

Étude Bibliographique de Master 2

Spécialité : **AIGLE**

**Personnalisation de page web :
application à l'amélioration de
l'accessibilité du web**

par **Franck PETITDEMANGE**

Mars 2014

Sous la direction de **Marianne HUCHARD,**
Michel MEYNARD, Yoann BONAVERO

Contents

1	Introduction et motivation	3
2	Exploitation des modèles	4
3	Modele de page web	7
3.1	Introduction	7
3.2	HTML 4	7
3.3	HTML 5	9
3.4	ARIA	11
3.5	Discussion	12
4	Extraction de structure (Reconstruction d’architecture?)	13
4.1	Introduction	13
4.2	Segmentation de contenu	13
4.2.1	Approche segmentation visuelle	13
4.3	Compréhension de contenu	15
4.3.1	Analyse syntaxique	15
4.3.2	Analyse fonctionnelle	18
4.4	Discution	19
5	Conclusion	20

1 Introduction et motivation

Le world wild web (www) est un reseau de ressource. La publication de ces ressources repose sur un langage universellement compréhensible et accepté par tous les ordinateurs : HTML. Historiquement conçu pour faciliter l'échange d'article dans la communauté scientifique. La démocratisation du web a fait radicalement évoluer le contenu d'une page web, sans pour autant que le langage ne suive ces évolutions. Ainsi les auteurs de page web ont détourné les pratiques de conception d'une page de manière anarchique. Ce manque d'homogénéité complique la compréhension du contenu publié sur le www par une machine. Ceci faisant perdre la propriété universelle du web voulu par son créateur Tim Berners Lee :

“La puissance du Web réside dans son universalité. L'accès à tous, quel que soit son handicap est un aspect essentiel”

Ceci introduit la motivation de ce stage et les problématiques qui en découlent.

Le sujet du stage est : la personnalisation des pages web. L'objectif est de fournir des méthodes et des outils afin d'adapter une page suivant les souhaits d'un lecteur. On s'intéresse à une application pour l'amélioration de l'accessibilité des pages web pour les personnes en situation de handicap visuel.

L'adaptation d'une page web implique notre problématique : la restructuration d'une page web. On souhaite expérimenter une approche basée sur les méta-modèles. L'idée étant d'extraire la structure d'une page et d'en construire une représentation plus abstraite. Cela doit nous permettre de s'affranchir de la diversité de conception de ces dernières. A partir de cette représentation on veut lui appliquer des transformations, puis générer une nouvelle page conforme aux transformations.

Dans la section 2 nous décrivons l'exploitation que nous souhaitons faire des modèles. En section 3 nous étudions différents modèles de page en perspective de la création d'un meta-modèle. Puis en section 4 nous explorons différentes approches à l' extraction de la structure d'une page web.

2 Exploitation des modèles

Definition. “Un modele est une description d’une partie d’un systeme écrit dans un langage formel”[7]

C’est une représentation simplifiée d’une partie d’un système, une abstraction du système étudié suivant un point de vue. Par exemple une carte routière est une abstraction d’un réseau routier, il existe plusieurs types de carte suivant ce que l’on veut étudier (chemin pédestre, chemin routier etc).

“Pour un observateur A, M est un modèle de l’objet O, si M aide A à répondre aux questions qu’il pose sur O” (Minsky)

La notion de modèle fait référence à un langage formel définissant les éléments conceptuels du modèle : le méta-modèle. La syntaxe et la sémantique du modèle sont conformes à un méta modèle.

Definition. “Un méta-modèle est un modèle qui définit un langage formel pour exprimer un modèle”[7]

Littéralement, c’est un modèle de modèle.

Dans le cadre du sujet nous nous plaçons dans un contexte de restructuration d’une page. On souhaite construire un modèle d’une page web à partir de cette dernière. L’avantage est d’en manipuler une représentation independante de la diversité de conception des pages. En effet les éléments, ou concepts, récurrents peuvent être structurés ou mises en forme de manière différente d’une page web à l’autre. [11] présente une méthodologie similaire. Les auteurs souhaitent appliquer des traitements homologues sur des systemes orientés objets implémentés dans différentes plateformes (C++, Java, ADA, etc). Pour cela ils ont réalisé un meta-modèle de systeme objet dans le but d’effectuer des traitements independamment d’une representation sous-jacente.

Nous voulons exploiter ce modèle de deux façons différentes. D’une part ce modèle doit nous servir de support pour appliquer un processus de transformation sur une page. Et d’autre part le modèle doit nous servir de support à l’extraction des règles de transformation, que nous ne traiterons pas dans la bibliographie mais par la suite dans le stage.

Definition. “Une transformation est la génération automatique d’un modèle cible depuis un modèle source selon une transformation définie”[7]

Une transformation est donc un ensemble de règles qui décrivent comment un modèle du langage source peut être transformé dans un modèle du langage cible. Plus précisément, elle est la description de comment une ou plusieurs constructions d’un langage source peuvent être transformées dans un langage cible. Plusieurs approches pour la transformation de modèle existent [6].

Suivant le langage du modèle source et du modèle cible, intervient une distinction (fig.1). On parle de transformation endogène (rééchelonnement) lorsque

les modèles sont exprimés dans le même langage. On parle de transformation exogène (translation) dans le cas contraire.

Il s'agit de transformation horizontale quand le modèle source et cible possèdent le même niveau d'abstraction. Lorsque ce n'est pas le cas, il s'agit de transformation verticale (fig.2).

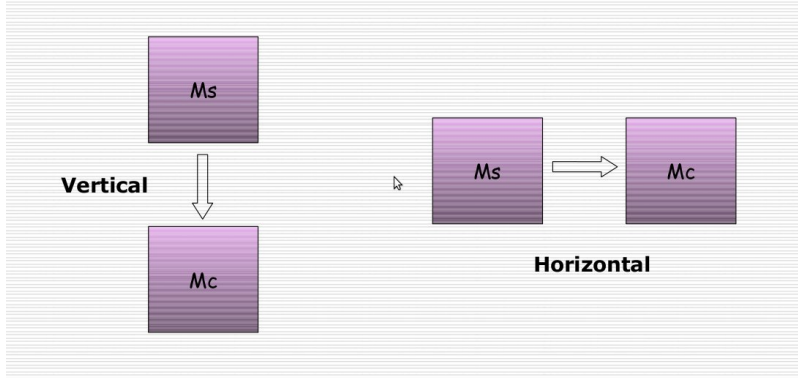


Figure 1: Transformation horizontale et verticale

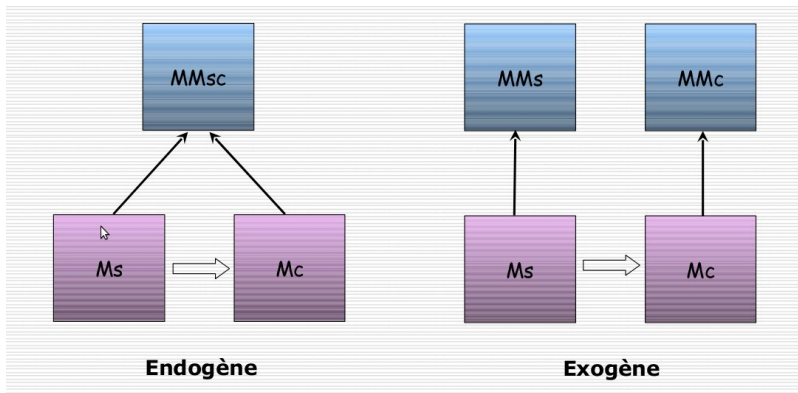


Figure 2: Transformation endogène et exogène

Il semble que notre processus de restructuration s'incrive dans le cas d'une transformation endogène et horizontale (voir figure 3). En effet, le modèle cible doit être conforme au méta-modèle et les deux modèles (cibles et sources) ont le même niveau d'abstraction.

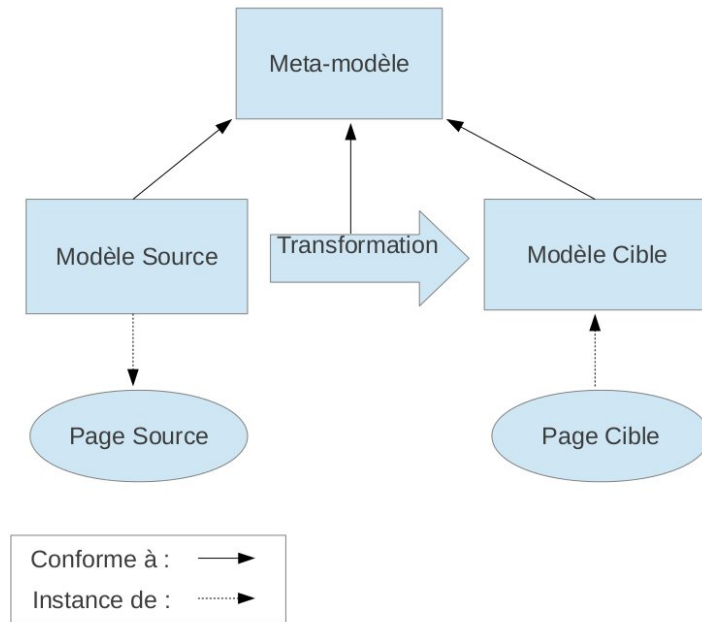


Figure 3: Transformation endogène et horizontale

3 Modele de page web

3.1 Introduction

Comme expliqué dans la section 2, le meta-modèle est le support à l'expression de transformation. Ainsi les concepts décrits par le méta-modèle doivent comprendre les éléments, ou concepts que l'on souhaite transformer dans une page. Par exemple, si l'on souhaite modifier la couleur du texte de la bar de navigation, on doit pouvoir exprimer dans la transformation le concept de bar de navigation et le concept de couleur.

On s'intéresse à la conception d'un meta-modèle proche de la vision qu'un lecteur peut avoir d'une page. Le but étant d'exprimer des transformations conformes aux souhaits du lecteur. On propose d'étudier et de comparer le langage standard de publication de page sur le web, HTML, dans la norme 4 et 5 mais aussi une toxiologie pour la description d'interface graphique ARIA.

3.2 HTML 4

HTML 4 [3] est un langage permettant la publication de contenu sur le web. C'est le langage standard actuel des pages web. Il permet de structurer le contenu et de lui associer une mise en forme. Le contenu est organisé de manière hiérarchique en le découpant en section et sous-section.

Structuration générique HTML 4 propose un mécanisme générique pour la composition de contenu formant la structure des pages web. Ce mécanisme gravite autour des éléments de type `<DIV>` leurs identifiants respectives : `id` et `class`.

DIV Signifiant division, est utilisé comme conteneur générique permettant d'organiser le contenu. Il est exploité pour :

- regrouper les éléments pour leur appliquer un style (une mise en forme particulière).
- signaler une section ou une sous-section.

id et class Chaque élément peut se voir attribuer un identifiant ou une classe d'appartenance. *id* assigne un nom à un élément. Ce nom est unique dans le document. *class* au contraire, assigne un ou plusieurs noms de classe à un élément. Un nom de classe peut être partagé par plusieurs instances d'éléments. Les identifiants et les classes sont des suites de caractères quelconque décidées arbitrairement par l'auteur du document.

Les éléments DIV utilisés conjointement avec les attributs `id` et `class` sont au cœur du mécanisme générique de structuration d'un document. DIV permet

de diviser le contenu d'un document en section et sous-section (e.g figure 5) pour décrire sa structure. Les éléments DIV ayant une sémantique neutre, c'est l'auteur du contenu qui attribut (de manière arbitraire) un nom de *class* ou un *id* (e.g figure 4). Le nom de *class* ou *id* définissant un rôle au contenu ou une mise en forme. On note le caractère implicite de la structuration.

```
<body>
  <div id="header" ></div>
  <div id="navigation_bar"><div />
  <div id="main_content">
    <div class="article"></div>
    <div class="article"></div>
  </div>
  <div id="footer"></div>
</body>
```

Figure 4: Exemple d'attribution de rôle

```
<body>
<div class="section" id="elephants-foret" >
  <h1>Les éléphants des forêts</h1>
  <p>Dans cette partie, nous abordons le sujet
moins connu des éléphants des forêts.</p>
  <div class="sous-section" id="habitat-foret" >
    <h2>L'habitat</h2>
    <p>Les éléphants des forêts ne vivent pas
dans les arbres mais au milieu d'eux.</p>
  </div>
</div>
</body>
```

Figure 5: Exemple découpage en section et sous-section

Figure 6: Architecture page web HTML 4



3.3 HTML 5

HTML 5 [5] étend HTML 4. La norme HTML 4 est toujours rétroactive. HTML 5 remplace la structure générique de HTML 4 par un nouveau modèle. Ce modèle amène de nouveaux éléments qui apportent une sémantique standard et explicite à la structure d'une page.

Structuration Les nouveaux éléments de HTML 5 spécifient donc une sémantique standard :

- **SECTION** : représente une section générique dans un document, c'est à dire un regroupement de contenu par thématique.
- **ARTICLE** : représente un contenu autonome dans une page, facilite l'inclusion de plusieurs sous documents.
- **NAV** : représente une section de liens vers d'autres pages ou des fragments de cette page.

- **ASIDE** : représente une section de la page dont le contenu est indirectement lié à ce qui l'entoure et qui pourrait être séparé de cet environnement.
- **HEADER** : représente un groupe d'introduction ou une aide à la navigation. Il peut contenir des éléments de titre, mais aussi d'autres éléments tels qu'un logo, un formulaire de recherche, etc.
- **FOOTER** : représente le pied de page, ou de la section, ou de la racine de sectionnement la plus proche.

La figure 7 montre un découpage explicite de la structure avec HTML 5 à contrario du découpage implicite de HTML 4 montré dans la figure 4

```
<body>
  <header></header>
  <nav></nav>
  <section>
    <article></article>
    <article></article>
  </section>
  <footer></footer>
</body>
```

Figure 7: Exemple d'attribution de rôle

Figure 8: Architecture HTML 5



3.4 ARIA

ARIA (Accessible Rich Internet Application) [4] est la spécification d'une ontologie décrivant une interface graphique. Elle fournit des informations sur la structuration d'un document et plus généralement décrit les éléments qui composent une interface au moyen d'un ensemble de rôles, d'états et de propriétés.

Rôle Les rôles permettent d'identifier la fonction de chaque élément d'une interface. Ils sont regroupés en trois catégories :

- **Widget Roles** : référence un ensemble de widget préfinis (alertdialog, button, slider, scrollbar, menu, etc)
- **Document Structure Roles** : décrit les structures qui organisent un document (article, definition, entête, ect)
- **Landmark Roles** : décrit les régions principales d'une interface graphique (main, navigation, search, etc)

Etats et propriétés ARIA prend en compte l'aspect dynamique et interactif des éléments d'une interface. Elle permet d'associer des états et des propriétés aux éléments graphiques. Un état est une configuration unique d'un objet. Par exemple, on peut définir l'état d'un bouton par l'état *aria-checked* qui peut prendre trois propriétés suivant l'interaction avec l'utilisateur : *true* - *false* - *mixed* . Aria prévoit même un système d'annotation pour les objets ayant des comportements asynchrones. Par exemple, on peut annoter qu'un élément se met à jour de manière autonome.

On peut associer un ensemble de propriétés par exemple la valeur minimale ou maximale que l'on doit remplir dans un champs de saisi *aria-valuemin*, *aria-valuemax*.

3.5 Discussion

L'étude ci-dessus présente de manière sommaire les éléments de langage de HTML 4, 5 et ARIA. Au regard de notre problématique, nous pouvons voir quelques perspectives à la conception d'un méta-modèle.

HTML 4 fournit une sémantique riche pour décrire les éléments textuels d'une page et ses hyperliens. En effet il y a pléthore d'éléments pour structurer un paragraphe. Cependant le mécanisme de structuration fournit est trop générique et ambiguë.

HTML 5 semble combler ces limites en apportant de nouveaux éléments lexicaux amenant une sémantique plus riche sur la structure d'une page et ses éléments. La norme 5 introduit également des balises pour la description d'élément propre à une page web comme des menus de navigation, bannière, logo, etc...

La norme ARIA intègre plus d'éléments pour décrire une page web et notamment une description du comportement des éléments d'une page web particulièrement utiles avec le développement des éléments graphiques appelés widget.

Un méta-modèle basé sur la syntaxe de HTML 5 semble être un compromis intéressant entre HTML 4 et ARIA. *En effet HTML 4 est peu adapté à l'expression de transformation, alors que ARIA est trop riche au regard de nos besoins.*

4 Extraction de structure (Reconstruction d'architecture?)

4.1 Introduction

L'extraction est la première phase du processus de restructuration. On peut la décrire comme le processus de découverture des éléments d'un tout. Les éléments à extraire de la page correspondent aux différentes briques conformes aux éléments d'un méta-modèle de page défini à priori. La problématique est due au langage de conception des pages web qui manque de sémantique, rendant un processus d'extraction automatique difficile.

La conception des pages web s'articule autour : d'un langage pour décrire la structure du document (HTML) et d'un langage pour décrire la mise en forme du document (CSS). Les pages sont constituées d'éléments hétérogènes : une page est constituée d'un ou plusieurs contenu principal, d'un menu de navigation, de publicité, etc... Chacun de ces éléments représentent une sous-structure de la page. Lorsqu'on regarde une page web depuis un navigateur, on constate que ces éléments sont structurés de façon sémantique, ils sont organisés selon leur sens. La difficulté dans la tâche d'extraction de la structure d'une page est due au manque d'expressivité du HTML. En effet, la norme actuelle de HTML (HTML 4), ne fournit pas de moyen de délimiter les éléments du document en fonction de leur sémantique. Par exemple, on ne peut pas délimiter de manière explicite la structure d'un menu dans une page avec ce langage. Le constat est que l'information de la structure d'une page apparaît principalement dans la mise en page. La structure d'une page est explicitée à travers l'utilisation de police, de couleur ou plus généralement d'éléments visuels pour caractériser les contenus qui ont la même signification.

Les approches étudiées dans la tâche d'extraction de la structure commence par une phase de segmentation. Cette étape permet de découper une page en région. On peut ainsi délimiter son contenu de façon explicite. Puis une phase de compréhension de contenu afin d'identifier le rôle de chaque région dans une page.

4.2 Segmentation de contenu

4.2.1 Approche segmentation visuelle

L'approche proposée par les auteurs [1] présente un algorithme de partitionnement basé sur les éléments de mise en forme des pages web. Le partitionnement extrait une structure qui regroupe les éléments d'une page sémantiquement proche en bloc (e.g figure 9). Le postulat est que les éléments d'une page possédant des caractéristiques de mise en forme proche, tels que la police, la couleur, la taille, sont sémantiquement proches.



Pour chaque nouveau bloc de la structure hiérarchique construite, l'algorithme teste le degré de cohérence attribué par rapport à un seuil de cohérence défini.

²Application Programming Interface

Ce seuil est défini suivant la granularité de la structure que l'on veut en sortie de l'algorithme. Si le degré de cohérence n'est pas supérieur au seuil de cohérence, le bloc est de nouveau partitionné. La structure finale est construite après que tous les blocs soient traités.

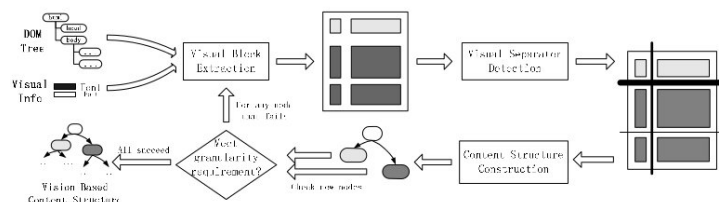


Figure 10: Algorithme de segmentation

4.3 Compréhension de contenu

4.3.1 Analyse syntaxique

Similarité de séquence L'un des postulats de base en bioinformatique est qu'une séquence génomique similaire donne une protéine présentant la même fonction. En d'autre terme, des séquences ayant des similitudes (syntaxiques) est un signe de proximité fonctionnelle. Appliqué à une page web, cela signifie que par exemple la structure syntaxique d'un menu d'une page web A est similaire à la structure syntaxique d'une page web B. Il semble cohérent d'appliquer ce postulat à une page web. En effet, la publication de contenu sur le web se standardise par l'intermédiaire des scripts. Ceux-ci génèrent des pages de manière automatique et standardisée (Wordpress, joomla, etc). Afin d'obtenir un bon référencement, les moteurs de recherche insistent les auteurs de page web d'adopter des normes de conception. En construisant une séquence représentative d'un élément de notre méta-modèle, il devrait être possible de déterminer la fonction d'un fragment de page web. Pour déterminer la similarité de deux séquences, les bioinformaticiens utilisent le concept du coût d'alignement.

Definition. L' alignement est la mise en correspondance de deux séquences. Soit deux séquences $X_{1:n}$ et $Y_{1:m}$ dont la valeur est définie dans le même alphabet fini Λ . Un alignement est une correspondance entre les lettres de la première séquence et celles de la deuxième, sans en changer l'ordre, et en autorisant éventuellement des « trous ».

```

G A A T C _ T G A C
C A _ _ C G T _ A _

```

Figure 11: Alignement possible des sequences $X_1=GAATCTGAC, Y_1=CACGTA$

La mise en correspondance repose sur trois types d'opérations élémentaires : la substitution, l'insertion, deletion. Plusieurs combinaisons d'alignements existent. Lorsque l'on souhaite comparer la similitude de deux séquences, la meilleure solution est celle qui minimise le nombre d'opérations d'insertions et de deletions.

- soient 2 séquences a priori homologues

- **CTGGGCCAGATC**

- **AACAGGGCCAAATC**

- voilà un alignement possible

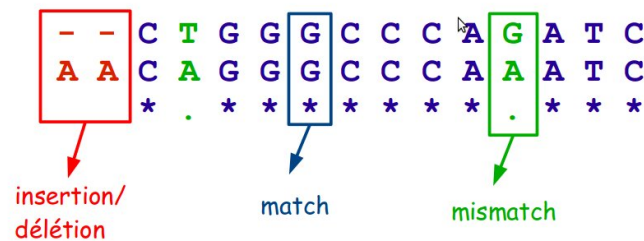


Figure 12: opérations d'alignements

Il existe deux types d'alignements : local et global.

L'alignement global L'alignement global est conçu pour comparer des séquences sur toute leur longueur. Une méthode optimale pour trouver un alignement global maximal de chaîne de caractères est l'algorithme de Needleman-Wunsch [8].

L'alignement local L'alignement local est conçu pour rechercher dans la séquence A des régions semblables à la séquence B (ou à des parties de la séquence B). Une méthode pour trouver un alignement local maximal de chaîne de caractères est l'algorithme de Smith & Waterman [9].

Global FTFTALILLAVAV
 F--TAL-LLA-AV

Local FTFTALILL-AVAV
 --FTAL-LLAAV--

Figure 13: Comparaison sequence d'alignement globale et locale

Comme on le voit dans la figure 13 l'alignement global tente d'aligner les séquences sur toute leur longueur, tandis que l'alignement local se focalise sur les zones de forte homologie. L'alignement global est adapté à la comparaison de deux fragments de taille approximativement égales. Alors que l'alignement local, permet de faire correspondre un fragment dans une plus grande séquence.

Similarité d'arbre Une autre approche dans l'étude de similarité de structure est la comparaison d'arbre. Cette approche consiste à trouver la plus petite ou la moins coûteuse séquence d'opération d'édition (substitution, suppression et insertion) qui permet la transformation d'un arbre vers un autre.

Notons Λ un noeud vide. Une opération d'édition est écrite $b \rightarrow c$, où b et c sont soit un noeud, soit Λ .

- $b \rightarrow c$ est une opération de substitution si $b \neq \Lambda$ et $c \neq \Lambda$,
- une opération de suppression si $b \neq \Lambda \doteq c$,
- une opération d'insertion si $b = \Lambda \neq c$

Pour exprimer une séquence d'opération élémentaire qui transforme l'arbre, on utilise le concept de mapping, introduit [10]. Un mapping établie une correspondance un-à-un entre les nœuds de deux arbres ordonnés et qui préservent l'ordre des nœuds.

Definition. Un Mapping M de l'arbre T1 vers l'arbre T2 est un ensemble de paire ordonnée d'entier (i, j) , $1 \leq i \leq n1$, $1 \leq j \leq n2$, satisfaisant les conditions suivantes, pour tous $(i1, j1), (i2, j2) \in M$:

- $i1=i2$ si et seulement si, $j1=j2$ (one-to-one condition);
- $t1[i1]$ est à droite de $t1[i2]$, si et seulement si, $t2[j1]$ est à droite de $t2[j2]$ (préservation de l'ordre des noeuds frères);
- $t1[i1]$ est un ancêtre de $t1[i2]$ si et seulement si, $t2[j1]$ est un ancêtre de $t2[j2]$ (préservation de l'ordre des ancêtres);

Definition. Soit M un mapping entre les arbres $T1$ et $T2$ décrivant des opérations de modification. S est l'ensemble de paire $(i, j) \in M$, D l'ensemble des nœuds $T1[i]$ n'ayant pas de paire $(i, j) \in M$, et I l'ensemble des nœuds $T2[j]$ n'ayant pas de paire $(i, j) \in M$. Le coût du mapping est donné par $|S|p + |I|q + |D|r$, où p est le coût des substitutions non identiques, q est le coût des insertions (1), r est le coût d'une suppression (1), le coût des substitutions identiques est 0.

Pour connaître la similarité entre deux structures, on veut calculer une distance d'alignement. C'est à dire trouver le coût minimum du mapping pour que $T1$ et $T2$ soient isomorphes. KUO-CHUNG TAI [10] propose un algorithme de programmation dynamique pour résoudre la question de distance d'arbre en temps séquentiel $O(|T1| \times |T2| \times \min(\text{depth}(T1), \text{leaves}(T1)) \times \min(\text{depth}(T2), \text{leaves}(T2)))$.

4.3.2 Analyse fonctionnelle

Pour connaître le rôle de chaque élément, [2] propose une approche basée sur une analyse fonctionnelle de ces derniers. Pour cela les auteurs proposent un modèle : Function-based object. Ce modèle considère chaque élément d'une page comme un objet ayant des propriétés, tel que la valeur de ces propriétés définit le rôle de l'objet. Les principales propriétés sont basées sur la présentation (type de média, agencement, etc), les hyperliens (le nombre d'hyperlien? vers où pointent ils?), l'interaction avec l'utilisateur (uniquement affichage, soumission de contenu?). L'extraction des propriétés fonctionnelles sont faites principalement par analyse de la nature des balises et les attributs associés aux balises. Suivant la valeur des propriétés, les objets sont rangés dans différentes catégories : objet de type informatif, de navigation, d'interaction, de décoration, etc. L'analyse de la catégorie d'un objet se fait au moyen d'un arbre de décision (e.g figure 14). Les principales catégories (rôle) des objets.

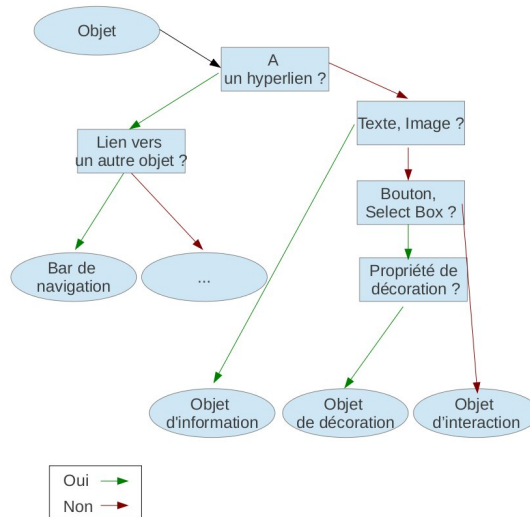


Figure 14: Exemple d'arbre de décision basique

4.4 Discution

(alignement global) La méthode de segmentation visuelle (section précédente) semble pouvoir extraire et regrouper les éléments sémantiquement proche d'une page. L'inconvénient est que l'on ne connaît pas la signification de ses éléments. Une approche serait de construire une séquence representative d'un élément de notre méta-modèle et de comparer les séquences extraites d'une page aux séquences associée au méta-modèle.

(alignement local) Cette approche ne necessite pas d'extraire de sequences dans le une page. On peut comparer directement une sequence du meta-modele à la page pour trouver la zone qui correspond.

5 Conclusion

References

- [1] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting content structure for web pages based on visual representation. In *Web Technologies and Applications*, pages 406–417. Springer, 2003.
- [2] Jinlin Chen, Baoyao Zhou, Jin Shi, Hongjiang Zhang, and Qiu Fengwu. Function-based object model towards website adaptation. In *Proceedings of the 10th international conference on World Wide Web*, pages 587–596. ACM, 2001.
- [3] World Wide Web Consortium et al. Html 4.01 specification. <http://www.w3.org/TR/REC-html40/>, 1999.
- [4] World Wide Web Consortium et al. Accessible rich internet applications 1.0. <http://www.w3.org/WAI/intro/aria>, 2014.
- [5] World Wide Web Consortium et al. Html 5 specification. <http://www.w3.org/TR/html5/>, 2014.
- [6] Krzysztof Czarnecki and Simon Helsen. Classification of model transformation approaches. In *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, volume 45, pages 1–17, 2003.
- [7] Anneke G Kleppe, Jos B Warmer, and Wim Bast. Mda explained: the model driven architecture: practice and promise. 2003.
- [8] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.
- [9] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.
- [10] Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):422–433, 1979.
- [11] Sander Tichelaar, Stéphane Ducasse, and Serge Demeyer. Famix and xmi. In *Reverse Engineering, 2000. Proceedings. Seventh Working Conference on*, pages 296–298. IEEE, 2000.