

# Parseur / Validateur XML-Lite

Mathis Deloge, Antoine Petot, Ange Picard

Lundi 10 octobre 2016

# Sommaire

- 1 Présentation du sujet
  - Le sujet
  - Prolongements possibles
  - Le XML-Lite
- 2 Modèle mathématique
- 3 Présentation du programme
- 4 Résultats
- 5 Conclusion

# Le sujet

# Le sujet

## Le XML-Lite

Le XML est un langage balisé. Nous utiliserons une version simplifiée : le XML-Lite.

# Le sujet

## Le XML-Lite

Le XML est un langage balisé. Nous utiliserons une version simplifiée : le XML-Lite.

## Descriptif

Un parseur / validateur XML-Lite est un programme capable de lire un fichier, d'indiquer s'il vérifie la norme XML-Lite et si oui, de l'analyser et de retenir sa structure ainsi que son contenu.

# Prolongements possibles

# Prolongements possibles

## Les différents prolongements du sujet

- Permettre le débogage du fichier XML.
- Permettre au validateur de s'accorder à un schéma prédéfini.
- Modifier le validateur afin qu'il prenne en compte un schéma.
- Ajouter un interpréteur suivant le schéma d'un générateur de QCM.

# Le XML-Lite



## Les règles

- Une balise possède un nom.
- Une balise doit être ouverte puis fermée.
- Une balise peut contenir du texte.
- Une balise peut contenir d'autre balises.
- L'ordre des balises filles n'a pas d'importance et tout le texte contenu dans une balise est regroupé en un seul bloc.
- Une balise fille doit être fermée avant la fermeture de la balise parent.
- Une balise peut contenir une balise du même nom.
- Un document doit commencer par l'ouverture d'une balise se fermant à la fin du document.

# Exemples de XML-Lite

# Exemples de XML-Lite

## XML-Lite correct

```
1 <FirstTag>
2     <ChildTag>
3         <AnotherChildTag>
4         </AnotherChildTag>
5     </ChildTag>
6     <tag>
7     </tag>
8 </FirstTag>
```

# Exemples de XML-Lite

## XML-Lite correct

```
1 <FirstTag>
2     <ChildTag>
3         <AnotherChildTag>
4         </AnotherChildTag>
5     </ChildTag>
6     <tag>
7     </tag>
8 </FirstTag>
```

## Raisons

- Une balise s'ouvre en début de document et se ferme en fin.
- Respect de toutes les règles d'ouverture et fermeture de balise.
- Toutes les balises sont correctement imbriquées et nommées.

# Exemples de XML-Lite

# Exemples de XML-Lite

## XML-Lite invalide

```
1  <FirstTag>
2      <SecondTag>
3          <EndTag>
4              <AloneTag>
5                  </>
6      </FirstTag>
7  </SecondTag>
8  Un peu de texte
```

# Exemples de XML-Lite

## XML-Lite invalide

```
1    <FirstTag>
2        <SecondTag>
3            <EndTag>
4                <AloneTag>
5                    </>
6        </FirstTag>
7    </SecondTag>
8    Un peu de texte
```

## Raisons

- Les balises ne sont pas correctement imbriquées.
- Une balise n'est pas correctement nommée.
- Une balise s'ouvre et ne se referme pas.

# Sommaire

- 1 Présentation du sujet
- 2 **Modèle mathématique**
  - Choix du modèle mathématique
  - Représentation de l'automate fini
  - Un arbre...
- 3 Présentation du programme
- 4 Résultats
- 5 Conclusion



# Automate fini

# Automate fini

Contrainte du validateur

Correspondance au modèle syntaxique du fichier XML-Lite.

# Automate fini

## Contrainte du validateur

Correspondance au modèle syntaxique du fichier XML-Lite.

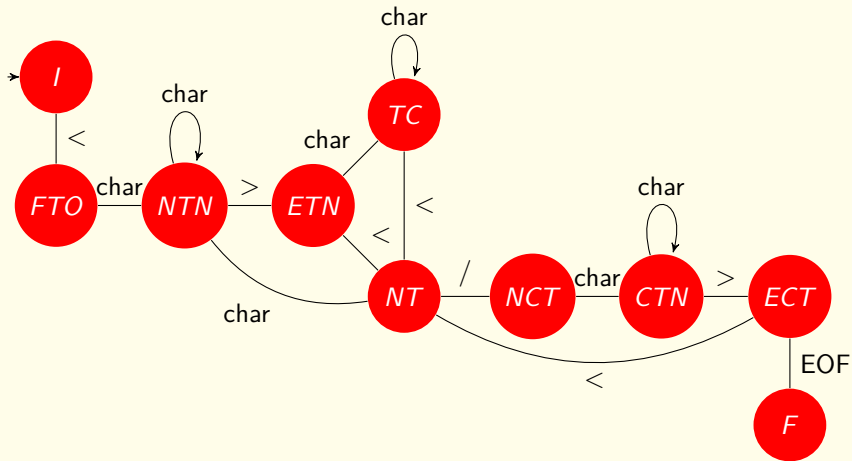
## Solution

L'automate fini nous permet :

- Une analyse du document caractère par caractère.
- Une implémentation simple et rapide.
- Un débogage simplifié.

# Représentation de l'automate fini

# Représentation de l'automate fini



# Légende de l'automate

# Légende de l'automate

## Légende

I	Initial
FTO	First Tag Opening
NTN	New Tag Name
ETN	End Tag Name
TC	Text Content
NT	New Tag
NCT	New Closing Tag
CTN	Closing Tag Name
ECT	End Closing Tag
F	Final

Un arbre...



# Un arbre...

## Contraintes

Représenter la structure d'un document XML-Lite.

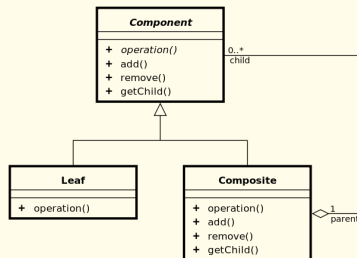
# Un arbre...

## Contraintes

Représenter la structure d'un document XML-Lite.

## Solution

Un document XML ayant déjà une structure d'arbre, nous avons adopté une structure d'arbre pour la représentation.



# Sommaire

- 1 Présentation du sujet
- 2 Modèle mathématique
- 3 Présentation du programme**
  - Le parseur / validateur
- 4 Résultats
- 5 Conclusion

# Le parseur / validateur

# Le parseur / validateur

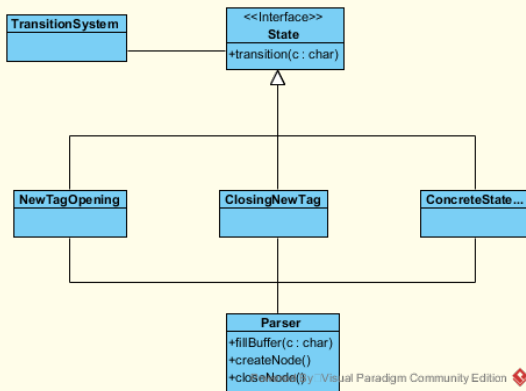
## Définition

- Analyse syntaxique d'un document
- Utilisation d'un système de transition

# Le parseur / validateur

## Définition

- Analyse syntaxique d'un document
- Utilisation d'un système de transition



# Sommaire

- 1 Présentation du sujet
- 2 Modèle mathématique
- 3 Présentation du programme
- 4 Résultats**
- 5 Conclusion

# Résultats



## Debugger

Le modèle d'automate permet une vérification à la volée dans le parseur.

# Résultats

## Débugger

Le modèle d'automate permet une vérification à la volée dans le parseur.

## Schéma

Validation de la structure en fonction de contraintes sur l'arbre.

# Résultats

## Debugger

Le modèle d'automate permet une vérification à la volée dans le parseur.

## Schéma

Validation de la structure en fonction de contraintes sur l'arbre.

## Interpréteur de schéma

Création des contraintes qui constituent le schéma en fonction d'un document DTD. Implémentation d'un deuxième automate fini.

# Résultats

## QCM

## Schéma

```
1 | <!ELEMENT QUESTION (ANSWER+, ANSWER ,  
  | ANSWER, ANSWER, ANSWER)>  
2 | <!ELEMENT ANSWER (VALID+)>  
3 | <!ELEMENT VALID ()>
```

## Interpréteur QCM

Remplir un objet QCM en utilisant l'arbre généré par le parser.

Démo

# Sommaire

- 1 Présentation du sujet
- 2 Modèle mathématique
- 3 Présentation du programme
- 4 Résultats
- 5 Conclusion**

# Conclusion

# Conclusion

## Observation

Des outils mathématiques très abordables. Implémentation des différents prolongements possibles.



# Conclusion

## Observation

Des outils mathématiques très abordables. Implémentation des différents prolongements possibles.

## Difficultés rencontrées

Parseur DTD, regex  $\rightarrow$  automate.

# Conclusion

## Observation

Des outils mathématiques très abordables. Implémentation des différents prolongements possibles.

## Difficultés rencontrées

Parseur DTD, regex  $\rightarrow$  automate.

## Outils acquis

- Utilisation d'un automate fini.
- Fonctionnement d'un langage de balisage.
- Structure d'arbre

# Questions

Des questions ?

# Conclusion

le XML c'est génial, sauf le vendredi.

## 1 Présentation du sujet

- Le sujet
- Prolongements possibles
- Le XML-Lite
  - XML-Lite correct
  - XML-Lite invalide

## 2 Modèle mathématique

- Choix du modèle mathématique
- Représentation de l'automate fini
- Un arbre...

## 3 Présentation du programme

- Le parseur / validateur

## 4 Résultats

## 5 Conclusion