



Langages de transformation incrémentaux

1^{er} avril 2003

Avancement et planning



<http://wam.inrialpes.fr>



Plan

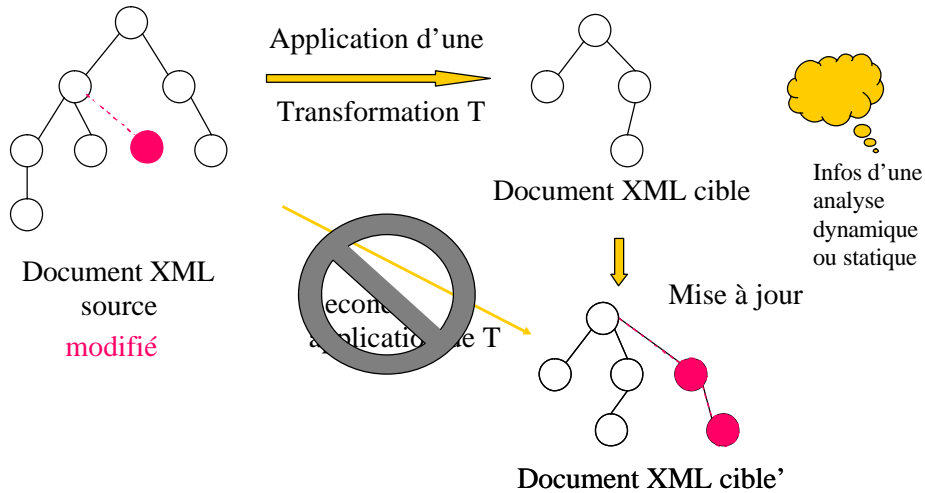
- Le problème de la transformation incrémentale
- État de l'art
 - Traitement incrémental
 - Langages de transformation
 - iXSLT
- Un nouveau langage « Omega », et une nouvelle approche pour la transformation incrémentale
- Planning pour la période à plein temps



<http://wam.inrialpes.fr>



La transformation incrémentale



Données : un arbre XML x , une transformation t , le résultat $t(x)$, la modification $?x$ de x , des infos auxiliaires éventuelles.



But : le calcul efficace de $t(x + ?x)$ <http://wam.inrialpes.fr>



État de l'art des langages de transformation

- Langages (XSLT, XQuery, Circus, CDuce)
- Synthèse
 - Modèle de règle de transformation :

Partie gauche :	Partie droite:
$< \text{Sélection dans l'arbre source}, \text{Génération} >$	
 - Contexte d'exécution des règles
 - Xpath est le langage de sélection du W3C
 - Permet de sélectionner un ensemble de nœuds d'un arbre XML
 - Exemples : `article/paragraph[2]`, `/livre[@auteur="Paul"]`
- Constats
 - Partie gauche des règles bien définie, partie droite ?
 - Aucun spécifiquement étudié pour l'incrémentalité



<http://wam.inrialpes.fr>



État de l'art du traitement incrémental en général

- Nombreux travaux, domaines variés; 2 catégories :
 - Algorithmes spécifiques
 - Méthodes générales
- Synthèse
 - 4 techniques de base :
 - Exploitation du résultat précédent
 - Exploitation de résultats intermédiaires
 - Exploitation d'informations auxiliaires
 - Évaluation partielle
 - Papier écrit résumant plusieurs techniques



<http://wam.inrialpes.fr>



Etat de l'art de la transformation incrémentale

- iXSLT
 - Processeur XSLT incrémental
 - Idée clefs :
 - Analyse statique de la feuille XSLT
 - Règles de réévaluation
 - Modification d'un processeur de transformation XSLT classique
 - Remarque :
 - Paramètres dynamiques imposent des règles de réévaluation « relâchées »




<http://wam.inrialpes.fr>



Etat de l'art - Conclusions

- Constats
 - Aucun langage de transformation spécifiquement étudié pour la transformation incrémentale (« processus batch »)
 - Partie gauche des règles bien définie, partie droite ?

< *expression XPath de sélection*,  >



<http://wam.inrialpes.fr>



Omega

- Un nouveau langage de transformation
- Pourquoi ?
- Idée clef : XPath partout

< *expr XPath de sélection*, *expr Xpath de construction* >



<http://wam.inrialpes.fr>



Principes de Omega

- Système de réécriture
- Forme d'une règle :

$$\langle i, o \rangle ? \langle o' \rangle$$

i : sélection dans l'arbre source

o : expression du contexte dans l'arbre cible

o' : réécriture de o

- i, o : expressions XPath avec sémantique classique (sélection)
- o' : expression de construction
- 2 exemples :
 - /a/b, /c ? /c/d
 - /article/author@name[.= \$name], /html/body ? /html/body/h1[.= \$name]



<http://wam.inrialpes.fr>



Premier intérêt

- Expression de transformations réversibles :

$$\langle i, o \rangle ? \langle i', o' \rangle$$

== ==



<http://wam.inrialpes.fr>



Exemple de règle réversible

/article/section,
/book
?
/article,
/book/chapter



<http://wam.inrialpes.fr>



Processus d'application d'une règle

- Règle :
 $\langle i, o \rangle ? \langle o' \rangle$
- Evaluation de i et o
- Produit cartésien $E = I \times O$
- Application d'une règle par pas : traitement du premier élément de E non traité



<http://wam.inrialpes.fr>



Stratégie Omega

- Stratégie décrit un ordre d'application des règles
- Exemples :

```
strategy is {  
  r1 (r2 | r3) r4  
}
```

```
strategy is {  
  Rule1 = r1 Rule2*  
  Rule2 = r3 | (r4* r5)  
}
```



<http://wam.inrialpes.fr>



Contextes Omega

- Un problème : manque d'expressivité du contexte dans l'arbre cible
- On aimerait pouvoir définir et réutiliser des contextes XML
- Solution : extension « = = »
- Exemple :

```
init: /article/section  
      /html/body  
-->  
      /html/body/h1[.==$subpath]  
defcontext C as {$o:=subpath}
```

- Contextes empilés dans le cas d'une stratégie récursive



<http://wam.inrialpes.fr>



Exemple (table -> liste)

```
ruleset N: HTML -> HTML
in context C : {i:NodeSet; o:NodeSet}
is {

  table:          $C:i/table[.==$ipath] ,
                  $C:o
                  -->
                  $C:o/ul[.==$opath]
  defcontext C as C + {$i:=$ipath; $o:=$opath}

  textcell:       $C:i/tr/td[.==$ipath]/text()[.=$val];
                  $C:o
                  -->
                  $C:o/li[.==$opath]/text()[.=$val],
  defcontext C as C + {$i:=$ipath; $o:=$opath}

  innertab:       $C:i/tr/td/table[.==$ipath],
                  $C:o
                  -->
                  $C:o/ul[.==$opath]
  defcontext C as C + {$i:=$ipath; $o:=$opath}
}

strategy is {
  Thetable = table Row*
  Row = textcell | (innertab Row*)
}
```



<http://wam.inrialpes.fr>



Conclusions sur Omega (1/2)

- Partie droite précisément définie
- Intérêts pour la transformation incrémentale :
 - Réversibilité : permet de défaire des règles déjà appliquées
 - Stratégie ordonnée => optimisation possible



<http://wam.inrialpes.fr>



Conclusions sur Omega (2/2)

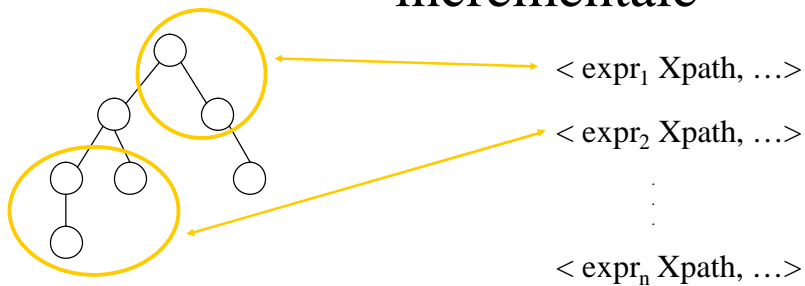
- Recherches envisagées
 - Définition de la sémantique de Omega :
 - « | » et « // » dans une expression de construction ?
 - Problème
 - Jusqu'où peut aller la réversibilité ?



<http://wam.inrialpes.fr>



Retour au cœur du problème de la transformation incrémentale



- Déterminer les expressions de sélection à réévaluer lors d'une modification



<http://wam.inrialpes.fr>



Approche par test d'inclusion

- Modification élémentaire du document source : expression Xpath de construction (attention : sur l'arbre source)
- Test d'inclusion de cette expression avec les expressions de sélection des règles Omega
- Exemple : $/\text{article}/\text{section}[2] \subset i_1? i_2? i_n?$
- Tests guidés par la stratégie



<http://wam.inrialpes.fr>



Fondements de l'approche

- Hypothèse : test d'inclusion d'expressions Xpath complet pour les types d'expressions considérées
- 2 formes d'expression de sélection dans Omega :
 - absolue : `/noeud`
 - préfixée par un contexte : `$C:i/noeud`
- Analyse statique des contextes nécessaire :
 - `/article/paragraph[2]` est-il contenu dans `/book/paragraph` ?
 - `/article/paragraph[2]` contenu dans `$C:i/paragraph?`
Analyse statique de l'évolution des contextes de la transformation...



<http://wam.inrialpes.fr>



Planning - Recherches envisagées

- Définition de la sémantique de Omega
- Analyse statique de l'évolution des contextes dans Omega
 - Basée sur :
 - Les règles de la transformation considérée
 - La stratégie spécifiée
 - Relative :
 - Aux éventuels modèles de documents source et cible
 - À l'instance de document transformé
- Propriétés d'efficacité – étude des cas valables



<http://wam.inrialpes.fr>



Conclusion

- Le problème de la transformation incrémentale
- Limite des langages de transformation
- Omega, ses intérêts pour la transformation incrémentale
- Nouvelle approche basée sur le test d'inclusion
- Planning envisagé



<http://wam.inrialpes.fr>



Merci !

Questions ?



<http://wam.inrialpes.fr>



Pistes annexes découvertes

- Transformation en streaming XML
 - Grâce à une approche orthogonale : évaluation progressive des expressions Xpath
- Contrôle de validité incrémental ?
 - Grâce aux règles Omega



<http://wam.inrialpes.fr>