

RAPPORT

Projet Programmation fonctionnelle

Réalisé par :

- KOLUBAKO Tennessy
- NICOLAS Jean-Eric
- PADONOU LOUEMBET Jimmy

Contenue

A. Architecture de l'application & Caractéristiques.....	2
I. Architecture de l'application.....	2
II. Caractéristiques de l' application.....	2
B. Difficultés rencontrées.....	3

Dans le cadre de ce projet, nous devons réaliser un programme permettant de tester la validité d'un fichier XML, par rapport à une DTD contenant l'ordre d'apparition des balises, ainsi que leur occurrence. Ainsi dans un premier temps, nous vérifions si la syntaxe du fichier XML est correcte (Comme, par exemple, si chaque balise ouverte à été fermée), puis nous vérifions que celui-ci soit en adéquation avec les règles fixées dans la DTD.

A. Architecture de l'application & Caractéristiques

I. Architecture de l'application

Notre programme peut se diviser en trois parties principales. La première étant la lecture de fichier et la création des différents types inductifs servant à la suite.

La deuxième partie s'occupe, de la vérification syntaxe du fichier XML (ex : voir si une balise ouverte a bien été formé ou l'ordre de fermeture des balises est le bon, etc ..) puis de construire l' arbre correspondant au fichier XML : Pour cela, nous avons décidé d'utiliser une liste de couple string*entry, avec entry prenant pour valeur, soit DATA, correspondant à une donné (chaîne de caractères), soit une liste string*entry avec string qui correspond au nom de la balise. Et la construction de l'arborescence du fichier DTD : Pour ce faire, nous avons décidé d'utiliser une liste de couple ide*model, avec ide correspondant au nom de la balise, et model pouvant prendre comme valeur, soit PCATA, qui correspond à une donnée(chaîne de caractères), soit une liste de couple occurrence*atom, avec occurrence le nombre d'occurrence possible d'une balise(ex : ATOM1, ATOM_MULT, etc), et atom le nom de la balise contenue, ou un couple occurrence*atom.

La troisième partie s'occupe de la validation XML DTD consistant à vérifier si le contenu du fichier DTD résume bien le fichier XML en tenant compte de l'ordre, des occurrences et des noms de balises (la méthode générale, c'est à dire celle qui appelle toutes les autres pour la validation, est « validation a b » avec a et b des arguments qui sont respectivement les fichiers XML et DTD.

II. Caractéristiques de l' application

Pour une clarté du code nous avons choisi d' écrire tous les types inductifs en minuscules et les constructeurs tout en majuscules.

B. Difficultés rencontrées

Le développement de ce programme ne s'est pas fait sans difficultés. La première rencontrée a été la construction de l'arborescence de la DTD. En effet, afin de traiter la DTD, il fallait dans un premier temps, pouvoir séparer chaque information importante. Et il réside, dans ce procédé, un problème que nous n'avons pas pu résoudre. Celui-ci étant qu'il ne doit pas y avoir d'espaces entre les éléments contenus dans les parenthèses. Voici un exemple :

- Cette ligne est correcte : <!ELEMENT contact (nom,prenom,telephone?)>
- Cette ligne ne l'est pas : <!ELEMENT contact (nom, prenom, telephone?)>

La seconde a été la méthode de validation XML DTD qui a coûté plus de temps que les autres étapes. Des difficultés à gérer les types d'occurrence, savoir ce que signifie réellement chacun des types d'occurrence (grâce à plusieurs recherches sur internet nous avons pu trouver la solution) et l'existence des mêmes balises dans la DTD et le XML. La solution trouvée pour régler le problème d'existence des balises a été de stocker dans une liste les différents noms des balises sans doublons pour chaque fichier et les comparer.