTTP (Hypertext Transfer Protocol)*

HTTP est le protocole utilisé par le World Wide Web : comme protocole standard. Il définit comment les messages sont formatés et transmis et quelles actions les serveurs web et les browsers devraient donner pour réponse à différentes commandes. Par exemple, quand on entre un URL (Uniform Resource Locator : il spécifie l'adresse d'une ressource sur le web) dans le browser, une commande HTTP est envoyée au serveur web. Celui-ci la dirige afin de

recupérer et transmettre la page demandée.

### ***WAP (Wireless Application Protocol)***

Il spécifie le cadre de l'application et les protocoles de réseaux pour les appareils sans fil tels que les téléphones mobiles et les PDAs. Les spécifications WAP prolongent les technologies de réseaux mobiles (telles que les standards de réseaux de données digitales) et les technologies Internet (telles que XML, URLs et différents formats de contenu).

### ***SMS (Short Message System)***

C'est un protocole défini pour le GSM. Il permet la transmission point par point de courts messages. Un message est dit court s'il a une longueur de 140 bytes (ou 160 caractères) et s'il n'utilise que les caractères d'un ensemble existant de 7-bits (=128 caractères). A l'aide de messages concaténés, il peut être utilisé pour transmettre jusqu'à 31 KB de texte binaire ou de données.

## **b) Langages de programmation Markup ou langage de marquage**

En règle générale, à chaque protocole, on peut associer un ou plusieurs langages de marquage. Pour illustration, HTTP a comme langage HTML et le WAP a comme langage WML... Afin de développer de futures applications sur Internet, nous avons besoin de langages de marquage. Ceux-ci décrivent les applications au moyen de tags. Les avantages principaux de ce langage sont de créer des applications indépendamment des plates-formes, facilement transposables sur le web et compilables gratuitement par un browser client.

Les standards du web sont régis par le W3C (World Wide Web Consortium). Les principaux sont HTML, CSS<sup>70</sup> et XML.

### ***HTML (Hypertext Markup Language)***

HTML est un fichier qui permet d'organiser les informations sur une page web décodable par un browser (cf. infra). Ce fichier se présente sous un format de fichier texte composé de tags de marquage dont l'extension est HTM ou HTML. Le standard HTML le plus récent est le XHTML1.0.

---

<sup>70</sup> CSS (Cascading Style Sheet) : Les styles définissent comment afficher les éléments HTML. Ils sont normalement enregistrés dans des styles sheets. Les styles sheets externes sont stockés dans des fichiers CSS.

### ***WML (Wireless Markup Language)***

C'est un langage de marquage basé sur XML (voir infra) et est destiné à l'usage de contenu spécifié et à l'IU pour des terminaux avec des bandes de transmission étroites tels que les téléphones mobiles.

### ***XML (EXtensible Markup Language)***

Contrairement à HTML et WML qui ont été conçus pour décrire l'apparence des données à l'affichage, XML l'a été pour décrire la substance des données. De plus, ses tags ne sont pas prédéfinis.

### **c) Langages de programmation**

Ces langages servent à programmer des applications sur un ordinateur. On trouve notamment Java, Flash et PCP, des langages de programmation pouvant être utilisés pour définir des applications ou des objet sur le web (menu déroulant, objets de navigation,...).

### **d) Systèmes d'exploitation<sup>71</sup>**

C'est le programme qui gère l'ensemble du terminal. Il s'agit de plate-forme sur laquelle viennent s'implanter les différentes applications. De manière générale, chaque terminal est lié à un système d'exploitation.

Dans le cas des terminaux mobiles, si l'on ne considère pas les cas particuliers, trois systèmes d'exploitations se partagent le marché : Eloc (Symbian), Palm OS et Microsoft WindowsCE.

### **e) Microbrowser**

C'est le programme qui permet de compiler les langages de marquage et de visualiser l'information ainsi codée sur un terminal mobile. Par exemple : Nokia Browser, Phone.com, UP.browser, Mobile Explorer...

### **f) Caractéristiques du système**

On regroupe sous cette dénomination les éléments physiques qui constituent le terminal et conditionnement ses capacités du terminal à représenter une information. Parmi ces éléments

---

<sup>71</sup> La littérature utilise généralement le terme anglais: Operating Systems (OS)

nous distinguons : le processeur (type et vitesse de cadencement), la mémoire (ROM et RAM), l'espace de stockage de l'information (disque dur, cartes mémoire...), la carte graphique (résolution, couleurs), le carte son (type, canaux) et le modem ou carte ethernet wireless<sup>72</sup> (vitesse de transmission kbit/s) sans lesquels la transmission de données est impossible.

### **2.2.3.2. Incidence des contraintes-systèmes sur les différentes caractéristiques de base de l'information**

Il est question de voir dans quelle mesure les différentes contraintes système (définies dans la section précédente) limitent les caractéristiques qui définissent la façon de représenter une information (cf. chap.2.1.2.). Il faut donc, pour chaque contrainte-système examiner :

1. Quels sont les modes (graphiques, sonores, tactiles) qu'elle supporte ou influence
2. Pour les modes qu'elle supporte, l'intervalle des possibilités pour chaque caractéristique de base et les types de fichiers supportés

Mais les incidences des contraintes systèmes sont multiples et interviennent avec des niveaux de pertinence différents sur la façon de représenter une information. Pour un objectif de concision et pour ne pas dépasser des limites de ce travail, les incidences présentées ci-dessous ne sont en rien exhaustives. Elles sont données à titre indicatif ou exemplatif afin de mieux illustrer la méthodologie suivie par le modulateur.

#### **a-b) Le protocole / langage de marquage**

Chaque langage de marquage est dépendant d'un protocole et ne peut aller outre les capacités de celui-ci. C'est pourquoi, nous considérerons les deux comme un tout et ne tiendrons compte que de celui qui a une influence directe sur l'information : le langage de marquage.

Les langages de marquage sont capables, par définition, de traiter tout type d'information mais ils ne génèrent, par eux-mêmes, que du texte ou des éléments graphiques (tableaux servant à l'arrangement des objets, formulaires, boutons...). Pour le reste, ils font appel soit à des fichiers externes, soit à des procédures écrites dans d'autres langages de programmation

---

<sup>72</sup> wireless = sans fil

(java, javascript, flash). Il est donc intéressant, pour les langages de marquage, de savoir d'abord quelles sont les caractéristiques possibles pour le texte ainsi que pour les éléments graphiques, ce qui revient à voir les tags existants et leurs effets. Ensuite on voit quels sont les types de fichiers que les langages supportent et à quel mode ils correspondent.

Les langages auxquels nous nous intéressons sont l'HTML et le WML car ce sont les deux langages actuellement accessibles aux terminaux mobiles.

### *HTML*

Etant donné que l'HTML est le standard général, nous considérons qu'il admet tous les fichiers standards et ne pose aucune restriction quant au texte et aux éléments graphiques.

Les fichiers standards d'échange sur Internet sont (liste non-exhaustive):

- pour les images : gif, jpeg et bmp
- pour les films : mpeg, avi, mov
- pour les sons : wav, RA (Real Audio), mpeg

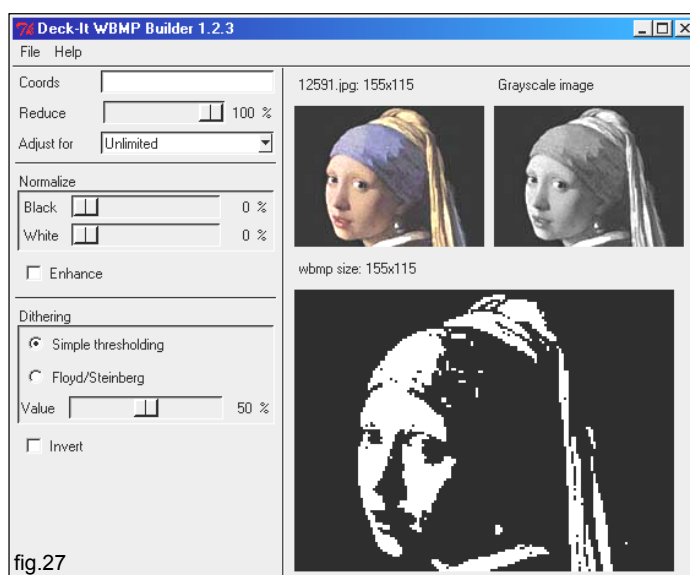


## WML<sup>73</sup>

Le WML ne supporte que le mode graphique. L'unique type de fichiers acceptés est le WBMP (WAP BitMap) qui est un format limité au noir et blanc (*cf. fig.27*). Pour ce qui est des éléments graphiques, le WML permet l'usage des tableaux et les formulaires (type texte, case radio...). En ce qui concerne le texte, les caractéristiques de celui-ci sont limitées aux quelques tags prévus à cet effet. Voici l'ensemble de ces tags illustrés avec un exemple (*cf. fig.28*)

<pre> &lt;wml&gt; &lt;p&gt; &lt;em&gt;emphasized text&lt;/em&gt;&lt;br/&gt; &lt;strong&gt;strong text&lt;/strong&gt;&lt;br/&gt; &lt;b&gt;bold text&lt;/b&gt;&lt;br/&gt; &lt;i&gt;italic text&lt;/i&gt;&lt;br/&gt; &lt;u&gt;underlined text&lt;/u&gt;&lt;br/&gt; &lt;big&gt;big text&lt;/big&gt;&lt;br/&gt; &lt;small&gt;small text&lt;/small&gt;&lt;br/&gt;  Normal&lt;br/&gt; &lt;/p&gt; &lt;/wml&gt; </pre> <p style="text-align: right;"><i>Ligne de programmation en WML_1.1.XML</i></p>	<p>emphasized text  <b>strong text</b>  <b>bold text</b>  <i>italic text</i>  <u>underlined text</u>  big text  small text  Normal</p> <p><i>Rendu en taille réelle</i>  (Simulé par WinWAP 3 Pro)</p>
--	--

fig.28



La figure 27 montre un convecteur d'image en WBMP, les possibilités offertes pour la conversion ainsi que l'image de exemple traité tout au long du texte. Il s'agit du programme Deck-it WBMP Builder Version 1.2.3 ©2000 [www.PyWeb.com](http://www.PyWeb.com)

<sup>73</sup> Pour toute information complémentaire voir [http://www.w3schools.com/wap/wml\\_reference.asp](http://www.w3schools.com/wap/wml_reference.asp) et [www.wap.org](http://www.wap.org)

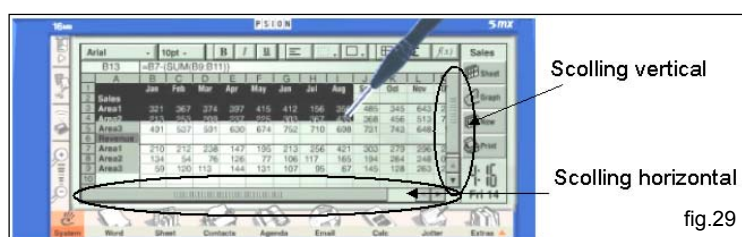
### c) Langages de programmation

Il est intéressant de voir quels sont les types d'informations pouvant être gérés par le langage et si le langage de programmation utilisé pour une application ou un objet est supporté par l'environnement du terminal mobile. Le cas échéant, de voir comment il est possible de reconverter une application écrite, dans un langage, dans un autre supporté par le terminal et les contraintes systèmes. Mais, pour une raison de concision du travail, nous n'entrerons pas dans ces distinctions.

### d-e) Systèmes d'exploitation et le Microbrowser

Aujourd'hui, en règle générale, il semblerait que chaque microbrowser soit conçu pour un système d'exploitation. Ce dernier est lui-même lié à un terminal. Vu la situation actuelle, nous ne parlerons plus que du microbrowser et de ses contraintes sur les caractéristiques de l'information, car tout ce qui est valable pour lui l'est également pour le système d'exploitation. Néanmoins, la distinction reste pertinente, surtout, pour le futur où la distinction entre système et microbrowser sera beaucoup plus évidente.

Pour le browser, il est impératif de savoir quels sont les langages de marquage qu'il est capable de compiler et la façon dont il le fait. En effet, tous les browsers n'interprètent pas les langages de la même manière et tous ne supportent pas toutes les caractéristiques ou toutes les versions des langages. Par exemple, certains browsers ne reconnaissent pas les frames<sup>74</sup> ou les tableaux. Nous devons également, comme pour les langages, voir quels sont les types de fichiers supportés. Une autre caractéristique essentielle est le scrolling<sup>75</sup> (ex. fig29).



En effet, la présence de cette fonction influence directement la taille et l'arrangement des

<sup>74</sup> Subdivision de la page en « sous-pages ».








<sup>75</sup> Scrolling : fonction qui permet de naviguer sur une page dont la taille dépasse celle de l'écran. Cette fonction s'exécute au moyen des touches du clavier (ex : flèches), de la souris ou des barres de défilement

objets. On en retire les règles suivantes :

**Règle 9.** *Si le microbrowser ne permet pas de scrolling, la taille maximale de l'objet (page, objet ou information) ne pourra être supérieure à la taille de l'écran ; Si le microbrowser admet un scrolling uniquement vertical ou horizontal, la taille maximale de l'objet sera ajustée respectivement soit horizontalement soit verticalement à l'écran ; Si le browser permet un scrolling vertical et horizontal, la taille de l'objet n'est pas limitée. Néanmoins attention à la R7.*

Notons également que la zone affichable sur l'écran est bien souvent réduite à cause de la place prise par les barres-menu du browser ou de la place que celui ci prend sur l'écran dans le cas où il n'aurait pas été affiché en fenêtre pleine. Par exemple l'écran du Nokia 9210 permet un affichage de 640x200pxls mais l'affichage maximum d'une image à l'écran est de 540x200plxs.

### **Tableau de synthèse reprenant les possibilités de scrolling pour les terminaux choisis**

Différents TERMINAUX							
Télé-phones	Hybrides		PDA (Personal Digital Assistant)				
mobiles			Palm Sized		Handheld PC		
	Sabian EPOC	Sabian EPOC	Palm OS	Microsoft Windows CE 3.0	EPOC (v.5.01)	Microsoft Windows CE (3.0)	
							
Nokia 6210	Nokia 9210	Ericsson R320	Général Palm VIIx	Général Compaq iPaq H3650	Général Psion 5MX	Général HP Jordana 720	
Scrolling							
vertical	O	O	O	O	O	O	O
horizontal	N	O	N	N	O	O	O

### **f) Caractéristiques du système<sup>76</sup>**

Vu la complexité des répercussions de ces contraintes sur la manière de représenter une information, nous nous limiterons, pour ce sujet, à quelques indications d'ordre général.

Le processeur, caractérisé par son type et sa vitesse de cadencement (ex. : Compaq iPaq a un processeur Intel strongARM SA-1110 cadencé à 206 MHz), influence la vitesse de

<sup>76</sup> pour connaître les caractéristiques des différents terminaux se référer à l'annexes 5

fonctionnement du terminal. Par exemple, la vitesse est un facteur clé pour Internet afin de jouer sur cet élément, nous pourrions réduire la taille des fichiers pour optimiser le temps nécessaire à leur affichage.

La mémoire RAM influe sur la taille des fichiers gérables en une seule fois par le terminal.

***Règle 10. Dans le cas du modulateur, nous prendrons comme règle réductrice que la taille de l'ensemble de la page (somme de la taille du code et de l'ensemble des objets affichés) soit inférieure à la mémoire disponible du terminal.***

L'espace de stockage (disque dur...) limite les possibilités de « download<sup>77</sup> ». Les terminaux mobiles ne disposant pas, généralement, de disque dur, ils utilisent leur mémoire RAM comme espace de stockage. Cet espace étant relativement faible (8 à 32 MB), il serait utile de l'optimiser. Afin d'y arriver, il faut faire un arbitrage entre la qualité de l'information (type de fichier, couleur, taille de l'objet...) et la place qu'il occupe.

Pour le modem, le même arbitrage (qualité-place) est à faire en vue d'optimiser la vitesse de transmission. En effet, même si de manière générale la vitesse doit être optimisée, pour certaines applications, les utilisateurs sont prêts à attendre le temps nécessaire (ex. imagerie médicale...). L'arbitrage est donc fonction des préférences des utilisateurs. Ce point sera développé ultérieurement.

La carte graphique régit le mode graphique. Elle a comme caractéristiques la mémoire, la définition (affichable) et les couleurs supportées. Par exemple, les cartes de type monochrome, cga, ega, vga, svga supportent respectivement les 1, 2, 4, 8 ou 16, 24 ou 32 bits. Les trois derniers cas sont pour des cartes de types égaux ou supérieurs à svga. On peut en tirer comme règle :

***Règle 11. Le nombre de couleurs de l'information ne peut être supérieur à celui supporté par la classe graphique.***

La carte son régit le mode sonore. Elle se définit par son type (Soundblaster™...) et son taux d'échantillonnage<sup>78</sup> (en bits). Les sons réels sont possibles à partir d'une carte 16 bits.

---

<sup>77</sup> Download = téléchargement de fichiers

<sup>78</sup> Ce taux correspond à la taille en bits d'un mot mémoire.









Autrement le nombre de bits définit la qualité et le nombre de canaux<sup>79</sup> par lesquels peuvent passer les sons synthétiques.

### **Tableau de synthèse: Implications des caractéristiques-système**

<b>Caractéristiques du système</b>	<b>Conséquences</b>										
• <b>Processeur</b> (type, vitesse de cadancement)	→ vitesse										
• Mémoire ROM-RAM	→ vitesse → taille des fichiers → résolution, taille										
• <b>Espace de stockage</b> espace disponible pour stocker ou «mémoriser» des données (fichiers) dans le terminal	→ vitesse → taille des fichiers										
• <b>Modem</b> (vitesse de transmission kbit/s)	→ vitesse										
• <b>carte graphique</b> (mémoire; couleurs supportées (bit))	→ vitesse, fluidité → définition → couleur										
<table border="1" style="width: 100%;"> <tr> <td>monochrome (hercule,...) (presque inexistantes)</td> <td>(1bit)</td> </tr> <tr> <td>cga</td> <td>(2bit)</td> </tr> <tr> <td>EGA</td> <td>(4bit)</td> </tr> <tr> <td>VGA</td> <td>(8bit)</td> </tr> <tr> <td>SVAGA</td> <td>(16,24,32 bit)</td> </tr> </table>	monochrome (hercule,...) (presque inexistantes)	(1bit)	cga	(2bit)	EGA	(4bit)	VGA	(8bit)	SVAGA	(16,24,32 bit)	
monochrome (hercule,...) (presque inexistantes)	(1bit)										
cga	(2bit)										
EGA	(4bit)										
VGA	(8bit)										
SVAGA	(16,24,32 bit)										
• <b>carte son</b> (canaux (bit)) canaux (nombre de voies (sons différents) qui peuvent être gérées simultanément)	→ qualité des sons synthétiques (format midi,...)										

<sup>79</sup> Nombre de sons que la carte peut gérer simultanément.

### Tableau de synthèse : Les caractéristiques-système des différents terminaux

TERMINAUX											
Télé-phones mobiles	Hybrides		PDA (Personal Digital Assistant)								
	Sabian EPOC		Palm Sized			Handheld PC					
	Sabian EPOC	Sabian EPOC	Palm OS	Microsoft WindowsCE		EPOC (v.5.01)		Microsoft WindowsCE			
											
	<b>Nokia 6210</b>	<b>Nokia 9210</b>	<b>Ericsson R320</b>	<b>Général</b>	<b>Palm Vlx</b>	<b>Général</b>	iPaq H3650	<b>Général</b>	<b>Psion 5MX</b>	<b>Général</b>	<b>HP Jornada 720</b>
Protocole	WAP	WAP, HTTP	WAP			HTTP	HTTP			HTTP	HTTP
Langage de programmation Markup	WML	HTML, WML	WML			HTML	HTML	HTML	HTML	HTML	HTML
Langage de programmation		Java				Java, Flash	Java, Flash			Java, Flash	Java, Flash
Système d'exploitation	Nokia	Nokia	EPOC (hybride)	Palm OS	Palm OS	Microsoft Windows CE	Microsoft Windows CE	Variable	EPOC	Microsoft Windows CE	Microsoft Windows CE
Microbrowser	Nokia	Nokia 9210 browser (EPOC)	Nokia R320 (EPOC)	Browser Palm (PALM)	Browser Palm (PALM)	Mobile explorer	Mobile explorer	N/a (EPOC)	N/a (EPOC)	Mobile explorer	Mobile explorer
Caractéristiques-système											
Processeur, (vitesse)				Variable	N/a	Variable	206 MHz	variable	36 MHz	Variable	206 MHz
Mémoire ROM				N/a	N/a	8 ou 16 Mb	16 Mb	N/a	N/a	Variable	N/a
Mémoire RAM		14 application+2 user+8SDRAM		8 Mb	8Mb	variable	32 Mb	Variable	16 Mb	16 ou 32 Mb	16 Mb
Espace de stockage		Plus de 16 Mb en carte									
Modem (vitesse)				N/a	N/a	N/a	N/a	N/a	N/a	56 ou 33,6 Kbps	56 Kbps
Carte graphique	Oui (caractéristiques supplémentaires non disponibles)										
Carte son	Oui (caractéristiques supplémentaires non disponibles)										

#### 2.2.4. Conclusion

Toutes les caractéristiques identifiées dans cette section ont été modélisées d'un certain point de vue. Les résultats de notre modélisation sont concordants avec ceux d'autres travaux. A ce titre, on trouve le CC/PP, développé par le WAP forum.

Le **CC/PP**<sup>80</sup> est un protocole dont la structure permet de communiquer l'information sur les capacités et les caractéristiques du terminal et du réseau, aux serveurs du web et aux gateways/proxies de telle façon que le contenu qui convient soit rendu au terminal. Pour ce faire, le CC/PP utilise la technologie RDF (Resource Description Framework), qui est elle-même un sous-ensemble du XML. La description RDF consiste en une collection structurée de propriétés RDF. Chacune d'elles est associée avec un type et une valeur de propriété et est exprimée comme un graphe directement labellisé avec des nœuds, des feuilles et des arcs ou un modèle objet orienté donné défini par un vecteur (objet, attribut, valeur) (ex. en annexe 6).

Le fait que le modulateur puisse utiliser un modèle de plate-forme n'est donc pas utopique puisqu'il pourrait être alimenté par ce type de protocole.

On peut alors imaginer deux approches du modulateur lorsqu'une requête d'un utilisateur apparaît. Soit le modulateur proposerait un menu composé de différents terminaux parmi lesquels l'utilisateur choisirait le sien. Ce serait une approche a priori : on aurait programmé manuellement dans le modulateur les caractéristiques générales de chaque catégorie de terminaux. Soit le modulateur identifierait automatiquement les caractéristiques du terminal client et lui proposerait la solution la plus propice. Cette solution pourrait se faire sur base de protocole du type CC/PP.

---

<sup>80</sup> [CC/PP] Franklin Reynolds, Johan Hjelm, Spencer Dawkins, and Sandeep Singhal, "Composite Capability/Preferences Profiles: A user side framework for content negotiation," W3C Note, 27 July 1999, <http://www.w3.org/TR/1998/NOTE-CCPP-19990727>. (Note that this specification refers to the 7/99 version of this specification, irrespective of the availability of future versions of that specification).  
 [CC/PPex] Hidetaka Ohto and Johan Hjelm, "Composite Capability/Preference Profiles (CC/PP) Exchange protocol based on HTTP Extension Framework," W3C NOTE 17 June 1999, <http://www.w3.org/Mobile/Group/IG/1999/06/NOTE-CCPPexchange-19990617>. (Note that this specification refers to the 6/99 version of this specification, irrespective of the availability of future versions of that specification).  
 [RDF] Lassila, "Resource Description Framework (RDF) Model and Syntax Specification," W3C Recommendation, 2/99, <http://www.w3.org/TR/1999/PR-rdf-syntax-19990105>.

## Tableau de synthèse : Caractérisation des terminaux

### I. Contraintes Physiques des UI

#### a) Graphique = écran

- Taille
- Résolution
- Couleur

#### b) Sonore = enceinte

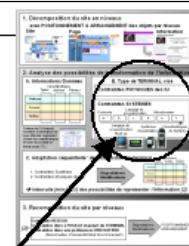
- Puissance
- bande de fréquences tolérées

#### c) Tactile = variable

- interne : vibreur
- externe : géré par des gestionnaire de périphérique



n'entre pas dans le champs du modulateur



### II. Contraintes Systèmes

Classées conventionnellement par ordre séquentiel (sous contrainte forte que ce soit possible)

#### a) Protocole

- HTTP (→ HTML), WAP (→ WML), i-mode, SMS...

#### b) Langages de programmation Markup

- HDML, CompactHTML, WML, Subsets of HTML, MML, PGA, Voice XML

#### c) Langage de programmation

- Java, flash, php

#### d) Systèmes d'exploitation

- Sábian (→ Quartz & Cristal), Epoc, Palm OS, Microsoft Windows CE

#### e) Microbrowser

- Nokia, Phone.com, Ericsson R320, UP.browser, Mobile Explorer

#### f) Caractéristiques du système

- Processeur (vitesse de cadancement)
- Mémoire ROM-RAM
- Espace de stockage (Mbite)
- Modem (vitesse kbit/s)
- Carte graphique (mémoire; couleurs supportées (monochrome, cga (2bit), EGA (4bit), VGA (8bit), SVAGA (16,24,32 bit))
- Carte son (type; mémoire; cannaux)



### 2.3. Adaptation de l'information au terminal (Etape 3)

Maintenant que nous avons défini et caractérisé d'un côté l'information et de l'autre les terminaux, il est possible d'adapter la caractérisation d'une information (définition de l'objet en tant que tel comme vu à la section 2.1) au profil d'un terminal (défini selon le point 2.2). Le but d'une telle adaptation est de donner, pour chaque caractéristique de l'information, un intervalle (min,max) des valeurs acceptables par le terminal.

Ce résultat est obtenu par l'optimisation des caractéristiques de l'information sous contraintes du terminal. Idéalement, on devrait y arriver grâce à un système de priorisation automatique des contraintes défini dès le départ. Cependant, la résolution d'un tel problème relève d'un CSP (Constrains Satisfaction Program<sup>81</sup>) ce qui dépasse largement le cadre fixé à ce mémoire. Toutefois, afin d'apporter une réponse, on traitera séquentiellement les contraintes. L'illustration d'une méthode séquentielle est développée au chapitre qui traite de l'application du modulateur à un arbre de décision (Chap 3).

A partir des différentes caractéristiques (possibles pour un terminal donné) ainsi obtenues, nous pouvons construire un ensemble de représentations d'une même information en combinant différentes valeurs de ces caractéristiques.

Finalement, sur base de l'ensemble des différentes informations ainsi traitées, nous pouvons reconstruire les objets par niveaux. Le choix des représentations pour chaque information ainsi que l'arrangement de celles-ci se fait en tenant compte des contraintes dues au réseau, à l'usage normal du terminal et aux préférences des utilisateurs.

---

<sup>81</sup> Programme de satisfaction de contrainte

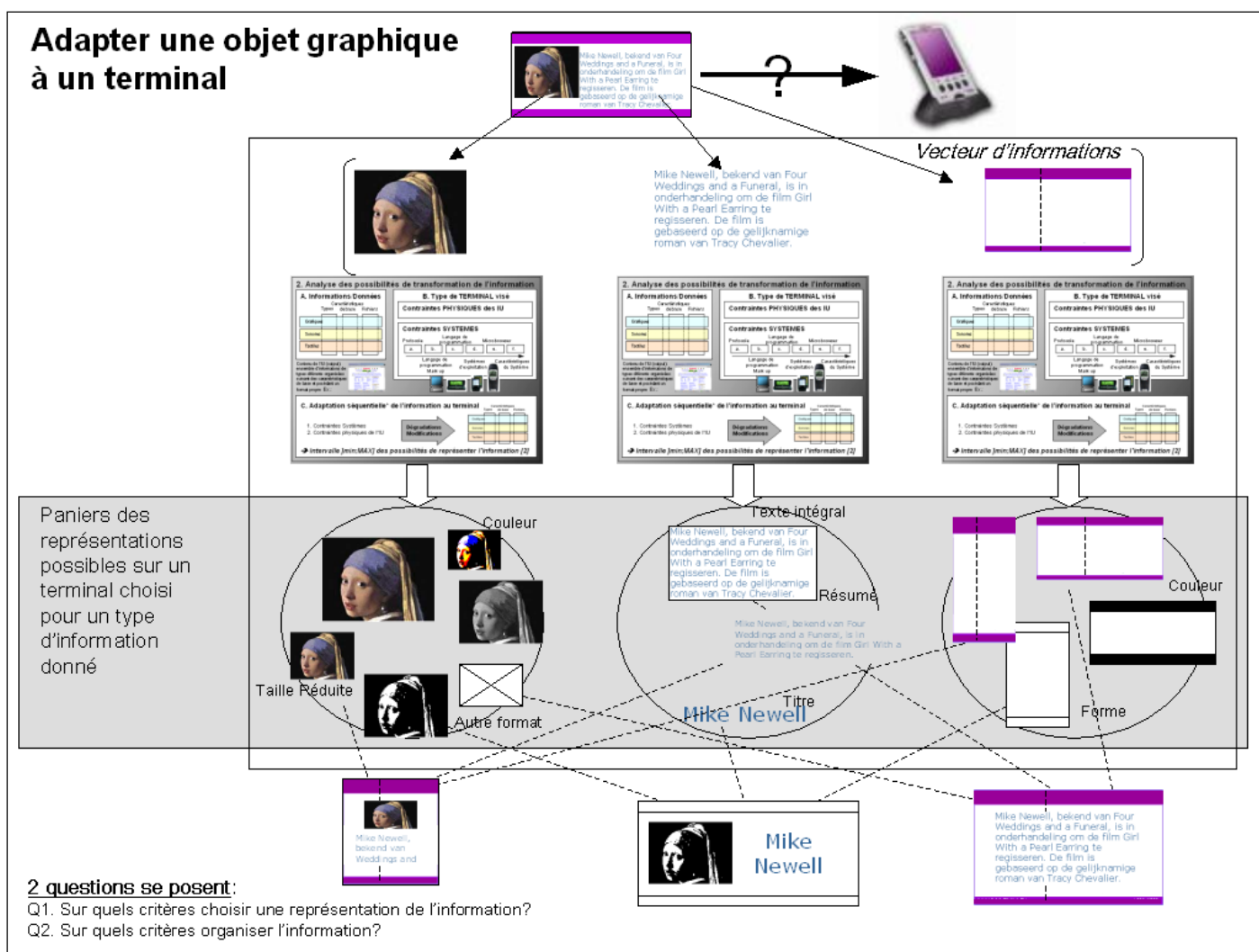
## 2.4. Synthèse graphique de des deux premières étapes

Voici une représentation de l'output généré à la suite des deux premières étapes lorsqu'on considère l'ensemble des informations du site (de l'exemple traité au fil du travail).

Suite à cela, deux questions se posent :

1. Sur quels critères choisir une représentation de l'information dans le panier disponible ?
2. Sur quels critères organiser ou réarranger les représentations choisies afin de reconstruire le site ?

Nous tenterons d'y apporter une réponse dans le chapitre suivant.



### 3. Recomposition du site par niveaux

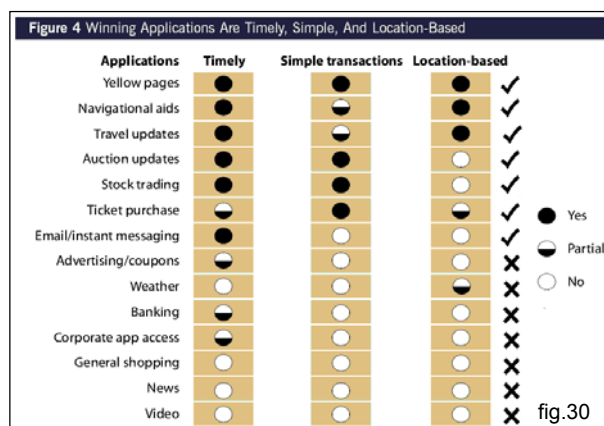
En bref, à présent, nous disposons de tout le site décomposé en objets et en niveaux et nous avons pris note, à chaque fois, du positionnement et de l'arrangement des objets. Grâce à la deuxième partie, nous possédons pour chaque information (dernier niveau de décomposition) l'ensemble des représentations acceptables par un terminal. Cet ensemble est le résultat de la combinaison des différentes valeurs dans les intervalles de possibilités des caractéristiques de base. On qualifiera cet ensemble de « panier ». Il s'agit maintenant de recomposer le site initial. Ce choix et ce réarrangement sont faits, dans les grandes lignes, par l'usage standard du terminal, puis en vue d'optimiser le résultat, par les préférences des utilisateurs à partir des possibilités offertes par les réseaux.

#### 3.1. Usage standard du terminal

Vu les études de marché portées sur les utilités des terminaux, on décèle les besoins des utilisateurs de base pour chaque terminal. Ce sont ces besoins de base que l'on appelle l'« usage standard ». Pour le cas du téléphone, des études comme celles de Forrester<sup>82</sup> (cf. fig. 30) nous montrent qu'une application pour ce terminal doit être rapide, simple et locale. Les utilisateurs du WAP sont, en effet, composés principalement d'hommes d'affaires recherchant des informations précises et ponctuelles.

D'un côté, on remarque dans la plupart des cas que le facteur de succès se nuance entre la vitesse et la qualité. D'un autre, on est face à cet immense panier de possibilités (illustré à la page précédente). Notre idée est d'organiser les éléments de ce panier sur un vecteur de vitesse et donc, sur un de

qualité (cf. fig.31 et 32) (ce dernier vecteur ayant la même grandeur, le même positionnement mais la direction opposée de celui d'avant).



<sup>82</sup> Forrester : The Dawn of Mobile eCommerce (octobre 99)

fig.31

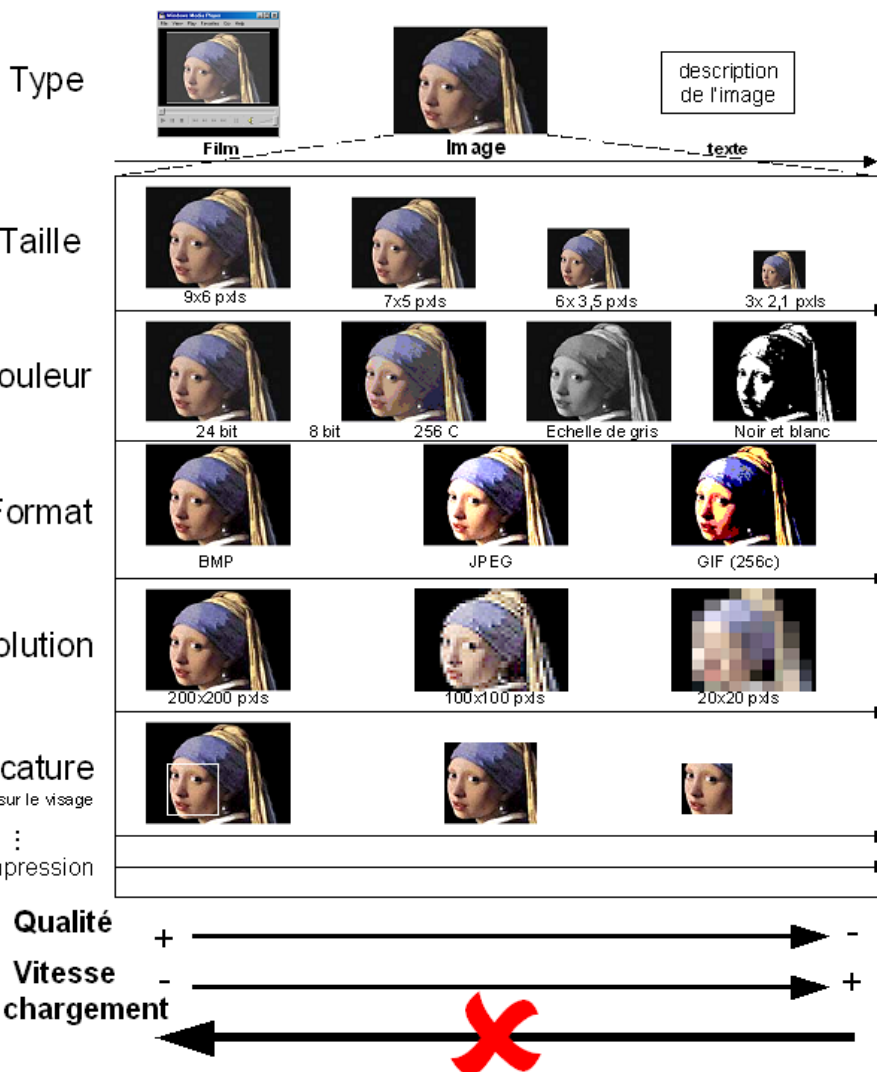
ex.	Forme			
	actuelle	recommandée		
Qualité		Min	moyen	Max
	Film	texte	image	film

**!** doivent être adaptées aux contraintes antérieures. C-à-d on ne peut jouer que sur les formes, modes et caractéristiques supportées par l'IU, le système,...

ex.	Caractéristiques de base			
	actuelle	recommandée		
Qualité		Min	moyen	Max
Image	couleur (16 bit)	1 bit	8 bit	16 bit
	...	...	...	...

ex.	Format			
	actuelle	recommandée		
Qualité		Min	moyen	Max
Image	BPM	gif	jpeg	bpm
	...	...	...	...

### Exemple de dégradation pour une image



Afin de parvenir à ce positionnement dans les deux axes, nous définissons pour chaque caractéristique de l'information une échelle. Pour simplifier, on divise cette échelle en trois parties seulement : qualité maximum, qualité moyenne ou qualité minimum. Lors des combinaisons, on sait ainsi, de suite, où positionner la combinaison obtenue sur le vecteur vu qu'on saura dans quelle partie de l'échelle on l'aura prise à chaque fois (cf.fig.32). Il est à noter que lorsque ces combinaisons sont faites, certaines ne peuvent être conservées et sont donc rejetées (ex. gif en 16 bits est rejeté).

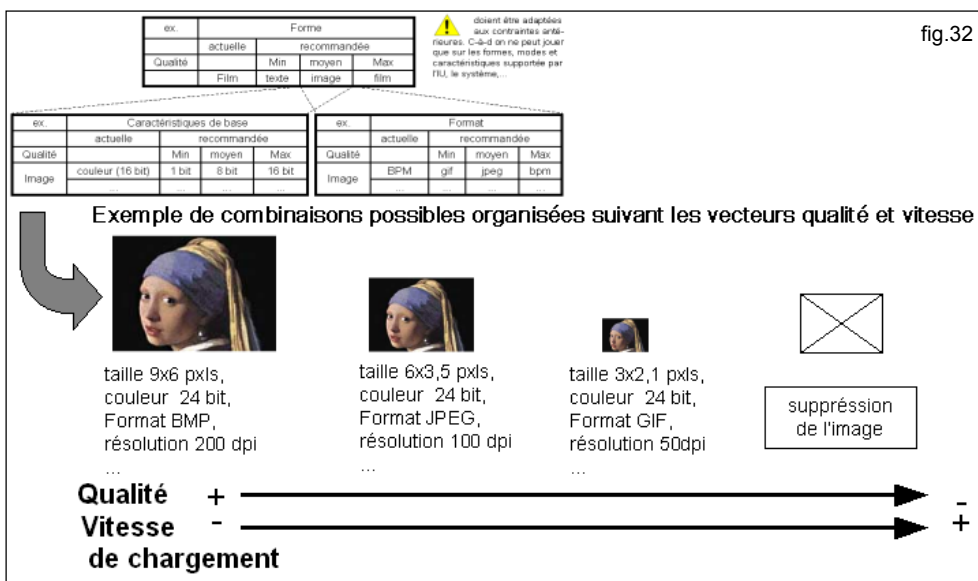


fig.32

Ensuite, vu chaque degré d'importance possible pour un type d'information, on définit une deuxième dimension qui apparaît perpendiculairement à l'autre (cf. fig.33).

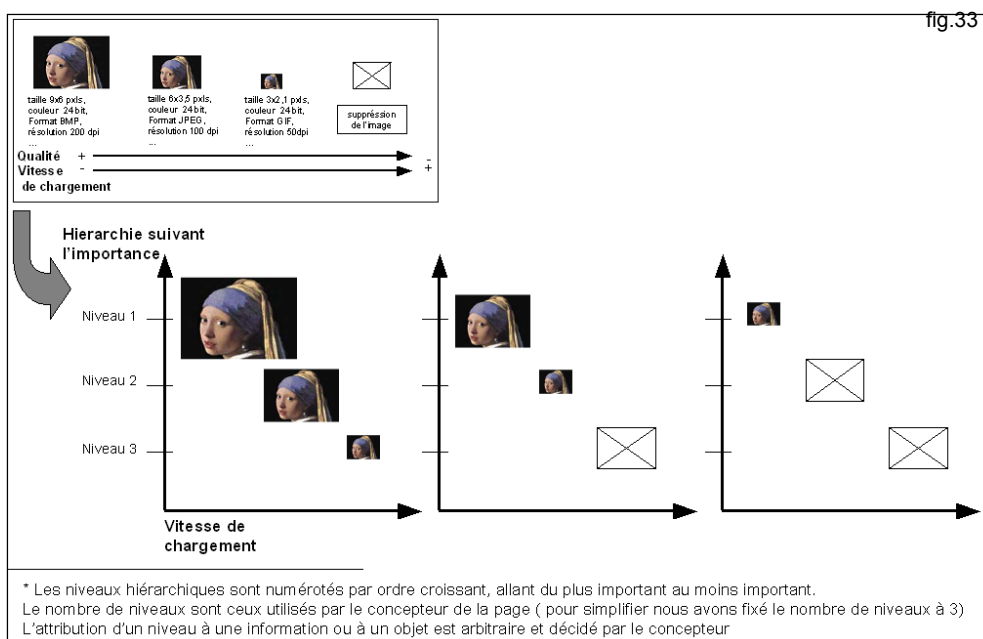
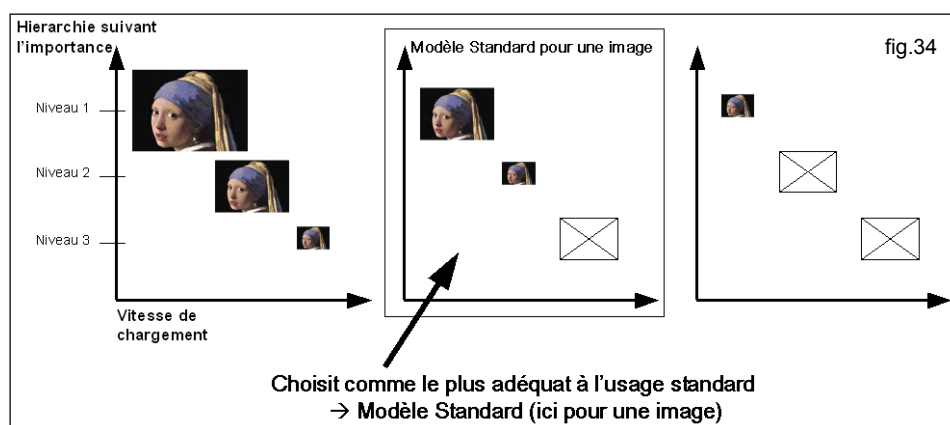
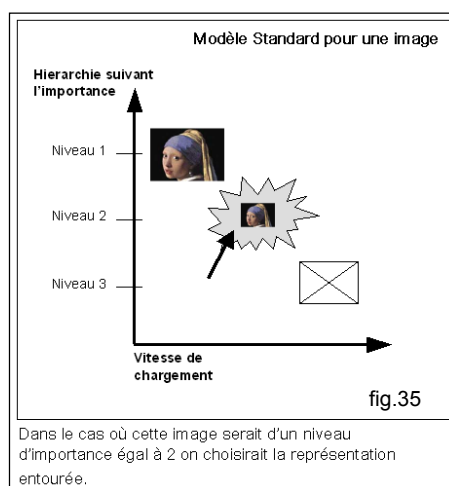


fig.33

Pour chaque type d'information que l'on traite dans le site, l'usage standard définit une représentation (sur l'axe qualité-vitesse) à chaque niveau de hiérarchisation possible (*fig.34*). Ce modèle de possibilités (appelé **modèle standard**) servira, désormais, pour tous les objets du même type. Ce qui signifie, par exemple, qu'après avoir défini une représentation pour chaque niveau de hiérarchisation d'une image, toutes les images du site d'un même niveau seront représentées de la même façon. L'ensemble des modèles standards (pour tous les types d'information) forme le **mode standard**.



Ensuite, vu le degré d'importance ayant été attribué par le concepteur du site à l'information on choisira la représentation adéquate (*fig.35*).

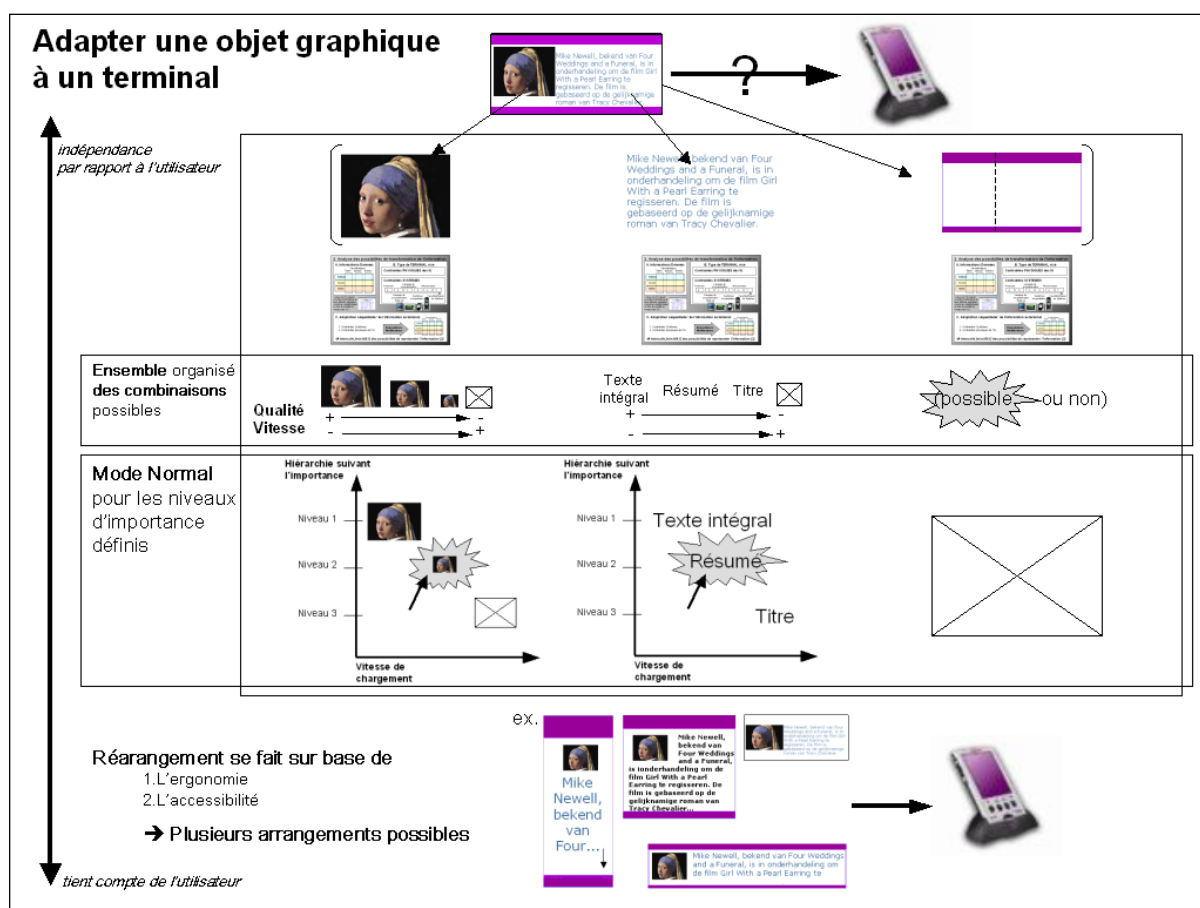


Une fois que l'on dispose de toutes les représentations adéquates, il ne reste plus qu'à réarranger les objets entre eux. Ce réarrangement se fait sur base de connaissances sur l'ergonomie des interfaces-utilisateurs de terminaux mobiles et sur les préférences des utilisateurs (accessibilité...). Cette discussion ne rentrant pas dans les limites fixées à ce travail, pour de plus amples renseignements à ce sujet, nous vous référons aux travaux cités

dans la bibliographie [34, 36, 37, 38, 39, 41, 75].

## Tableau de synthèse

Pour obtenir une classification par rapport à la qualité ou à la vitesse, notons que la dégradation des informations se fait sur les caractéristiques de base pour les informations importées (fichier son, image...). Par contre, elle se fait sur le contenu pour les informations générées par le système. Ainsi, la manière d'afficher un texte (type de la police, taille, couleur...) n'influence pas la vitesse. Le seul moyen de gagner de la vitesse est d'afficher moins de caractères. Cette dégradation du contenu se fait sur base des niveaux hiérarchiques. Cette façon de raccourcir un texte peut se faire, par exemple, en coupant le texte (extrait, titre...) ou en le résumant suivant les niveaux. Pour ce qui est du résumé, il peut être soit manuel, soit fait automatiquement par des programmes ou scripts prévus à cet effet (Perl<sup>83</sup>...).

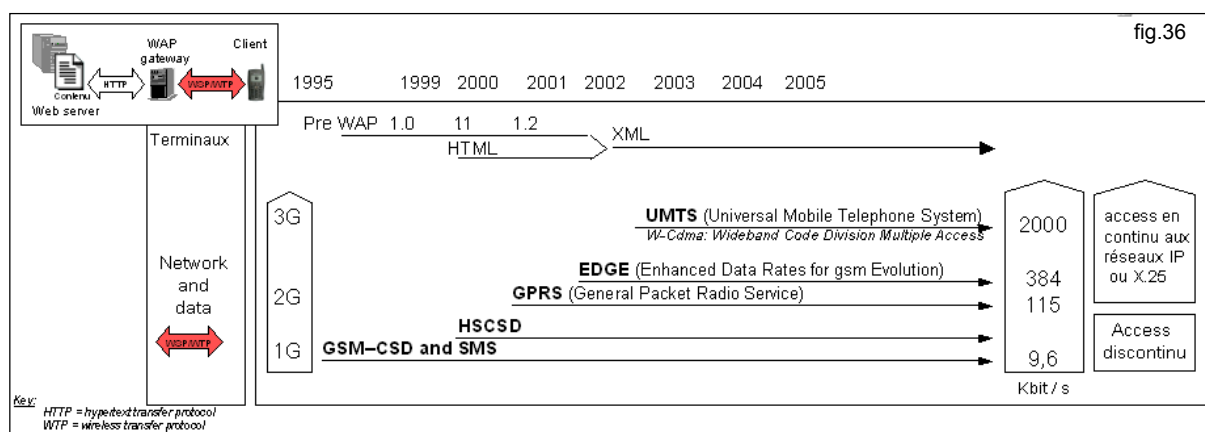


<sup>83</sup> Practical Extraction and Report Language: script CGI ([www.apache.org](http://www.apache.org))

Cependant, le mode standard (ensemble des modèles hiérarchisation-vitesse standard) peut être influencé par les deux points suivants : le réseau et les préférences-utilisateurs.

### 3.2. Réseau

Le réseau se compose de deux parties : un câblage physique (http) et un réseau aérien (WSP/WTP) dont nous allons nous occuper. Le réseau influence la vitesse. Le graphique ci-dessous (*cf. fig.36*), montre que les possibilités de vitesse de transmission se sont améliorées dans le temps, entre la première et la troisième génération. Ce qui a permis de supporter des protocoles de plus en plus lourds.



L'exemple (*cf. fig.37*) des images en jpeg et l'évolution du temps nécessaire à leur affichage au fil du temps. On voit que le temps diminue d'une manière drastique ce qui permet d'espérer en de nouvelles applications telles que le wireless HDTV.

→ vitesse = KSF

fig.37

ex: image jpeg 16KB	GSM (9,6KB/S)	GPRS (115KB/S)	EDGE (384 KB/S)	UMTS (2000 KB/S)
temps nécessaire à l'affichage (s)	1,667	0,139	0,042	0,008

Pour une approximation des différents types d'informations de chaque réseau, on se référera au tableau en *annexes 7*. En conclusion, la solution choisie par le standard peut être modifiée en fonction du réseau utilisé.



### 3.3. Préférences des utilisateurs

Le mode normal (défini au point 3.1) peut aussi être décliné en sous-modes sur base de l'environnement dans lequel se trouve l'utilisateur, des problèmes d'accessibilité ou plus simplement les goûts/préférences propres de l'utilisateur.

#### 3.3.1. Discussion générale

Il y a deux manières de concevoir cette **modification du mode normal**:

##### 3.3.1.1. Hors modulateur

Dans la plupart des cas, les paramétrages « fins » sont gérés par les browsers, les programmes systèmes des terminaux (ex. modes silencieux, normal, extérieur des GSM; mode mal-voyant dans les configuration windowsCE...), ou bien même par les gestionnaires de périphériques (surtout tactiles). Dès lors, nous ne pouvons qu'émettre des recommandations afin de créer le layout original le plus complet et réfléchi possible. Il est à noter que dans cette solution l'interaction se fait surtout sur les caractéristiques primaires (volume, agrandissement des icônes, taille de la police, affichage ou non les images (menu « apparence » du browser Nokia 6210)...) et moins, voire pas du tout, sur le retraitement ou l'agencement des objets et des informations. En effet, la gestion de cette partie nécessiterait de télécharger tous les éléments pour y faire une sélection par la suite. Cela entraînerait une grande perte de temps (transfert et traitement) et nécessiterait beaucoup de mémoire. Sans compter la perte de contrôle de la part du créateur du site.

##### 3.3.1.2. Dans le modulateur

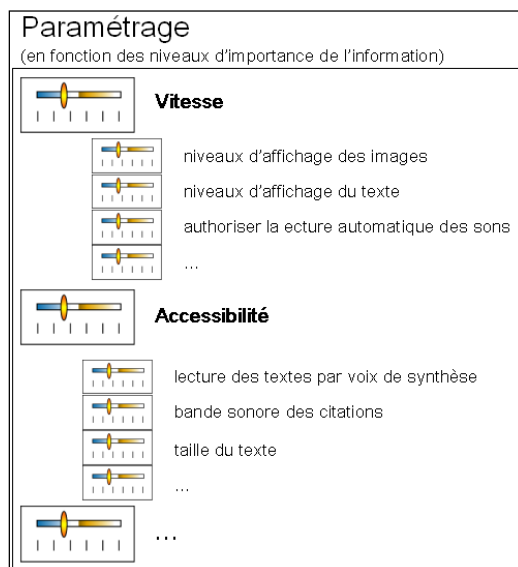
Il serait possible d'imaginer que l'utilisateur puisse modifier l'apparence du site web à partir de celui-ci, de la même façon que l'on peut choisir la langue de certains sites. Il faudrait alors définir des facteurs de préférence qui permettraient de calibrer les différents standards pour les différents types d'informations. Ceux-ci pourraient se présenter sous la forme de contrôles digitaux placés sur la page d'accueil du site (*exemple, fig. 38*).

Concrètement, le modulateur générerait des versions différentes de pages « identiques » suivant un nombre défini de combinaisons de facteurs. Ces facteurs (accessibilité, vitesse...) présentent comme avantages, pour l'utilisateur, de ne télécharger que les éléments utilisés

(gain de temps et de mémoire) et pour le créateur, de garder le pouvoir sur son site.

Dans le cas où un standard serait mis en place, ce choix pourrait être fait une fois pour toutes dans la configuration du terminal qui, à son tour, choisirait les pages correspondantes.

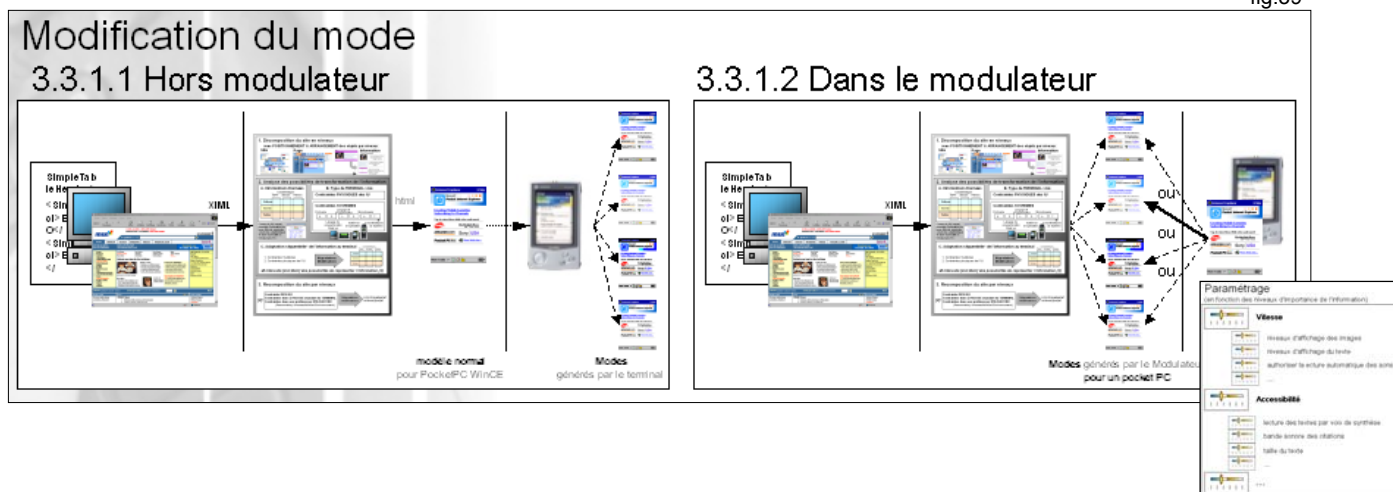
Pour cette solution, il serait nécessaire de définir différents facteurs de préférence, la façon dont ils varient et l'incidence qu'ils ont sur la construction des standards des différents types d'information. La définition et l'argumentation de ces facteurs préférentiels dépassent le cadre de ce travail. C'est pourquoi, nous laissons ce sujet à d'éventuels travaux futurs.



### Illustration de la section (3.3.1)

La figure 39<sup>84</sup> les possibilités présentées aux points 3.3.1.1 et 3.3.1.2

fig.39



<sup>84</sup> Cette figure est reprise en annexes 8

### 3. Application du modulateur à un arbre de décision

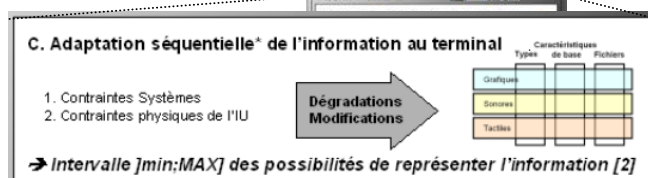
Dans ce chapitre nous allons illustrer la méthode suivie par le modulateur en l'appliquant à un arbre de décision (séquentiel). Comme nous l'avons déjà dit, cette façon de faire n'est pas optimale, mais elle permet de donner une idée assez précise quant au résultat. Cet arbre a été simulé sur ObjectVision 1.0<sup>85</sup>, une copie du programme figure sur la disquette jointe en annexe (le fichier contenant l'arbre est « modul.ovd »).

ObjectVision<sup>®</sup> a l'immense avantage de parcourir automatiquement l'arbre de décision en fonction des données introduites par l'utilisateur au moyen de formulaires. Après avoir parcouru l'arbre il donne la réponse conséquente aux données introduites et permet en double-cliquant sur la conclusion de se situer dans l'arbre. On peut donc bien parler d'un « simulateur ».

L'arbre de décision, porte sur l'adaptation d'une information à un terminal. Ce qui correspond à la troisième section de la deuxième étape du modulateur, soit la 2.2.3 dans le texte (zone entourée de la *cf. fig.40*). L'objet de cette simulation est de construire l'ensemble (ou panier) des possibilités de représentation d'une information possible sur un terminal pour ensuite pouvoir passer à la phase de recomposition du site par niveau (chap3 du Modulateur). Cet ensemble est construit suivant les règles et la méthodologie identifiée dans le travail (*cf. table des règles*).



fig.40



Ce programme ne permet pas de combiner les différentes possibilités pour chaque caractéristique de base de l'information. En conséquence, il se contente de donner pour chaque caractéristique les possibilités affichables, du moins dans les limites du programme. En effet, bien que le programme le permette, ce fichier n'est pas relié à une base de données. Ce qui a pour effet que les données doivent donc être introduites manuellement et de ce fait, elles ne peuvent être exhaustives. Cette limitation du « simulateur » a entraîné beaucoup de simplifications dans la caractérisation des contraintes-système et notamment pour ce qui est

<sup>85</sup> Object Vision, Copyright<sup>®</sup> 1991, Borland International

des langages de marquage et des browsers. L'output final du simulateur est donc un mélange de possibilités et de conseils ou de marches à suivre pour les différentes caractéristiques d'une information. Rappelons que le but ici, est de comprendre la logique du modulateur plus que de développer un outil performant. Notons que le fait de lier cette interface à une base de données qui comprendrait de manière exhaustive les informations sur les différentes composantes des contraintes-système (en particulier sur les langages de marquage et les browsers...) rendrait cet outil beaucoup plus précis.

Le « simulateur » reprend les trois parties évoquées au chapitre 2.2. La troisième partie, celle qui fait l'objet de l'arbre, se tant base sur la caractérisation de l'information que sur celle du terminal. L'introduction de ces données se fait au moyen des formulaires suivants : « définition de l'information », pour l'information, et « définition du terminal », « incidence du langage de marquage » et « incidence du microbrowser » pour ce qui est de la partie du terminal. Pour accéder aux différents formulaires, il faut les sélectionner en suivant le chemin : barre d'outil, formulaire, sélectionner et puis le nom.

## 1. Formulaire : « Définition de l'information »

Ce formulaire reprend la caractérisation de l'information comme vu au point 2.1.2.2. Notons qu'étant donné le caractère exemplatif de l'arbre, seul les informations de types graphiques sont traitables par le programme. Le formulaire présenté ci-dessous a été complété pour l'exemple qui nous a accompagné tout au long de l'exposé. Les données relatives à l'exemple viennent du tableau présenté dans la section 2.1.2.2.A et de la section 2.1.2.3 pour ce qui est des données sur le fichier.

The image shows a screenshot of a software interface. The main window is titled '1. Décomposition du site en niveaux avec POSITIONNEMENT à ARRANGEMENT des objets par niveau: Site' and '2. Analyse des possibilités de transformation de l'information'. Below this, there are sections for 'A. Informations/Données' and 'B. Type de TERMINAL, voir programmes PROCÉDÉS des IIR'. The 'A. Informations/Données' section contains a table with the following structure:

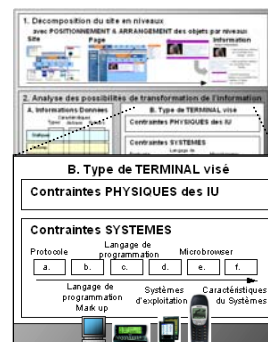
	Caractéristiques	
	Types de base	Fichiers
Grafiques		
Sonores		
Tactiles		

A callout box labeled 'A. Informations/Données' points to the 'Grafiques' row in the table.

Définition de l'information (En cours)		
Identificatif de l'information		
<b>Image Fille</b>		
Mode <b>Graphique</b>	<i>Une fois l'information complétée, ouvrir les formulaire "Définition terminal" ET "CONCLUSION" (Barre à outil, Formulaire -&gt; sélectionner -&gt; nom du formulaire)</i>	
<b>Mode Graphique</b>		
Type de l'information <b>Objet Graphique</b> Texte Elément Graphique	Propriétés de l'information <b>Fixes</b>	Format du fichier <b>.Jpeg</b> Type de compression (en MAJ) Taille de fichier (Kbits) <b>16</b>
<b>Caractérisation - définition de l'information en tant que telle</b>		
Pour objets ou éléments graphiques		
Taille objet source (LxH) <b>0155x0115 Px1s</b>	Taille Objet tel que représenté (LxH) <b>0107x0077 Px1s</b>	
<i>Taille objet-source (L) (Px1s)</i> <b>0155</b>	<i>Taille objet tel que représenté (H) (Px1s)</i> <b>0107</b>	
<i>Taille objet-source (H) (Px1s)</i> <b>0115</b>	<i>Taille objet tel que représenté (L) (Px1s)</i> <b>0077</b>	
Résolution (L) <b>0075 DPI</b>	Résolution (H) <b>0075 DPI</b>	
Couleurs (Général: pour tout type)		
Palette de couleurs (bits) <b>24</b>		
Type de représentation de couleur <b>RGB</b>		
Texte (UNIQUEMENT)		
Type de la Police (en MAJUSCULES) <b>ARIAL</b>	Effets appliqués à la police	
Taille de la police <b>14 Points</b>	<input type="checkbox"/> Souligné	<input type="checkbox"/> Outline
Style de la police <b>Normal</b>	<input type="checkbox"/> Double souligné	<input type="checkbox"/> Ombré
Couleur de la police (RGB) <b>R:102 G:153 B:255</b>	<input type="checkbox"/> Exposant	<input type="checkbox"/> Embross
	<input type="checkbox"/> Indice	<input type="checkbox"/> Engrave
		<input type="checkbox"/> Barré
		<input type="checkbox"/> Petites Majuscules
		<input type="checkbox"/> Tout en Majuscules
<b>Mode Sonore (à titre d'exemple de la suite)</b>		
Type d'information (sonore)	Propriété de l'information (sonore)	Format du fichier (sonore) ...

## 2. Formulaire : « définition du terminal »

Les données reprises dans ce formulaire sont celles présentées dans la section 2.2 du Modulateur. Pour illustrer ce formulaire nous avons décidé d'appliquer les informations (cf. formulaire précédent) à un Nokia 6210. Le formulaire complété avec les données relatives à ce terminal est dans le fichier ModN6210.ovd.



**Définition du terminal [Complet]**

Type de terminal

**Téléphones Mobiles**

Hybride

PDA - Palm Sized

PDA - Handheld PC

Modèle du terminal visé: **Nokia 6210**

**!!! Si vous changez de terminal visé, effacez le contenu des formulaires "incidence du Langage de Marquage" et celui de "l'incidence du Microbrowser"**

Contraintes Système

Protocole - Langage de Marquage \* (OK = sous-formulaire complété)

Système d'exploitation - Microbrowser\* (OK = sous-formulaire complété)

Caractéristiques du Système

Processeur (type)	
Mémoire RAM (! Kbit !)	<b>150</b>
Espace de stockage (! Kbit !)	<b>0</b>
Modem (vitesse max de transmission en Kbit/s)	<b>9,6</b>

Carte Son

Carte Graphique

Carte Graphique: Nombre de couleurs supportées (Bits) **16**

Contraintes Physiques des IU

**Ecran (Mode Graphique,terminal)**

Taille écran (HxL)(Pouce) **02 96x01 85Inch**

*Taille écran (L) (en pouce)* **02 96**

*Taille écran (H) (en pouce)* **01 85**

*Optionnel (non utilisé dans cette version du*

Taille Ecran (diag en Pouce) **1,57**

Rapport écran (L:H) **47:30**

Résolution de l'écran en DPI **0032x0048 DPI**

Résolution écran (LxH) **0096x0060 Px15**

*Résolution écran (L) (Pixels)* **0096**

*Résolution écran (H) (Pixels)* **0060**

Ecran couleur

**Enceintes (Mode Sonore,terminal)**

\* Si le formulaire était relié à une base de donnée, il serait possible de juste choisir (ou ajouter) le nom du Langage de marquage ou du Microbrowser. Les champs seraient directement ajustés. Ici nous devons les réintroduire à chaque fois.

Notons que les catégories « Protocole – Langage de marquage » et « Système d’exploitation – Microbrowser » font appel à des formulaires qui les définisse. Ces sous-formulaires sont appelés automatiquement lorsqu’on introduit les données pour la première fois. S’il s’agit de

modifications, il faut les appeler manuellement.

## 2.1. « Incidence du Langage de marquage »

La copie d'écran ci-dessous, traite le cas du WML.

**Incidence du Langage de marquage [Complet]**

LM - Nom du langage  
 HTML  
 WML

LM - supporte Mode Graphique       LM - supporte mode sonore

En terme de types de fichiers

LM (G)- supporte type statique       LM (S)- supporte type statique  
 LM (G)-supporte type dynamique       LM (S)- supporte type dynamique

LM - supporte les objets graphiques  
 LM - supporte les éléments graphiques       LM - supporte les types de fichiers-sons

Formats de fichiers supportés par le langage de marquage (X=oui) (1->4 = du plus volumineux au moins volumineux)

LM - Format Fichier 1	LM - Format Fichier 2
LM - Format Fichier 3	LM - Format Fichier 4 <b>.WBPM</b>

LM - Admet la compression des fichiers acceptés

Caractéristiques max parmi les Formats supportés  
 LM - Palette de couleurs (bit)      **1**

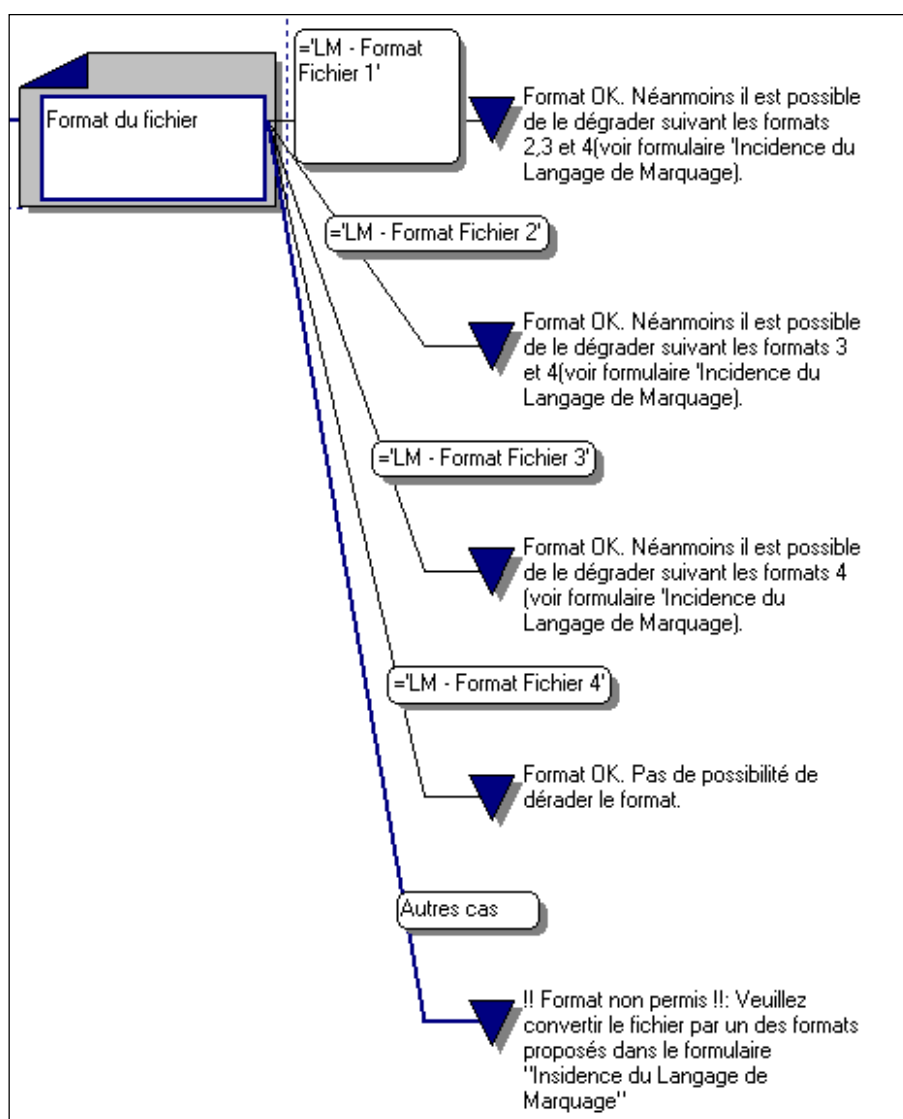
REMARQUE: En principe le formulaire est relié à une base de données. Il suffit alors de poser une requête du type: si extension du fichier est dans les extensions supportées par le langage de marquage, alors accepte et montre les extensions possibles pour ce type de fichier. De plus pour chaque type de fichier les caractéristiques ainsi que les compressions possibles seraient définies de même que le mode et le type d'information correspondent. Comme, ici, ce n'est pas le cas, on se limite à donner quatre fichiers supportés et à dire si oui ou non la compression de fichier est supportée. La définition des types est limitée de la même manière

**==> Tous les champs de ce formulaire (à l'exception des éléments graphiques supportés) font référence aux types de fichiers**

Lorsque vous avez fini de compléter ce formulaire, cliquez "oui"       Oui     Non

Comme nous l'avons vu plus haut et est mentionné dans la remarque du cadre, il ne s'agit pas de reprendre, ici, toutes les caractéristiques du langage de marquage, mais seulement les conclusions auxquelles ces caractéristiques nous amènent. Par exemple, la rubrique « Format de fichier... » devrait en principe reprendre tous les formats de fichiers et donner automatiquement leurs propriétés (ex. .gif permet un maximum de 256 couleurs, permet la transparence des couleurs,...) . Il devrait aussi les classer du plus « large » (et le plus lourd en taille) au plus « léger » et suivant les possibilités qu'ils offrent. Dans cette classification un fichier *bmp* viendrait avant un *jpeg* qui lui-même précéderait un fichier *gif*. Dans le programme, nous devons nous-même entrer les fichiers dans l'ordre et donner le nombre de

couleurs du fichier dans lequel nous allons convertir l'information de base. Cette manière de procéder permet de définir si oui ou non il y a moyen de dégrader le format du fichier de l'information. En effet, si le format initial occupe une position 1, 2 ou 3, on peut le dégrader suivant respectivement le format 2, 3 ou 4 ; 3 ou 4 et 4 pour le cas où il occuperait la troisième solution. S'il occupe la place 4, on ne peut pas dégrader cette information car elle est déjà dans le format le plus restrictif accepté par ce langage. Finalement, si le format initial ne fait pas partie de la liste, on doit le convertir au format qui correspond le mieux aux possibilités du terminal. Dans l'arbre de décision, pour l'exemple traité (image fille.jpeg à afficher sur un Nokia 6210 (WML) qui ne supporte que des images en WBPM), cette argumentation se traduit visuellement comme suit : suivre la ligne bleue





## 2.2. « Incidence du Microbrowser »

Comme pour le point précédent, ce formulaire est un sous-ensemble du formulaire portant sur la définition des terminaux. On s'intéresse ici aux spécificités du Microbrowser. Les remarques sur la précision et la complétude des informations traitées faites aux points précédents sont également valables ici. Voici l'écran rempli pour le cas du Nokia 6210.

**Incidence du Microbrowser [Complet]**

MB - Nom du MicroBrowser  
**Nokia 6210**

Il faudrait savoir par browser quel langage il est capable de compiler et quelles sont les caractéristiques de celui-ci qu'il n'assimile pas. De plus, il faudrait voir quels types de fichiers il peut lire pour pouvoir dire quels sont les modes qu'il supporte. Dans ce cas, nous nous limitons uniquement à la conclusion.

MB - Mode graphique       MB - Mode Sonore

MB - supporte les éléments graphiques  
 MB - supporte les objets graphiques (types de fichier)

MB - Scrolling vertical  
 MB - Scrolling horizontal

MB - supporte les formats-fichiers des sons réels  
 MB - supporte de formats-fichiers des sons synthétiques  
 MB - a un programme de lecture de texte par voix de synthèse

Lorsque vous avez fini de compléter ce formulaire, écrivez "oui"

Oui  
 Non

## 3. Conclusions

Une fois les données remplies, le simulateur de l'arbre nous donne les solutions ou les conseils nécessaires à suivre en vue d'obtenir l'ensemble des représentations possibles de l'information pour un terminal donné. Si on ouvre le « formulaire conclusion » en même temps que les formulaires précédents, il est possible de voir comment les conclusions varient en fonction des options choisies. La mise à jour des conclusions est immédiate et automatique. En cliquant deux fois sur un cadre du formulaire, le programme donnera l'emplacement de la réponse présentée dans le cadre sur l'arbre. La figure précédente nous montre le cheminement sur l'arbre lorsqu'on sélectionne la case intitulée : « conclusion – type

de fichier ».

Le cadre ci-dessous nous montre les conclusions obtenues pour l'affichage de l'image « fille » sur un Nokia 6210.

CONCLUSION: Ensemble des possibilités pour représenter une information sur un terminal donné (En cours)		
Donne le "Panier" (ou intervalle) possible pour chaque caractéristique		
Identificatif de l'information <b>Image Fille</b>	Modèle du terminal visé: <b>Nokia 6210</b>	
LM - Nom du langage <b>WML</b>	Conclusion - Type d'information Possible <b>Type texte + objets graphiques</b>	Conclusion - Propriété de l'information <b>Graphique Statique ou Dynamique</b>
Conclusion - Mode <b>Mode Graphique</b>		
Conclusion - Type de Fichier <b>!! Format non permis !!: Veuillez convertir le fichier par un des formats proposés dans le formulaire "Insidence du Langage de Marquage"</b>	Conclusion - Taille du Fichier <b>Taille Max = Taille du fichier "source"</b>	Conclusion - Type de compression <b>Compression non valide. Veuillez enregistrer le fichier dans un format non compressé.</b>
REM: Si l'on a deux valeurs, il faut garder la proportion de l'objet en se basant sur le chiffre le plus petit. Sinon, les chiffres vous donnerons l'image maximale mais sans tenir compte du rapport L:H initial.	Conclusion - Taille de l'objet MAX en Largeur (Pxls) <b>0096</b>	Conclusion - Résolution MAX en Largeur
	Conclusion - Taille de l'objet MAX en Hauteur (Pxls) <b>Adapter la hauteur de l'image en fonction de sa Largeur</b>	Conclusion - Résolution MAX en Hauteur
<b>Couleurs</b>		Conclusion - Couleur MAX (bits) (SOLUTION) <b>Le nombre de couleurs que vous désirez afficher est supérieur aux possibilités du format-fichier supporté par le langage de marquage. Veuillez adapter la palette de couleurs ou changer de format de fichier</b>
Conclusion - Couleur MAX (bits) (CONTRAINTE SYSTEME) <b>1</b>		
Conclusion - Couleur: RENDU A l'ECRAN <b>Monochrome</b>		
<b>Texte</b>		
Conclusion - Taille police <b>WML permet trois tailles: big - normal - small</b>	Conclusion - Couleur Police <b>Noir et blanc</b>	
Conclusion - Style Police <b>WML permet trois style: gras - normal - italique</b>	Conclusion - Effets sur Police <b>WML permet comme effet l'emphasis, "strong" et souligné</b>	

## Table des règles

- Règle 1. Lors de la création de l' « interface de base » prendre la forme d'information la plus large possible (ex. image en bmp...). Il est toujours possible de dégrader un format, l'inverse ne l'est pas. .... 22
- Règle 2. Lors de l'édition d'un site, organiser l'information suivant un ordre d'importance. Cette hiérarchisation est faite sur base d'un système d'annotations ou de tags..... 27
- Si aucun renseignement n'est donné à ce sujet, le modulateur prendra comme ordre d'importance l'ordre des niveaux. Par exemple, pour le mode graphique, au sein de chaque niveau, l'ordre d'importance des objets est fixé arbitrairement sur un critère d'usage. Le modulateur découpe la page en zones (ou suivant les éléments graphiques comme des tableaux) et considère comme la plus important la plage centrale. Ensuite, il classe les objets restants en partant du coin supérieur gauche de l'écran en essayant de distinguer des objets de navigation (ex. menu bouton...). Au dernier niveau, les objets graphiques et le texte sont classés sur le critère de la taille. .... 27
- Règle 3. Toujours illustrer une représentation linguistique par une représentation analogique (ex. donner la description d'un tableau, et en montrer une reprographie) et inversement. (cf. infra) ..... 30
- Règle 4. Organiser et doubler l'information de façon hiérarchique suivant le degré d'importance et, si possible, par plusieurs types de fichiers (sons réels (wav, au,...) et synthétiques (midi). De cette façon, le modulateur pourra donner des préférences quant au fichier à utiliser pour représenter une information suivant le terminal visé. .... 54
- Règle 5. Pour tout texte donner des informations de lecture pour la voix de synthèse. Dans le cas où il n'y a pas d'information, c'est la voix « standard » définie par l'utilisateur (si le programme de lecture est au niveau du terminal) ou le designer de l'interface (si la lecture se fait en amont)..... 54
- Règle 6. Lors de la programmation de la page source, toute information doit être présentée suivant les trois modes (graphique, sonore et tactile (si possible)) en la dédoublant de sorte que toute information linguistique soit annotée/illustrée par une information analogique ou arbitraire, et inversement. (ex.:annoter/expliquer une image avec du texte, pour le cas d'une alarme il faudrait la représenter par une image, un son, un signal tactile (vibrations) et l'accompagner d'une explication écrite et « parlée ») ..... 57
- Règle 7. Sauf usage particulier (zoom), ne jamais afficher une image avec une taille supérieure à celle de l'écran. En cas d'absence de possibilité de scrolling cette règle devient impérative. .... 71
- Règle 8. Dans les cas où la vitesse d'accès à l'information est un facteur prépondérant dégrader les caractéristiques de l'information de manière à ce qu'elles soient inférieures ou égales aux capacités de l'IU du terminal. .... 71
- Règle 9. Si le microbrowser ne permet pas de scrolling, la taille maximale de l'objet (page, objet ou information) ne pourra être supérieure à la taille de l'écran ; Si le microbrowser admet un scrolling uniquement vertical ou horizontal, la taille maximale de l'objet sera ajustée

respectivement soit horizontalement soit verticalement à l'écran ; Si le  
 browser permet un scrolling vertical et horizontal, la taille de l'objet n'est pas limitée.  
 Néanmoins attention à la R7. 79

Règle 10. Dans le cas du modulateur, nous prendrons comme règle réductrice que la taille de  
 l'ensemble de la page (somme de la taille du code et de l'ensemble des objets affichés)  
 soit inférieure à la mémoire disponible du terminal. .... 80

Règle 11. Le nombre de couleurs de l'information ne peut être supérieur à celui supporté par la  
 classe graphique. .... 80

### III. Bibliographie

#### Sites Internet :

- [1] Berst, J., *Why the world is going mobile*, ZD Net, 12 décembre 1998. Accessible à [http://www.zdnet.com/anchordesk/story/story\\_2909.html](http://www.zdnet.com/anchordesk/story/story_2909.html)
- [2] *IBM Application Framework for e-Business*, Mobile enablement, IBM Corp., 1999. Accessible à [www.software.com/ebusiness/mobile.html](http://www.software.com/ebusiness/mobile.html)
- [3] Juan F. Lopez ([juan@isi.edu](mailto:juan@isi.edu)), Pedro Szekely ([szekely@isi.edu](mailto:szekely@isi.edu)), *Position Paper: Transforming the UI for anyone, anywhere Automatic Web Page Adaptation*, University of Southern California - Information Sciences Institute.
- [4] Mark Zohar, *The Dawn of Mobile eCommerce*, **Forrester**, 10-1999, [www.forrester.com](http://www.forrester.com)
- [5] Matthew R.Sanders, *Global eCommerce Approaches Hypergrowth*, **Forrester**, 18 avril 2000 [www.forrester.com](http://www.forrester.com)
- [6] OVUM, *Press white Paper: WAP Market Strategies*, Michiele MacKenzie, Mary Ann O'Loughling, Mai 2000, Accessible à [www.phone.com](http://www.phone.com)
- [7] Siemens, (*descriptif des téléphones*), <http://www.siemens.be>
- [8] *The Business Implications of Mobile Computing and Sybase's Complete Solution*, Sybase, Inc., 1998. Accessible à <http://www.sybase.com/products/system11/workplace/mobile.html>
- [9] Vincent Alzieu, *Comparatif: Les organisateurs électriniques*, , 5/10/2000. Accessible à <http://www.hardware.fr/html/artices/imprimer.php3?article=262>
- [10] Vincent Alzieu, *Handspring Prism et Platinum*, Hardware.fr, 25.10.2000. Accessible à <http://www.hardware.fr/html/artices/lire.php3?article=272&page=1>
- [11] Vincent Alzieu, *Hardware Palm m100*, Hardware.fr, 25.10.2000. Accessible à <http://www.hardware.fr/html/artices/lire.php3?article=272&page=1>
- [12] Vincent Alzieu, *L'EDGE: photo, détail, vente*, Hardware.fr, 12.10.2000. Accessible à <http://www.hardware.fr/hm/news/index.php3?date=12-03-2000#3111>
- [13] White paper EHPT, *How to make money with NextGen Wireless Services*, 2000 ([www.ehpt.com](http://www.ehpt.com)) accessible via [www.ericsson.com](http://www.ericsson.com)
- [14] ZDNet, Page Web, *Descriptif technique de tous les PDA*, ZDNet <http://www.zdnet.com/supercenter/stories/specs/>

#### Fabricants de Terminaux - OS

##### Ericsson, <http://www.ericsson.com>

[15] *Ericsson R320 User manual*, accessible à <http://www.ericsson.com> nom du fichier "GDP R320 Esur Manual EN 330105.pdf"

Ericsson Descriptif des terminaux, voir site <http://www.ericsson.com>

##### Helwet Packard [www.hp.com](http://www.hp.com)

[16] HP, *HELPFUL FACTS ABOUT PERSONAL DIGITAL ASSISTANTS*, [www.hp.com/jordana](http://www.hp.com/jordana)

##### Microsoft

[17] Microsoft Mobile Explorer :Key Features, [www.microsoft.com/mobile/phones/mme/default.asp](http://www.microsoft.com/mobile/phones/mme/default.asp)

[18] Microsoaf Pocket PC Features, [www.microsoft.com/mobile/pocketpc/bguide/features/default.asp](http://www.microsoft.com/mobile/pocketpc/bguide/features/default.asp)

[19] Microsoft Mobile phones, [www.microsoft.com/mobile/phones/default.asp](http://www.microsoft.com/mobile/phones/default.asp)

[20] Microsoft Wireless Data Solution, [www.microsoft.com/ISN/misc/wirelessdata.asp](http://www.microsoft.com/ISN/misc/wirelessdata.asp)

[21] Microsoft WindowsCE 3.0 Documentation, [www.microsoft.com/windows/embedded/ce/](http://www.microsoft.com/windows/embedded/ce/)

Microsoft, *Creating Mobile Windows DNA Applications (Step-by-step guides)* [www.microsoft.com/mobile/pocketpc/stepbystep/dnappart1](http://www.microsoft.com/mobile/pocketpc/stepbystep/dnappart1)

##### Nokia [www.nokia.com](http://www.nokia.com)

[22] Nokia, (*descriptif des téléphones mobile*), <http://www.nokia.com/phones/>

- [23] *Nokia WAP Toolkit (v2.1), Developer's Guide*, 01-2001, <http://www.forum.nokia.com>, product number: SDK-01-000-002
- [24] *Nokia's Communicators WWW Browser Style Guide*©, Nokia Mobile Phones 1999, 21.04.99
- [25] *Nokia WAP toolkit*,
- [26] *Nokia WAP Browser Client 2.0 (Product Overview)*, Septembre 2000
- [27] Alamgir Farouk, *Feature-based Device Description and Content Annotation*, Nokia, W3C Workshop on Web Device Independent Authoring <http://www.w3.org/2000/10/DIAWorkshop/> Wednesday 4 october 2000

#### **Palm ([www.palm.com](http://www.palm.com))**

- [28] Palm File Format Specification, 17.01.2000
- [29] Palm OS Programmer's Companion
- [30] Palm OS® Software Description 3.5 [www.palmos.com/dev/tech/docs/palmos35](http://www.palmos.com/dev/tech/docs/palmos35)
- [31] WebClipping Developer's Guide

#### **Symbian**

- [32] Symbian Platform v6.0 Product Support [www.symbiadevnet.com/downloads/sdks/v6sdks](http://www.symbiadevnet.com/downloads/sdks/v6sdks)
- [33] Symbian, *The Symbian Platform v6.0: Power and Innovation*, Technical Paper [www.symbian.com/technology/v6-papers/sp/papers-sp.html](http://www.symbian.com/technology/v6-papers/sp/papers-sp.html)

#### **Recherches académiques et autres**

- [34] Computer Department and LRDC, University of Pittsburg “*Visual Prompts and Graphical Design: A Framework for exploring the Design Space of 2D Charts and Graphs*” (1997, Copyright [www.aaai.org](http://www.aaai.org))
- [35] Costin Pribeanu°, Quentin Limbourg\*, Jean Vanderdonckt\*, *Task Modelling for Context-Sensitive User Interfaces*, National Institute for °Research and Development in Informatics, \*Université catholique de Louvain – Institut d'Administration et de Gestion
- [36] D.L.Cuomo, E.Jablonka, J.N.Mosier, « Data Entry Interaction Techniques for Graphical User Interface », (HFES 1996)
- [37] J.D.Gould, S.J.Boies, A.Meluson, M.Rasamny, A-M.Vosburgh, “*Entry and Selection Methods for Specifying Dates*”, IBM Research Center (NY-USA) (1989)
- [38] Jean Vanderdonckt, *Conception d'applications de gestion sensibles au contexte*, Présentation lors du Séminaire à l'Institut d'Administration et de Gestion le 4 décembre 2000.
- [39] Levy, E. Schiano, D.J., Zacks, J., Tversky, *Representing information in graphs*, Paper presented at the meeting of the Psychonomics Society Los Angeles (1995), CA
- [40] Pawan Vora, Universal Access, XOR, Inc., Présentation du 4 juin 2001
- [41] S.L.Greene, J.D.Gould, S.J.Boies, A.Meluson, M.Rasamny “Entry-Based vs. Selection-Based Interaction Methods”, IBM Research Center (NY-USA) (1988)

**Amodeus:** Esprit Basic Research Action 7040 (06-1992 à 10-1995)

([www.mrc-cbu.cam.ac.uk/amodéus](http://www.mrc-cbu.cam.ac.uk/amodéus)).

**Amodeus : Taxonomy of Modality**, Tous les documents ci-dessous (avec leur résumé) sont accessibles à [www.mrc-cbu.cam.ac.uk/amodeus/tm](http://www.mrc-cbu.cam.ac.uk/amodeus/tm)

- [42] Bernsen Niels Ole CCS, TM/WP02: *MODALITY THEORY IN SUPPORT OF MULTIMODAL INTERFACE DESIGN*, Esprit Basic Research project AMODEUS-2 Working Paper TM-WP02, (CCS:Centre for Cognitive Science, Roskilde University)
- [43] Bernsen Niels Ole, *Why are Analogue Graphics and Natural Language both Needed in HCI?*, Esprit Basic Research project AMODEUS-2 Working Paper TM-WP03
- [44] Bernsen Niels Ole, *FOUNDATIONS OF MULTIMODAL REPRESENTATIONS, A Taxonomy of Representational Modalities*, Esprit Basic Research project AMODEUS-2 Working Paper TM-WP04
- [45] Bernsen, N.O., Lu, S. and May, M. (1994): Towards a design support tool for multimodal interface design. The taxonomy workbench and theory demonstrator. *Esprit Basic Research project AMODEUS-2 Working Paper RP5-TM-WP5*.
- [46] Steven Verjans, A thesis for the Master of Artificial Intelligence and Cognitive Science Study Programme at the K.U.Leuven: *Computer semiotics as a basis for integrating different views on Human Computer Interaction*, 1993-1994, Esprit Basic Research project AMODEUS-2 Working Paper TM-WP10
- [47] Bernsen, N.O. (1994c): *A revised generation of the taxonomy of output modalities*. Esprit Basic Research project AMODEUS-2 Working Paper RP5-TM-WP11, 1994.

- [48] TM-WP12, *Taxonomy WorkBench*, Esprit Basic Research project AMODEUS-2 Working Paper TM-WP12
- [49] Steven Verjans, *State-of-the-art for Input Modalities*, 06-1995, Esprit Basic Research project AMODEUS-2 Working Paper TM-WP20
- [50] Bernsen, N.O., *A TOOLBOX OF OUTPUT MODALITIES, Representing Output Information in Multimodal Interfaces*, Esprit Basic Research project AMODEUS-2 Working Paper TM-WP21
- [51] Bernsen, N.O. and Verjans, S. (1995): *Designing interfaces by information mapping. A case study of the CERD design*. Esprit Basic Research project AMODEUS-2 Working Paper (forthcoming).

#### **Taxinomy of Modality (autres)**

- [52] Bernsen, N.O. and Bertels, A. (1993): *A methodology for mapping information from task domains to interactive modalities*. Esprit Basic Research project GRACE Deliverable 10.1.3.

#### **Langages de programmation**

- [53] "Extensible Markup Language (XML) 1.0", W3C Recommendation 10 February 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>
- [54] "HTML 4.01 Specification", W3C Recommendation, 24 December 1999, D. Raggett, A. Le Hors, I. Jacobs, on-line at <http://www.w3.org/TR/1999/REC-html401-19991224>
- [55] "Voice eXtensible Markup Language (VoiceXML™) version 1.0", W3C Note 05 May 2000, <http://www.w3.org/TR/voicexml/>
- [56] "XHTML TM 1.0: The Extensible HypterText Markup Language," W3C Proposed Recommendation, 8/99, <http://www.w3.org/TR/PR-xhtml1-19990824.html>.
- [57] Jacob Eisenstein, *XIML: The eXtensible user-Interface Markup Language*, RedWhale Software Internal Papers, November 18, 1999
- [58] Lassila, [RDF] "Resource Description Framework (RDF) Model and Syntax Specification," W3C Recommendation, 2/99, <http://www.w3.org/TR/1999/PR-rdf-syntax-19990105>.

#### **WAP – WML ([www1.wapforum.org](http://www1.wapforum.org))**

- [59] (toutes la documentation descriptive du langage est disponible en downloadant de fichier "WAP Technical June 2000-20001011.zip" à partir du site [www.ericsson.com](http://www.ericsson.com)
- [60] "Wireless Markup Language Specification, version 1.3", 19 February 2000  
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-191-WML-20000219-a.pdf>

#### **CC/PP**

- [61] Wireless Application Group User Agent Profile Specification (WAGUAPROF), Wireless Application Protocol Forum, Ltd. 1999, 10.11.1999 accessible à [www.ericsson.com](http://www.ericsson.com) fichier "WAP Technical June 2000-20001011.zip"
- [62] Franklin Reynolds, Johan Hjelm, Spencer Dawkins, and Sandeep Singhal, [CC/PP] "Composite Capability/Preferences Profiles: A user side framework for content negotiation," W3C Note, 27 July 1999, <http://www.w3.org/TR/1998/NOTE-CCPP-19990727>. (Note that this specification refers to the 7/99 version of this specification, irrespective of the availability of future versions of that specification).

#### **Autres solutions existantes**

##### **AUIL**

- [63] Ruud Siebelink, *Towards a profile driven service authoring and adaptation platform*, Alcatel Bell, W3C Workshop on Web Device Independent Authoring <http://www.w3.org/2000/10/DIAWorkshop/siebelink.htm> Wednesday 4 October 2000

##### **UIML ([www.uiml.org](http://www.uiml.org) et [www.harminia.org](http://www.harminia.org))**

- [64] Marc Abrams and Constantinos Phanouriou, *UIML: an XIML Langage for building device-independent User Interface* (no réf.)
- [65] Marc Abrams and Constantinos Phanouriou, Alan L.Batongbacal, Steven M.Williams, Janathan E.Shuster, *UIML: An Appliance-Indépendent XML User Interface Langage*, 8<sup>th</sup> International WWW Conference, Toronto, mai 1999, accessible à <http://www8.org/w8-papers/5b-hypertext-media/uiml/uiml.html>
- [66] Marc Abrams, *Device-Independent Authoring with UIML*,

**Phone.com Inc. ([www.phone.com](http://www.phone.com))**

- [67] Paul Smethers, *WAP Application Usability In GSM Markets*, Présentation de phone.com au W3C Workshop on Web Device Independent Authoring <http://www.w3.org/2000/10/DIAWorkshop/> Wednesday 4 October 2000
- [68] Phone.com .Inc, *Application Style Guide:For GSM 900 and 1800 For markets with both Phone.com and Nokia Model 7110 WAP browsers(White Paper)*, 06-2000
- [69] Phone.com .Inc, *Enabling the Wireless Internet, Why You Should Develop Applications Using the Phone.com WAP Solution (White Paper)*, 01-2000
- [70] Phone.com .Inc, *HDML Style Guide (White Paper)*, 11-1999
- [71] Phone.com .Inc, *Interoperability & Compliance Testing (White Paper)*, 03-2000
- [72] Phone.com .Inc, *Mobile Management Architecture, Creating New Opportunities for Activation and Provisioning (White Paper)*, 11-1999
- [73] Phone.com .Inc, *WML Style Guide (White Paper)*, 08-1999

## Réseaux

- [74] Peter Lisle, GPRS Programme Manager, BTCCellnet Jim Donahue, BTCCellnet, *GPRS Network Programme Manager*, W3C Workshop on Web Device Independent Authoring <http://www.w3.org/2000/10/DIAWorkshop/> Wednesday 4 October 2000

## Accessibilité

- [75] Web Accessibility Initiative [www.W3C/WAI/](http://www.W3C/WAI/)

## Revues

- [76] Ciancetta, M.C., *Convergence Trends for Fixed and Mobile Services*, IEEE Pers. Com., avril 1999, pp. 14-21
- [77] [dany@sjynet.be](mailto:dany@sjynet.be), *La révolution du XML*, Revue : Inside Internet – Septembre 2000 (p74-76)
- [78] *Dossier, Le GSM et la communication de données*, Revue : Mobile World – Décembre 1999 (p10-16)
- [79] *La bataille pour lesystème d'exploitation mobile*, Revue : Mobile World – Décembre 1999 (p20-27)
- [80] Ludovic Gilles ([ludovicgilles@best.be](mailto:ludovicgilles@best.be)), *Dossier: MobileOffice*, Revue : Mobile World – Septembre 2000 (p6-7)
- [81] Ludovic Gilles ([ludovicgilles@best.be](mailto:ludovicgilles@best.be)), *MobileOS, Quoi de neuf pour nos portables*, Revue : Mobile World – Octobre 2000
- [82] *MCommerce :premier éléments*, Revue : Mobile World – Septembre 2000 (p5)



## **IV. Annexes**

***Annexe 1***

**Description et table comparative des différents langages potentiellement candidat à décrire une IU**

Le texte ci-dessous est extrait du texte : **Task Modelling for Context-Sensitive User Interfaces**

Costin Pribeanu<sup>‡</sup>, Quentin Limbourg\*, Jean Vanderdonck\*

<sup>‡</sup>National Institute for Research and Development in Informatics

Bd.Maresal Averescu Nr.8-10, 71316 Bucharest (Romania) [pribeanu@acm.org](mailto:pribeanu@acm.org),

\*Université catholique de Louvain – Institut d'Administration et de Gestion

Place des Doyens, 1 – B-1348 Louvain-la-Neuve (Belgium) {[@qant.ucl.ac.be](mailto:limbourg.vanderdonck)}

« ...

The **Phone.com** paradigm (<http://www.phone.com>) is intended to describe a UI dialog that will be rendered afterwards for fixed or mobile phone. It is therefore aimed at supporting phone-based applications, which is not the case of the **GeoWorks solution** (<http://www.geoworks.com>): it supports the same idea for multiple platforms. SISL [6] uses an event-based model of services for platform independence.

The **User Interface Markup Language (UIML)** is an XML-compliant language that allows a declarative description of a user interface in a device-independent manner [2,3]. To create a UI, one writes a UIML document, which includes presentation style appropriate for devices on which the UI will be deployed. UIML is then automatically mapped to a language used by the target device of a particular computing platform, such as HTML, Java/JFC, WML, HTML, PalmOS and VoiceXML.

The **Abstract User Interface Markup Language (AUIML)** [5] is an XML vocabulary designed to allow the intent of an interaction with a user to be defined. It allows device independent encoding of information. All the interaction information can be encoded once and subsequently rendered using device dependent rendering.

The **eXtensible User-interface Language (XUL)** [13] is an XML-based UI specification language specifically designed for network applications like browsers, mailers, etc. Thanks to the portability of its rendering engine Gecko and the use of its XPCOM Technology, a XUL definition can be rendered on several computing platforms.

The **eXtensible Interface Markup Language (XIML)** [2] is an extensible XML specification language for multiple facets of multiple models in a model-based approach. An XIML specification can lead to both an interpretation at runtime and a code generation at design time.

The **Vizzavi** ([www.vizzavi.com](http://www.vizzavi.com)) paradigm adopts another approach: rather than starting from initial models to produce a UI either at design time or at run-time, it reads a family of web pages and converts them on the fly to support other languages such as WML. Table 1 reports on some major differences noted between some of these solutions.

**Table 1.** Comparison of languages for context-sensitive user interfaces.

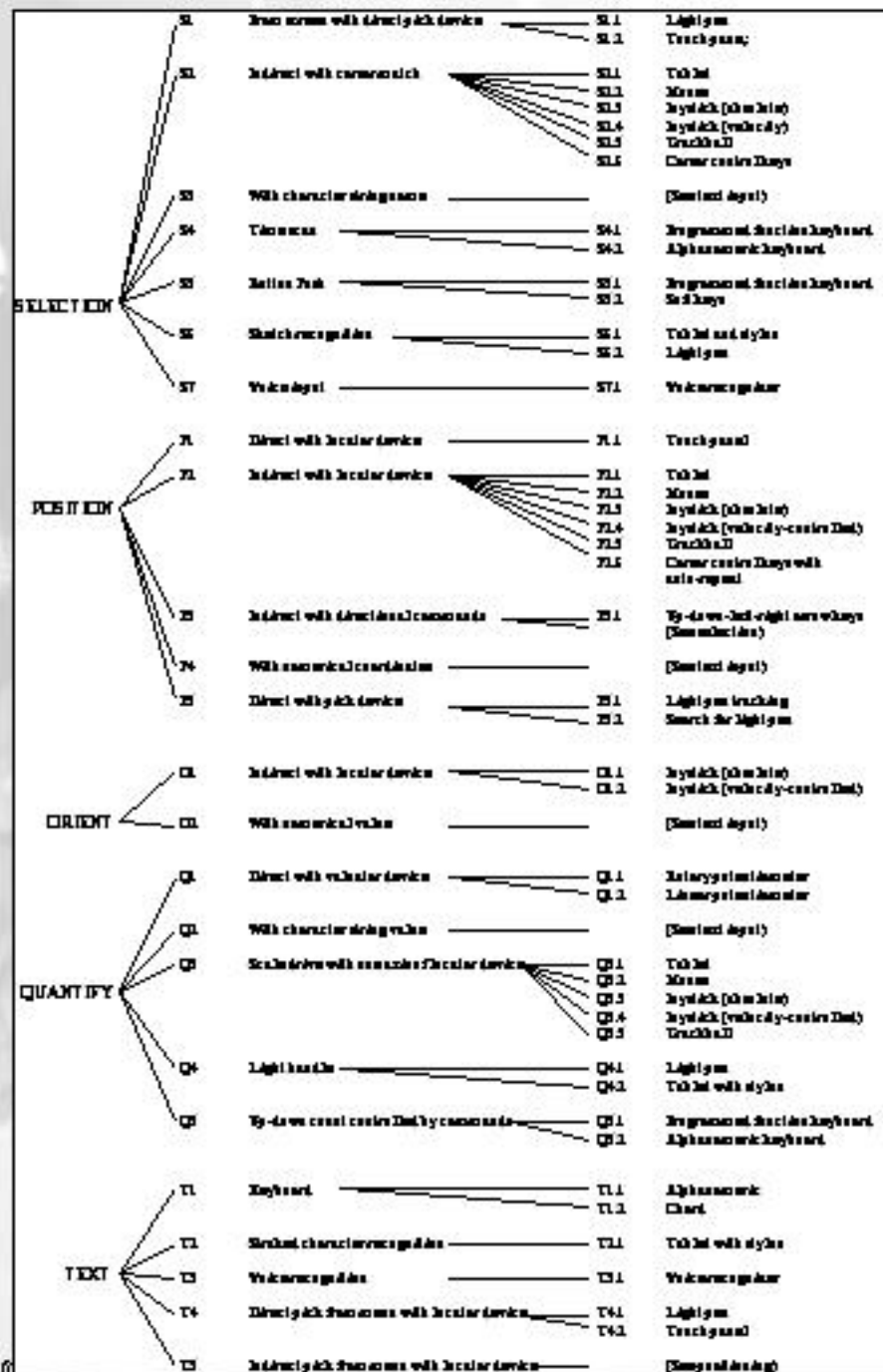
Language	Models	Output	Design Knowledge	Methodology	XML compliance	Supporting tool	Scope
Phone.com	Presentation Dialog	Vocal UI	None	None	No	Code generator	Phone-based applications
GeoWorks	Dialog Presentation	Fully functional interface	Partial	None	No	Rendering engine	none
UIML	Dialog Presentation Domain (parts)	Fully functional interface	None	None	Yes	Code generator	None
AUIML	Dialog Presentation	Fully functional interface	None	None	Yes	Rendering engine	None
XUL	Dialog Presentation	Fully functional interface	None	None	Yes	Rendering engine	Network specific applications
Vizzavi	Dialog Presentation	Fully functional UI	Transformation rules	Rule-based system	No	Conversion engine	Web pages
XIML	<b>Task Domain User Dialog Presentation Platform Design</b>	Fully functional interface	Design guidelines Heuristics	Model-based approach	Yes	Rendering engine Code generator Editor	None

...»

**Annexe 2**

e n d

# INPUT MODALITIES



Foley et al. (1984) describe a number of interaction tasks and techniques, and controlling tasks and techniques. They define an interaction task as a sentence in a formal language, e.g. the task "move entity" has three basic entities: a position, an entity and the imperative 'move'.

*The entry of each of these three entities is an interaction task, performed by means of an interaction technique. Each task can be implemented by many different techniques. The designers of the system must select interaction techniques that best match both the user's characteristics and the specific requirements of the task, and they must also select the appropriate device. In some cases the devices are predetermined, having been selected by the hardware procurers rather than by the user interface designers. This unfortunate situation reduces the number of options for the designers and may result in a suboptimal design.'*  
(Foley et al. 1984:17)

Fig. 6: Six main interaction task types and their related interaction techniques and devices

(Reproduced from Foley et al. 1984)

*The limitation of the Foley - Wallace - Chan scheme is that the categories, while reasonable, are somewhat ad hoc, and there is no attempt at defining a notion of completeness for the design space.*  
(Card et al., 1991: 100)

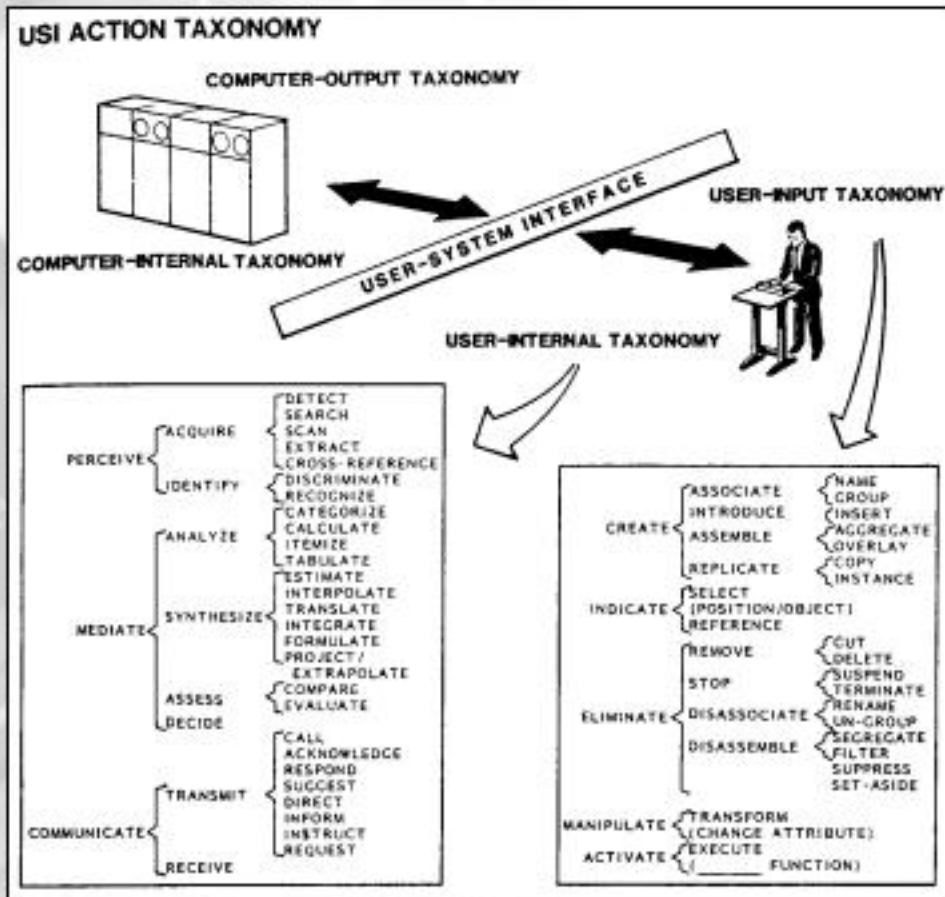
	Touch Screen	Light Pen	Graphic Tablet	Mouse	Track- ball	Joy- stick
Eye-hand coordina- tion	+	+	0	0	0	0
Unobstructed view of display	-	-	+	+	+	+
Ability to attend to display	+	0	+	0	+	+
Freedom from paral- lax problems	-	-	+	+	+	+
Input resolution ca- pability	-	-	+	+	+	+
Flexibility of place- ment within work- place	-	-	0	0	+	+
Minimal space re- quirements	+	+	-	-	+	+
Minimal training re- quirements	+	0	0	0	0	0
Comfort in extended use	-	-	0	0	+	+
Absolute mode capa- bility	+	+	+	-	-	0
Relative mode capa- bility	-	-	+	+	+	0
Capability to emulate other devices	-	-	+	-	-	-
Suitability for:						
pointing	+	+	+	+	+	-
rapid pointing	+	+	0	0	0	-
pointing with confir- mation	-	0	0	+	0	-
drawing	-	-	+	0	-	-
tracing	-	-	+	-	-	-
continuous tracking, slow targets	0	0	+	+	+	-
continuous tracking, fast targets	-	-	0	0	0	+
alphanumeric data entry	-	-	-	-	-	-

Table 3: Advantages and disadvantages of the standard pointing devices.

Legends: + Advantage, 0 Neutral, - Disadvantage

(From Greenstein & Arnaut 1988)

Another such type of study is the one by Greenstein & Arnaut (1988), in which they analyse input devices which designate locations and movements in space, including touch screen devices, light pens, graphic tablets, mice, trackballs and joysticks, as well as some more novel technologies. These devices are all reasonably well suited to pointing or selection among items on a display as well as the input of graphical information. They are not well suited to alphanumeric data entry. Greenstein & Arnaut consider both ergonomic considerations and comparative human performance and preference data, with the aim of offering support in the selection of an appropriate device for an application. Table 3 summarizes their study.



However, we found one interesting study (Lenorovitz et al. 1984) that tries to build a taxonomy of input actions, i.e. of the types of information that can be 'presented' to a computer.

The authors distinguish between four major sub-taxonomies of actions involved in User System Interaction (USI), i.e. taxonomies of terms used to characterize the activities and actions of users in a wide range of user-computer interactive applications.

Fig. 5: Taxonomy of User Interface Actions  
(From Lenorovitz et al. 1984)



*Figure 1. Tableau of Continuous Input Devices*

Number of Dimensions

		1	2	3				
Property Sensed	Position	Rotary Pot	Sliding Pot	Tablet	Light Pen	Joystick	3D Joystick	Mechanical
				Touch Tablet	Touch Screen			Touch Sensitive
	Motion	Continuous Rotary Pot	Treadmill Thumbwheel	Mouse		Trackball	Trackball	Mechanical
			TASA Penmat			TASA X Y Pad		Touch Sensitive
	Pressure	Torque Sensing	Pressure Pad			Isometric Joystick		

*Fig. 2: Tableau of continuous input devices (From Buxton 1983)*

The last study we want to highlight in this section (Bleser & Sibert 1990) tries to incorporate knowledge acquired from a variety of disciplines such as semiotics, ergonomics, and industrial design into a software environment which supports a designer in selecting an interaction technique. In the selection procedure special emphasis is given to the physical characteristics of input devices and the pragmatics of their use. Bleser & Sibert's tool (Toto) includes knowledge bases with:

1. a network of task concepts which describe physical actions for a large variety of primitive tasks like *touch*
2. a collection of task methods in the form of sequences of primitive actions,
3. an input device taxonomy which provides detailed descriptions of both primitive and composite devices,
4. a technique catalog which includes all predefined action sequences using the input devices,
5. the set of current design constraints such as device availability, and
6. a set of design heuristics taken from human factors guidelines.

Figure 7 shows the input device model used by the authors.

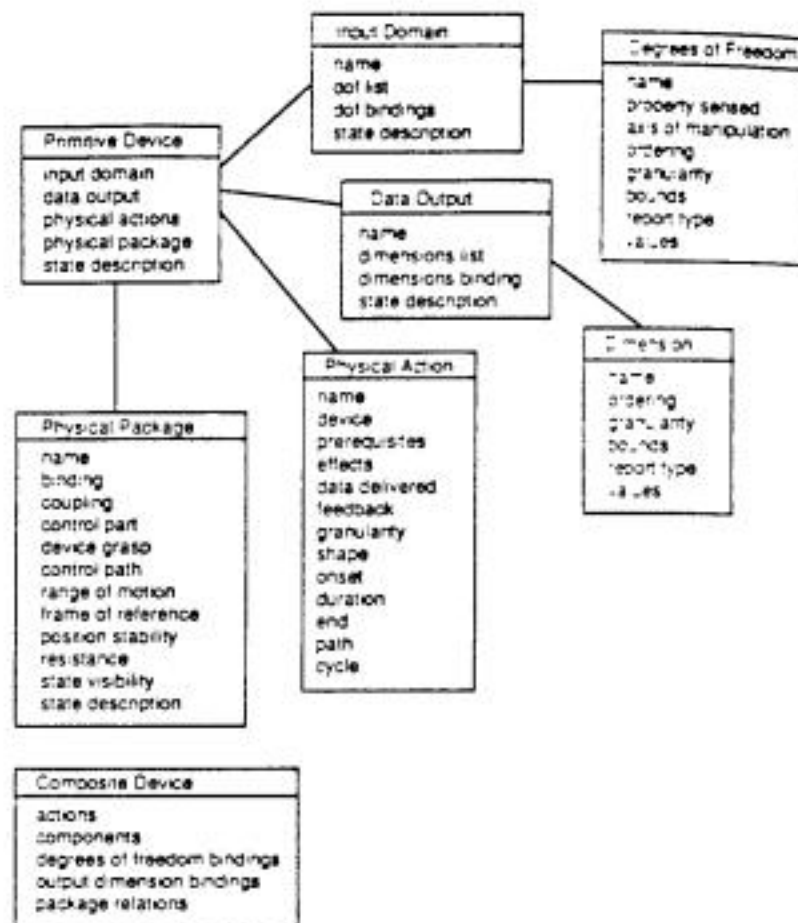


Fig. 7: Input Device Model from the Toto design decision aiding tool (Bleser and Sibert 1990:138)

***Annexe 3***

# Gratuitous Graphics? Putting Preferences in Perspective

*Quels type de graphique employer et quand ?*

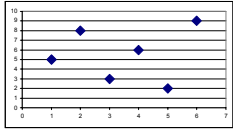
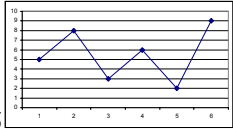
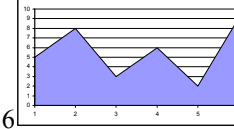
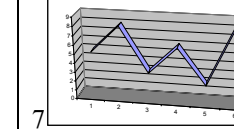
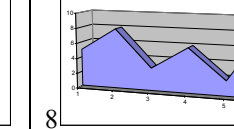
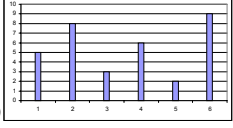
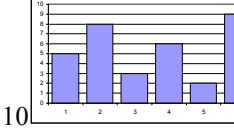
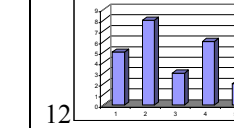

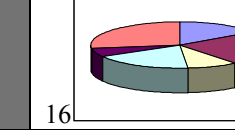
*L'importance de la perspective dans la présentation des objets et en particulier les graphiques sur UI.*

## Résumé

Des chercheurs de l'université de Cleveland ont étudié les capacités du système perceptif humain à extraire de l'information à partir de graphique suivant leur présentation (angle de vue, aires, longueur, volume,...). Suite à ces recherches, et à ses propres expériences le département de psychologie de l'université de Stanford en est venu aux conclusions que:

1. Les sujets ne sont pas indifférents à la présentation des graphiques
2. De façon systématique, certains types de graphiques sont plus appropriés à certains usages
3. Les sujets préfèrent les graphiques suggérant un volume 3D à ceux qui représentent une surface 3D (→ 8,12 > 7,11)

Ils ont différencié les types suivants:

	<i>Simple</i>	<i>Aire</i>	<i>Surface 3D</i>	<i>Volume 3D</i>
Points	1 	2	3	4
Lignes*	5 	6 	7 	8 
Barres	9 	10 	11	12 
Parts**	13	14 	15	16 

\* = points reliés

\*\* l'étude s'est surtout intéressée aux trois premières catégories

## Choix du type de graphique suivant les usages

Cette étude a été menée sur 110 étudiants en psychologie à l'université de Stanford.

Les chiffres du tableaux donnent la fréquence d'utilisation des différents types de graphiques suivant les scénarios mis en contradiction (en %).

<i>usage</i>	Simple		Aire 2D		Surface 3D		Volume 3D		<i>usage</i>
Personnel	39	16	35	32	1	8	25	44	Présentation, à d'autres personnes

Immédiat	17.4	7.5	48.5	37	6.9	15	27	40.1	Différé dans le temps
Tendance	35	22.9	40.5	33.3	11.3	16.6	13.2	27	Détails

usage	Barres		Lignes		usage
Immédiat	56.3	38.3	43.7	61.7	Différé dans le temps
Tendance	83.9	52	16.1	48	Détails

### Considérations à propos de la 3D

**3D > 2D :** lorsqu'on est dans une situation où l'on a besoin de **mémoriser** et/ou lorsque le graphique sert à une **présentation** à d'autres personnes.

→ plus approprié quand :

- détails (quantitatifs) > tendance
- mémoire > impression immédiate
- communication avec d'autres > besoin individuel

### Explications :

De manière générale, la 3D est utilisée pour des présentations car beaucoup considèrent que le fait d'enjoliver un graphique lui donne plus de poids dans le sens qu'il impressionne plus.

De façon plus scientifique, des études ont montré que la 3D donne au système visuel l'impression d'un espace plus réaliste par ressemblance avec des objets solides. Cela facilite la schématisation ainsi que la codification des données par le cerveau. La mémorisation en est donc plus rapide et l'on a l'impression que le graphique est plus parlant.

Néanmoins ces résultats sont à nuancer. Si l'on compare des versions analogues de graphiques en barres (aires 2D - volume 3D)<sup>1</sup>, il en résulte un effet pertinent de la dimension. Ajouter de la profondeur réduit la précision aussi bien en ce qui concerne l'estimation de l'ampleur que la *perceptual match task*. Mais cette différence de performance est faible et disparaît même lorsque le sujet doit faire son évaluation de mémoire car la *perceptual match task* n'agit que sur une mémoire à court terme.

<sup>1</sup> Levy, E. Schiano, D.J., Zacks, J., Tversky, B. (1995). Representing information in graphs, *Paper presented at the meeting of the Psychonomics Society* Los Angeles, CA [39]

***Annexe 4***

# WAG UAPROF

Version 10-Nov-1999

---

## Wireless Application Group User Agent Profile Specification

...

### 7.2 User Agent Profile Components

The schema for WAP User Agent Profiles consists of description blocks for the following key components:

**HardwarePlatform:** A collection of properties that adequately describe the hardware characteristics of the terminal device. This includes, the type of device, model number, display size, input and output methods, etc.

**SoftwarePlatform:** A collection of attributes associated with the operating environment of the device. Attributes provide information on the operating system software, video and audio encoders supported by the device, and user's preference on language .

**BrowserUA:** A set of attributes to describe the HTML browser application

**NetworkCharacteristics:** Information about the network-related infrastructure and environment such as bearer information. *These attributes can influence the resulting content, due to the variation in capabilities and characteristics of various network infrastructures in terms of bandwidth and device accessibility.*

**WapCharacteristics:** A set of attributes pertaining to WAP capabilities supported on the device. This includes

details on the capabilities and characteristics related to the WML Browser, WTA [WTA], etc.

Additional components can be added to the schema to describe capabilities pertaining to other user agents such as an

Email application or hardware extensions. See Section 7.5 for details.

Exemple du code:

```
<!-- ***** -->
<!-- ***** Component: HardwarePlatform ***** -->

<rdf:Description ID="BitsPerPixel">
  <rdf:type rdf:resource="http://www.w3.org/TR/PR-rdf-schema#Property"/>
  <rdfs:domain rdf:resource="#HardwarePlatform"/>
  <rdfs:comment>
    Description:  The number of bits of color or grayscale information per
                  pixel, related to the number of colors or shades of gray
                  the device can display.
    Type:        Number
    Resolution:  Override
    Examples:    "2", "8"
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="ColorCapable">
  <rdf:type rdf:resource="http://www.w3.org/TR/PR-rdf-schema#Property"/>
  <rdfs:domain rdf:resource="#HardwarePlatform"/>
  <rdfs:comment>
    Description:  Indicates whether the device's display supports color.
                  "Yes" means color is supported. "No" means the display
                  supports only grayscale or black and white.
    Type:        Boolean
    Resolution:  Override
    Examples:    "Yes", "No"
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="CPU">
  <rdf:type rdf:resource="http://www.w3.org/TR/PR-rdf-schema#Property"/>
  <rdfs:domain rdf:resource="#HardwarePlatform"/>
  <rdfs:comment>
    Description:  Name and model number of the device CPU.
    Type:        Literal
    Resolution:  Locked
    Examples:    "Pentium III", "PowerPC 750"
  </rdfs:comment>
</rdf:Description>
```



Représentation RDF :

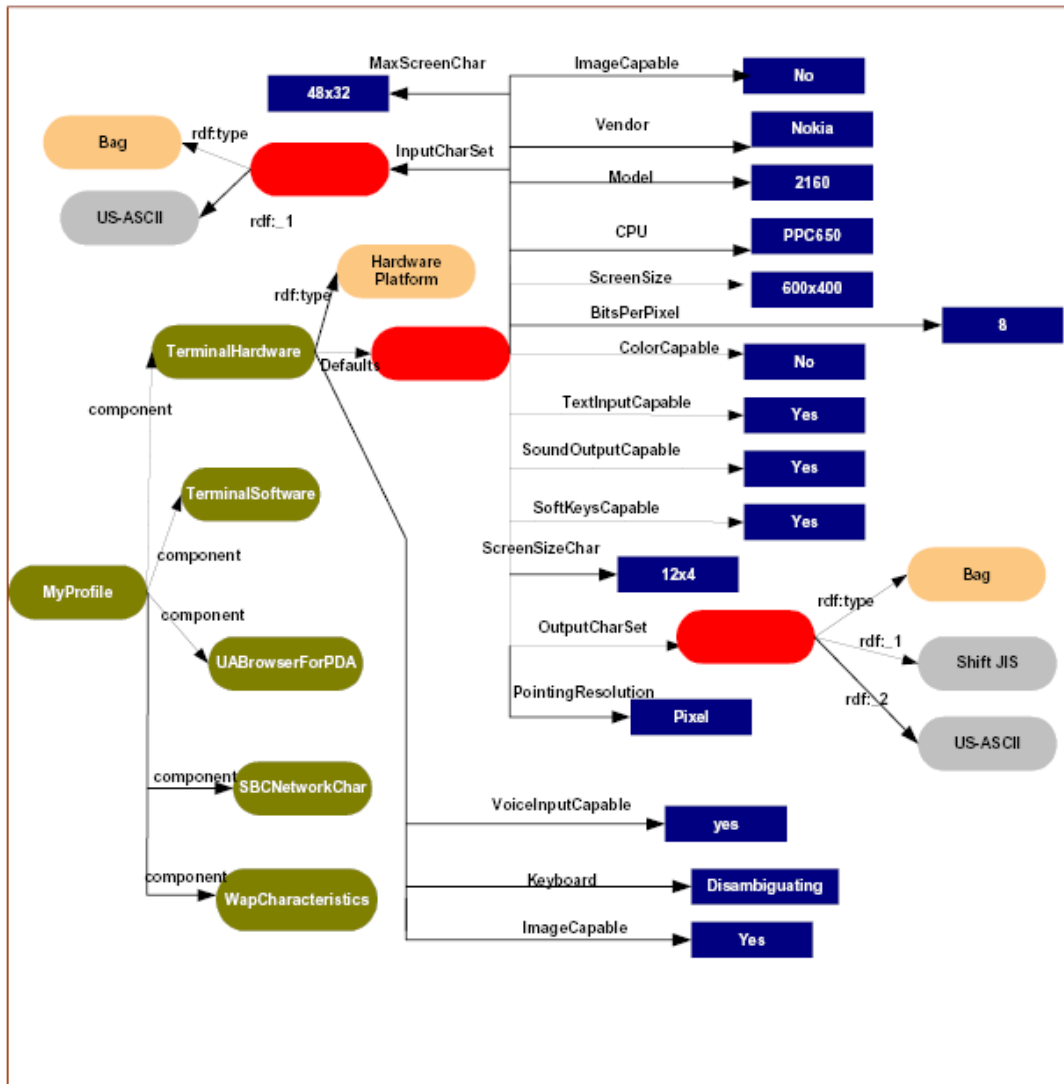


Figure 7.1: Root graph for Profile and Sub-graph for Hardware characteristics

exemple suite :

## A.1 Summary of User Agent Profile Schema

The table below summarizes the components and attributes defined within the WAP User Agent Profile schema (see Section 7.4). This section is informative.

<u>Attribute</u>	<u>Description</u>	<u>Resolution Rule</u>	<u>Type</u>	<u>Sample Values</u>
<b>Component: HardwarePlatform</b>				
BitsPerPixel	The number of bits of color or grayscale information per pixel	Override	Number	"2", "8"
ColorCapable	Whether the device display supports color	Override	Boolean	"Yes", "No"
CPU	Name and model number of device CPU	Locked	Literal	"Pentium III", "PowerPC 750"
ImageCapable	Whether the device supports the display of images	Locked	Boolean	"Yes", "No"
InputCharSet	List of character sets supported by the device for text entry	Locked	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift JIS"
Keyboard	Type of keyboard supported by the device	Locked	Literal	"Disambiguating", "Qwerty", "PhoneKeypad"
MaxScreenChar	Size of the virtual page onto which a document is rendered, in units of characters	Locked	Dimension	"16x80", "48x32"
Model	Model number assigned to the terminal device by the vendor or manufacturer	Locked	Literal	"Mustang GT", "Q30"
OutputCharSet	List of character sets supported by the device for output to the display	Append	Literal (bag)	"US-ASCII", "ISO-8859-1", "Shift JIS"
PointingResolution	Type of resolution of the pointing accessory supported by the device	Locked	Literal	"Character", "Line", "Pixel"
ScreenSize	The size of the device's screen in units of pixels	Locked	Dimension	"160x160", "640x480"
ScreenSizeChar	Size of the device's screen in units of characters	Locked	Dimension	"12x4", "16x8"
SoftKeysCapable	Indicates whether the device supports programmable soft keys	Locked	Boolean	"Yes", "No"
SoundOutputCapable	Indicates whether the device supports sound output	Locked	Boolean	"Yes", "No"
TextInputCapable	Indicates whether the device supports alphanumeric text entry	Locked	Boolean	"Yes", "No"
Vendor	Name of the vendor manufacturing the terminal device	Locked	Literal	"Ford", "Lexus"
VoiceInputCapable	Indicates whether the	Locked	Boolean	"Yes", "No"

***Annexe 5***

## Simplification de la typologie des modalités de sortie pour l'adapter au modulateur

### a) Mode Graphique

I. Linguistic	1. Static analogue sign graphic language	Microglyphs. Rarely used.	Included in 4 a-c and 5 a-c.	
	4. Dynamic analogue sign graphic language	Gestural language. Dynamic microglyphs would appear anachronistic.	4a. Static/dynamic gestural discourse	
			4b. Static/dynamic gestural labels/keywords	
			4c. Static/dynamic gestural notation	
	5. Static non-analogue sign graphic language	Letters, words, numerals, other written language signs, text, logograms (e.g. arrows), special-purpose notations (e.g., programming languages, formal logic, music). Sequential, list and tabular orderings.	5a. Static graphic written text	5a1. Static graphic typed text
				5a2. Static graphic hand-written text
			5b. Static graphic written labels/keywords	5b1. Static graphic typed labels/keywords
			5b2. Static graphic hand-written labels/keywords	
			5c. Static graphic written notation	5c1. Static graphic typed notation
				5c2. Static graphic hand-written notation
	8. Dynamic non-analogue sign graphic language	Dynamicizations of (5). Moving text, running numerical counters, digital clocks. Graphically viewed spoken language discourse (lip reading).	8a. Dynamic graphic written text	8a1. Dynamic graphic typed text
				8a2. Dynamic graphic hand-written text
8b. Dynamic graphic written labels/keywords			8b1. Dynamic graphic typed labels/keywords	
			8b2. Dynamic graphic hand-written labels/keywords	
8c. Dynamic graphic written notation			8c1. Dynamic graphic typed notation	
			8c2. Dynamic graphic hand-written notation	
8d. Graphic spoken text or discourse				
8e. Graphic spoken labels/keywords				
8f. Graphic spoken notation				
II. Analogue  <-li,an-ar>	9. Static analogue graphics	Pure photographs, naturalistic drawings, holograms	9a. Static graphic images	9a1. real-world picture
				9a2. non-real-world picture
		Pure diagrams, maps, cartoons. Sequential, list and tabular orderings.	9b. Static graphic maps	
	9c. Static graphic compositional diagrams			
		9e. Static graphic conceptual diagrams		
	1D, 2D or 3D graph space containing geometrical forms. Pure charts (dot charts, bar charts, pie charts, etc.). Non-iconic use difficult due to the absence of linguistic annotation.	9d. Static graphic graphs	9d1. Line graphs	
				9d2. Bar graphs
				9d3. Pie graphs
	12. Dynamic analogue graphics	Pure movies, videos, realistic animations, cartoons	12a. Dynamic graphic images	cf 9a1 and 9a2
			12b. Dynamic graphic maps	
Pure graphic (see 11) moving in graph space. Sequential, list and table orderings possible. 1D, 2D or 3D.		12d. Dynamic graphic graphs		
Pure animated diagrams, sequential, list and table orderings. Pure standard animations.		12c. Dynamic graphic compositional diagrams		
		12e. Dynamic graphic conceptual diagrams		
III. Arbitrary  <-li-an-ar>	13. Arbitrary static graphics	Diagrams consisting of geometrical elements. Sequential, list and tabular orderings. Non-iconic use difficult due to the absence of linguistic annotation.		
	16. Dynamic arbitrary graphics	Diagrams consisting of geometrical elements. Non-iconic use difficult due to the absence of linguistic		
IV. Explicit structures  <-li-an-ar>	17. Static graphic structures	Form fields, frames, table grids, line separations, trees, windows, bars. Sequential, list and tabular orderings. Non-iconic use difficult due to the absence of linguistic annotation.		
	20. Dynamic graphic structures	Dynamic frames, windows, scroll bars. Non-iconic use difficult due to the absence of linguistic annotation.		

### 3 Types d'éléments

Objets graphiques (image)

Texte (fonction de la police, taille, effet)

Éléments graphiques du système (ligne, traits, fenêtre,...)

### 2 propriétés:

Statique

propriétés fixes

propriétés fixes répétitives

Dynamique (changeantes, non-répétitives)

est fonction de la police

est fonction du contenu, ne nous intéresse pas dans cette partie

## b) Mode Sonore

2.a. Static analogue sign acoustic language	Part of everyday spoken language.	Included in 6 a-c.
b.Dynamic analogue sign acoustic language		
6.a. Static non-analogue sign acoustic language	Spoken letters, words, numerals, other spoken language related sounds, discourse, list orderings.	6a. Static/dynamic spoken discourse
b. Dynamic non-analogue sign acoustic language		6b. Static/dynamic spoken labels/keywords
		6c. Static/dynamic spoken notation
10.a. Static analogue acoustics	Sound 'pictures', Single sounds, sound sequences. List ordering possible.	10a. Static/dynamic acoustic images
b. Dynamic analogue acoustics	Apparently none, but many possibilities, synthetic or manipulated, exist. Music?	10b. Static/dynamic acoustic maps
	E.g. Geiger counters.	10c. Static/dynamic acoustic compositional diagrams
		10e. Static/dynamic acoustic conceptual diagrams
		10d. Static/dynamic acoustic graphs
14.a. Arbitrary static acoustics	Sound signal icons.	
b. Dynamic arbitrary acoustics		
18.a. Static acoustic structures	Apparently none.	
b. Dynamic acoustic structures		

## c) Mode Tactile

3.a. Static analogue sign haptic	Apparently none.	Included in 7 a-c.
b.Dynamic analogue sign haptic		
7.a. Static non-analogue sign haptic	Touch letters, numerals, words, other touch language related signs, text, list and table orderings. Example: Braille.	7a. Static/dynamic haptic text
b. Dynamic non-analogue sign haptic language		7b. Static/dynamic haptic labels/keywords
		7c. Static/dynamic haptic notation
11.a. Static analogue haptics	Single touch representations, touch sequences.	11a. Static/dynamic haptic images
b. Dynamic analogue haptics	Apparently none, but many possibilities exist.	11b. Static/dynamic haptic maps
		11c. Static/dynamic haptic compositional diagrams
	1D, 2D or 3D graph space containing geometrical forms. Pure charts (dot charts, bar charts, pie charts, etc.).	11e. Static/dynamic haptic conceptual diagrams
		11d. Static/dynamic haptic graphs
15.a. Arbitrary static haptics	Touch signals of different sorts.	
b. Dynamic arbitrary haptics		
19.a. Static haptic structures	Form fields, frames, grids, line separations, trees.	
b. Dynamic haptic structures		

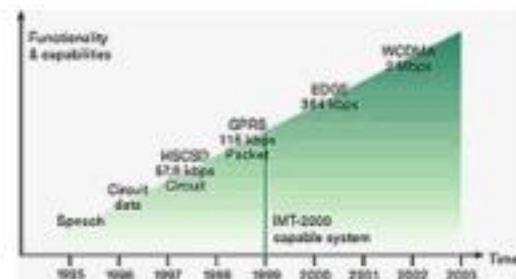
***Annexe 6***



Fonctions	Inclus	<b>Autres Fonction</b>	SMS; calculatrice; convertisseur de monnaie; réveil	SMS; Calculator; Games; NOKIA PC SUITE; calendrier	SMS; FAX; PREDICTIVE TEXT INPUT	SMS; FAX; enregistrer; unit converter; imaging (digital camera connectivity) + autres sur CD Rom	SMS; Fax; Notepad; Authentication; World clock; Games	Téle T9 (correcteur orthographique); SMS;	Address Book; Calendar; E-mail; Expense Tracker; Handwriting Recognition; Note Pad; calculator; games	Address Book; Calendar; E-mail; Expense Tracker; Handwriting Recognition; Note Pad; calculator; games	Address Book; Calendar; E-mail; Expense Tracker; Note Pad; Scheduler	Address Book; Calendar; E-mail; Expense Tracker; Note Pad; Scheduler	Address Book; Calendar; Note Pad; Scheduler	Audio Playback; Audio Recording; E-mail; Handwriting Recognition; Note Pad; Spreadsheet; Web Browser; Word Processor	Audio Recording; Calculator; Calendar; E-mail; Note Pad; Web Browser	Calculator; Calendar; Database; E-mail; Web Browser	Address Book; Audio Playback; Audio Recording; Calculator; Calendar; E-mail; Expense Tracker; Handwriting Recognition; Note Pad; Presentations; Scheduler; Spreadsheet; Web Processor	Address Book; Audio Playback; Audio Recording; Calculator; Calendar; E-mail; Expense Tracker; Handwriting Recognition; Note Pad; Presentations; Scheduler; Spreadsheet; Web Processor	Spreadsheet; Word Processor	Address Book; Calendar; Scheduler	Audio Recording	n/a	Address Book; Audio Playback; Audio Recording; Calendar; Database; E-mail; Handwriting Recognition; Note Pad; Presentations; Scheduler; Spreadsheet; Web Browser;	Address Book; Audio Playback; Audio Recording; Calculator; Database; E-mail; Note Pad; Presentations; Scheduler; Spreadsheet; Web Processor	Spreadsheet; Word Processor; E-mail; Web Browser; Database; Scheduler; Presentations	Word Processor; Address Book; Calendar; Scheduler; E-mail; Calculator; Database; Web Browser; Spreadsheet; Presentations; Handwriting Recognition	Address Book; Calendar; Database; E-mail; Handwriting Recognition; Scheduler; Spreadsheet; Web Browser; Word Processor	
		<b>Special Features</b>	lecteur MP3; dictaphone; SIM Application Toolkit	3v technology;	3v technology;		WTLS Security class 1&2;		carte de mémoire; MP3; livres electronics....	n/a	Jog Dial Navigator; gMedia; PictureGear Docket	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Mémoire		<b>Installed RAM</b>				14 application + 2 user+8 SDRAM		8 MB	2 MB	2 MB	n/a	32 MB	16 MB	32 MB	8 MB	32 MB	16 MB	512 KB	16 MB	16 MB	16 MB	32 MB	32 MB	16 MB	16 MB	32 MB	24 MB	
		<b>Flash ROM</b>						n/a	n/a	n/a	n/a	16 MB	16 MB	16 MB	8 MB	16 MB	n/a	n/a	n/a	8 MB	16 MB	n/a	16 MB	n/a	16 MB	n/a	24 MB	
		<b>Max Memory</b>				plus 16 MB card		n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
		<b>Memory Technology</b>				Memory Card slot (MultiMedia Card standard)		n/a	n/a	n/a	n/a	SDRAM	SDRAM	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	SDRAM	SDRAM	n/a	n/a	n/a	
		<b>Class</b>				32-bit ARM9-based RISC CPU		Motorola Dragonball V	Motorola Dragonball V	Motorola Dragonball E2 6832a	Motorola Dragonball E2 6832a	Motorola Dragonball E2 6832a	Motorola Dragonball E2 6832a	Motorola Dragonball E2 6832a	Intel StrongARM SA-1110	NEC MIPS-based RISC	Hitachi SH-3	Hitachi SH-3	NEC VR4122	NEC VR4122	n/a	ARM710T	ARM Processor	SH-4	SA-1110	SA-1100	VR4111	VR4121
		<b>Speed</b>						33 MHz	n/a	n/a	n/a	20 MHz	206 MHz	70 MHz	133 MHz	100 MHz	150 MHz	150 MHz	90 MHz	36 MHz	36 MHz	128 MHz	206 MHz	190 MHz	90 MHz	168 MHz		
Processeur		<b>Included Devices</b>					Docking Cradle	Docking Cradle	Docking Cradle	AC Adapter included; Docking Cradle	Connectivity Cable	Docking Cradle	n/a	n/a	n/a	n/a	Docking Cradle	n/a	n/a	n/a	n/a	Modem	Docking Cradle; Connectivity Cable; Modem; AC Adapter included;	Connectivity Cable	n/a	Connectivity Cable		



***Annexe 7***



## GPRS

The global GPRS (General Packet Radio Service) market is now beginning to take off. The introduction of GPRS is one of the key steps in the evolution of today's GSM networks to 3G (Third Generation), and GSM operators around the world are upgrading their networks, with a view to launching full commercial GPRS services in 2001.

Data traffic is increasing enormously, with growth and expectations of 40-50 per cent per year. This growth in demand for Internet access and services has paralleled the explosion in demand for mobile communications. Users want access to the Internet while they are away from their offices and homes, and GPRS can deliver this mobile Internet functionality.

The key benefit with GPRS is the ability to sustain a permanent data connection allowing a free flow of information for the mobile user. In addition, sophisticated billing systems will allow more flexible charging based not on the duration of the connection, but upon the many services that a user will access. This is facilitated by the fact that the GPRS network is designed to dynamically allocate resources to GPRS enabled terminals based on demand. Bandwidths in GPRS will be significantly higher due to the ability of GPRS mobile terminals to transmit and receive more information than standard GSM terminals. Typical bandwidths in early systems will reach 50kbps depending on network load and terminal capacity.

GPRS will allow innovative services to be created enabling new and previously inaccessible market segments to be addressed, increasing customer loyalty and reducing churn. Successful services will offer value to consumers and business users by exploiting the specific capabilities of the wireless networks and complementary terminologies. These are seen as personalized, quick response, localised to the user.

In theory, GPRS packet-based service should cost users less than circuit-switched services since communication channels are being used on a shared-use, as-packets-are-needed basis rather than dedicated only to one user at a time irrespective of usage. Once GPRS becomes available, mobile users of a virtual private network (VPN) will be able to access their private networks continuously rather than through a dial-up connection.

The next stepping stone towards 3G will be the implementation of EDGE (Enhanced Data rates for Global Evolution), offering data services and applications at speeds up to 384kbps essentially using existing infrastructure. Ericsson has been involved in the standardization of GPRS market with its complete solution, which can be easily and quickly, integrated into existing GSM networks.

GPRS will also complement Bluetooth, a standard for replacing wired connections between devices with wireless radio connections.

## UMTS

The twin streams of telecommunications and datacommunications are converging, driven by the increasing importance of Internet Protocol (IP). In parallel with the rising demand for Internet services has been the huge growth in mobile communications.

Today's 400-million mobile subscriber base is expected to grow to one billion early in the next century. The mobile phone is fast becoming the main means of personal communications for an increasing number of people.

## **A new standard for 3G**

To date, the main driver for mobile communications has been voice telephony. However, the introduction of new high-speed data capabilities, including GPRS and EDGE, and the evolution to Universal Mobile Telecommunications System (UMTS), will give new and existing GSM operators the potential for a whole range of mobile multimedia services. Electronic postcards, web surfing, access to corporate LANs and Intranets, and e-mail from a mobile terminal, to name but a few.

UMTS is the standard for delivering 3G services being developed under the auspices of ETSI. It builds on the world's most widely deployed mobile technology – GSM – and offers the prospect of a truly global wireless standard for personal multimedia communications.

### **Ericsson leading the way**

Almost 40% of the world's mobile phone calls are connected via Ericsson technology. Over 350 operators around the world are Ericsson customers. And Ericsson is already establishing a lead in the next generation of mobile systems.

The key technology for UMTS will be Wideband CDMA (WCDMA). Ericsson has signed a contract with Japanese operator NTT DoCoMo to deliver phones and base stations for the world's first commercial WCDMA network.

Ericsson is a leader in the introduction of GPRS technology and has already won more than half of all GPRS deployment contracts – in Europe, China and the USA.

Ericsson is adding IP capabilities through its own development programs and the acquisition of small and medium-sized companies with the appropriate IP skills and expertise.

Ericsson is the only vendor with complete GSM and end-to-end IP capabilities, encompassing networks, devices, terminals, applications and service creation.

UMTS is a so-called "third-generation (3G)," broadband, packet-based transmission of text, digitized voice, video, and multimedia at data rates up to 2 megabits per second (Mbps), offering a consistent set of services to mobile computer and phone worldwide. Based on the Global System for Mobile (GSM) communication standard, UMTS, endorsed by major standards bodies and manufacturers, is the planned standard for mobile users around the world by 2002. Once UMTS is fully implemented, computer and phone users can be constantly attached to the Internet as they travel and, as they roam, have the same set of capabilities no matter where they travel. Users will have access through a combination of terrestrial wireless and satellite transmissions. Until UMTS is fully implemented, users can have multi-mode devices that switch to the currently available technology (such as GSM 900 and 1800) where UMTS is not yet available.

Today's cellular systems are mainly circuit-switched, with connections always dependent on circuit availability. Packet-switched connection, using the Internet Protocol (IP), means that a virtual connection is always available to any other end point in the network. It will also make it possible to provide new services, such as alternative billing methods (pay-per-bit, pay-per-session, flat rate, asymmetric bandwidth, and others). The higher bandwidth of UMTS also promises new services, such as video conferencing. UMTS promises to realize the Virtual Home Environment (VHE) in which a roaming user can have the same services to which the user is accustomed when at home or in the office, through a combination of transparent terrestrial and satellite connections.

Trials of UMTS technology, using advanced mobile phone/computing device prototypes, are currently being conducted by a number of companies, including Ericsson and Nortel.

The spectrum for UMTS has been identified as frequency bands 1885-2025 MHz for future IMT-2000 systems, and 1980-2010 MHz and 2170-2200 MHz for the satellite portion of UMTS systems.

## **WCDMA (Wideband Code Division Multiple Access)**

is the radio access technology selected by ETSI (European Telecommunications Standards Institute) in January 1998 for wideband radio access to support third-generation multimedia services.

Optimized to allow very high-speed multimedia services such as voice, Internet access and videoconferencing, the technology will provide access speeds at up to 2Mbit/s in the local area and 384kbit/s wide area access with full mobility. These higher data rates require a wide radio frequency band, which is why WCDMA with 5MHz carrier has been selected; compared with 200kHz carrier for narrowband GSM.

Easy integration into existing infrastructure WCDMA can be added to the existing GSM core network. This will be particularly beneficial when large portions of new spectrum are made available, for example in the new-paired 2GHz bands in Europe and Asia. It will also minimize the investment required for WCDMA rollout – it will, for example, be possible for existing GSM sites and equipment to be reused to a large extent.

## **A single standard for all**

An agreement on a globally harmonized third-generation CDMA radio standard that addresses the needs of all current wireless communities was reached by the Operators' Harmonization Group in May 1999. There will be three modes in the harmonized 3G CDMA standard; a direct-sequence mode for WCDMA, a multi-carrier mode for cdma2000 (an evolution of narrowband CDMA), and a time division duplex (TDD) CDMA mode.

## **EDGE**

A major step in the evolution of GSM is the implementation of EDGE (Enhanced Data rates for Global Evolution). This will allow GSM operators to use existing GSM radio bands to offer wireless multimedia IP-based services and applications at speeds up to 384kbit/s.

EDGE will allow the advantages of GPRS to be fully explored, with fast connection set-up and higher bandwidth than traditional GSM. The combination of GPRS and EDGE will also result in much improved utilization of the radio network.

Introducing EDGE will have little technical impact, since it is fully based on GSM, and will require relatively small changes to network hardware and software. Operators do not have to make any changes to the network structure, or invest in new licenses. For example, EDGE uses the same TDMA (Time Division Multiple Access) frame structure, logic channel and 200kHz carrier bandwidth as today's GSM networks, which allows existing cell plans to remain intact. This makes the technology particularly beneficial to existing operators seeking a way to roll out wideband services rapidly and cost-efficiently across large areas of existing networks.

With EDGE, operators can offer more wireless data applications for both consumer and business users, including wireless multimedia, e-mail, web infotainment and videoconferencing.

EDGE is expected to be commercially available in 2001. It is regarded as an evolutionary standard on the way to Universal Mobile Telecommunications Service (UMTS).

# Contraintes dues aux réseaux « mobiles »

Examples of applications	Required data rate
wireless HDTV	1000 Mbit/s <i>WiFi (jusqu'à 2Gbit/s)</i> <i>Gbit (jusqu'à 1,2 Gbit/s)</i>
Mobile LAN interconnection	100 Mbit/s <i>Fast Ethernet</i> Ethernet sans fils <i>ADSL</i> <i>Cable</i> Capacité jusqu'à 11Mbit/s
High Volume data transfer	10 Mbit/s <i>Ethernet</i>
High definition video	
High quality audio	1 Mbit/s UMTS Capacité jusqu'à 2Mbit/s
Video Telephony	100 kbit/s EDGE Capacité jusqu'à 384kbit/s GPRS Capacité jusqu'à 170kbit/s
High quality voice	<i>30 kbit/s</i> 10 kbit/s <i>Modem</i> GSM
Messaging	
Telematic	
Fichier* (image/son)	3 EObit/s 1 kbit/s

■ réseaux fixes  
■ réseaux mobiles

\* En ce qui concerne la transmission de fichier image ou sonore, leur taille et par conséquent la résolution de leur contenu est fonction du temps que l'utilisateur est prêt à attendre pour obtenir le transfert  
→ Usage normal nous donnera les tailles max des fichiers suivant les terminaux

→ vitesse = KSF



Taxonomie de l'information	<	GSM	GPRS	EDGE	UMTS	>
<b>OUTPUT GRAPHIQUE</b>						
<u>Statique</u>						
propriétés fixes						
• Objets graphiques						
• Texte						
• Éléments graphiques du système						
sites réceptifs						
<u>Dynamique (charges, non-réceptives)</u>						
• Objets graphiques						
• Texte						
• Éléments graphiques						
<b>OUTPUT SONORE</b>						
Sons (analog) - Langage réel		III				
Sons synthétiques		III				
Langage synthétique		???	???	???		

■ = types d'information tolérés

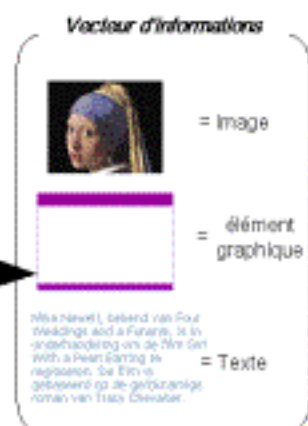
***Annexe 8***

# Décomposition d'un site Web

Site = ensemble de pages organisées entre elles



Page = ensemble d'objets définis comme des vecteurs d'informations



- Définies (type, caractéristiques, format)
- Positionnées (p/r à un point de référence)
- Arrangées (p/r aux autres objets ou informations)