Informatique pour le TAL 2 (SL02244X)
Manipuler des données XML en Perl

Franck Sajous/CLLE-ERSS

5 avril 2012

http://w3.erss.univ-tlse2.fr/membre/fsajous/

Repérage des tours de parole contenant des pauses

Les tours de parole contenant des pauses -et uniquement ceux-là- en affichant le locuteur : ???

Repérage des tours de parole contenant des pauses

Les tours de parole contenant des pauses -et uniquement ceux-là- en affichant le locuteur : ???

Par étapes !

1. exploration des données et du fonctionnement du parseur : faire afficher le nom des balises ouvrantes, fermantes, des attributs, du texte, etc.
   N'hésitez pas à abuser des instructions d'affichage pour comprendre :

```
print "Entree dans la balise $nomBalise\n";
print "Liste des attributs = ...";
# afficher ici la liste des attributs
print "Reception de portion de texte : $txt\n";
print "Fermeture de la balise $balise\n";
```

Repérage des tours de parole contenant des pauses

Les tours de parole contenant des pauses -et uniquement ceux-là- en affichant le locuteur : ???

Par étapes !

1. exploration des données et du fonctionnement du parseur : faire afficher le nom des balises ouvrantes, fermantes, des attributs, du texte, etc.
   N'hésitez pas à abuser des instructions d'affichage pour comprendre :

```
print "Entree dans la balise $nomBalise\n";
print "Liste des attributs = ...";
# afficher ici la liste des attributs
print "Reception de portion de texte : $txt\n";
print "Fermeture de la balise $balise\n";
```

2. ne faire afficher que le texte correspondant aux tours de parole (*i.e.* contenu dans les balises <u></u>)
3. préciser le locuteur
4. restreindre aux balises <u> qui contiennent des pauses
5. subtilités complémentaires

Exploration

```
sub handleStartTag {
    shift;
    my $tagName = shift;
    print "OPENING: $tagName\n";
}

sub handleEndTag {
    shift;
    my $tagName = shift;
    print "CLOSING: $tagName\n";
}

sub handleChar {
    shift;
    my $txt = shift;
    chomp $txt;
    $txt =~ s/\s+/ /g;
    print "String: $txt\n";
}
```

Exploration

```
sub handleStartTag {
    shift;
    my $tagName = shift;
    print "OPENING: $tagName\n";
}

sub handleEndTag {
    shift;
    my $tagName = shift;
    print "CLOSING: $tagName\n";
}

sub handleChar {
    shift;
    my $txt = shift;
    chomp $txt;
    $txt =~ s/\s+/ /g;
    print "String: $txt\n";
}
```

```
./saxAirFrance1.pl ../AirFrance.xml |less
```

Exploration

```
sub handleStartTag {
    shift;
    my $tagName = shift;
    print "OPENING: $tagName\n";
}

sub handleEndTag {
    shift;
    my $tagName = shift;
    print "CLOSING: $tagName\n";
}

sub handleChar {
    shift;
    my $txt = shift;
    chomp $txt;
    $txt =~ s/\s+/ /g;
    print "String: $txt\n";
}
```

```
./saxAirFrance1.pl ../AirFrance.xml |less


++OPENING: teiCorpus.2
        STRING: ******
        STRING: *** ***
++OPENING: teiHeader
...
++OPENING: head
        STRING: ***COMMUNICATION I-1***
--CLOSING: head
        STRING: ******
        STRING: *** ***
++OPENING: u
        STRING: ***Air France bonjour***
--CLOSING: u
        STRING: ******
        STRING: *** ***
++OPENING: u
        STRING: ***bonjour je voudrais
           faire uneréservation pour ***
++OPENING: vocal
--CLOSING: vocal
        STRING: *** Londres***
--CLOSING: u
```

Gestion des attributs

```
sub handleStartTag {
  shift; # discarding 1st arg
  my $tagName = shift @_;
  my %attributes;

  while ($#_ > 0) {
    my $key   = shift @_;
    my $value = shift @_;
    $attributes{$key} = $value;
  } # while ($#_ > 0)

  print "++OPENING: $tagName";

  if (keys (%attributes)) {
    print "{";
    foreach my $key (keys (%attributes)) {
      my $value = $attributes{$key};
      print " $key=$value";
    } # foreach my $key (keys (%attributes))
    print "}";
  } # if (keys (%attributes))

  print "\n";
  } # handleStartTag
```

Gestion des attributs

./saxAirFrance2.pl ../AirFrance.xml |less

```
sub handleStartTag {
  shift; # discarding 1st arg
  my $tagName = shift @_;
  my %attributes;

  while ($#_ > 0) {
    my $key   = shift @_;
    my $value = shift @_;
    $attributes{$key} = $value;
  } # while ($#_ > 0)

  print "++OPENING: $tagName";

  if (keys (%attributes)) {
    print "{";
    foreach my $key (keys (%attributes)) {
      my $value = $attributes{$key};
      print " $key=$value";
    } # foreach my $key (keys (%attributes))
    print "}";
  } # if (keys (%attributes))

  print "\n";
  } # handleStartTag
```

Gestion des attributs

```
sub handleStartTag {
  shift; # discarding 1st arg
  my $tagName = shift @_;
  my %attributes;

  while ($#_ > 0) {
    my $key   = shift @_;
    my $value = shift @_;
    $attributes{$key} = $value;
  } # while ($#_ > 0)

  print "++OPENING: $tagName";

  if (keys (%attributes)) {
    print "{";
    foreach my $key (keys (%attributes)) {
      my $value = $attributes{$key};
      print " $key=$value";
    } # foreach my $key (keys (%attributes))
    print "}";
  } # if (keys (%attributes))

  print "\n";
  } # handleStartTag
```

```
./saxAirFrance2.pl ../AirFrance.xml |less


++OPENING: u{ n=1 who=O}
        STRING: ***Air France bonjour***
--CLOSING: u
        STRING: ******
        STRING: *** ***
++OPENING: u{ n=1 who=C}
        STRING: ***bonjour je voudrais
           faire uneréservation pour ***
++OPENING: vocal{ desc=e}
--CLOSING: vocal
        STRING: *** Londres***
--CLOSING: u
```

Restriction aux balises <u>

Quand on reçoit du texte (déclenchement de l'événement qui appelle la procédure associée), « *comment je peux savoir que je suis dans un balise <u> ?* »

Restriction aux balises <u>

Quand on reçoit du texte (déclenchement de l'événement qui appelle la procédure associée), « *comment je peux savoir que je suis dans un balise <u> ?* »
→ Gestion d'une variable globale.
Première possibilité : stocker le nom de la balise courante.

```
my $currentTag = "";

sub handleStartTag {
  shift; # discarding 1st arg
  $currentTag = shift @_;
} # handleStartTag

sub handleChar {
  shift;
  my $txt = shift;

  if ($currentTag eq "u") {
    print "\tSTRING: ***$txt***\n";
  }
} # handleChar
```

Restriction aux balises <u>

Quand on reçoit du texte (déclenchement de l'événement qui appelle la procédure associée), « *comment je peux savoir que je suis dans un balise <u> ?* »
→ Gestion d'une variable globale.
Première possibilité : stocker le nom de la balise courante.

```
my $currentTag = "";

sub handleStartTag {
  shift; # discarding 1st arg
  $currentTag = shift @_;
} # handleStartTag

sub handleChar {
  shift;
  my $txt = shift;

  if ($currentTag eq "u") {
    print "\tSTRING: ***$txt***\n";
  }
} # handleChar
```

```
<u who='0' n='1'>
   Bonjour <pause/>
   Je voudrais...
</u>
```

Restriction aux balises <u>

Quand on reçoit du texte (déclenchement de l'événement qui appelle la procédure associée), « *comment je peux savoir que je suis dans un balise <u> ?* »
→ Gestion d'une variable globale.
Première possibilité : stocker le nom de la balise courante.

```
my $currentTag = "";

sub handleStartTag {
  shift; # discarding 1st arg
  $currentTag = shift @_;
} # handleStartTag

sub handleChar {
  shift;
  my $txt = shift;

  if ($currentTag eq "u") {
    print "\tSTRING: ***$txt***\n";
  }
} # handleChar
```

```
<u who='0' n='1'>
   Bonjour <pause/>
   Je voudrais...
</u>
```

1. <u> → handleStartTag
   $currentTag <- "u"

Restriction aux balises <u>

Quand on reçoit du texte (déclenchement de l'événement qui appelle la procédure associée), « *comment je peux savoir que je suis dans un balise <u> ?* »
→ Gestion d'une variable globale.
Première possibilité : stocker le nom de la balise courante.

```
my $currentTag = "";

sub handleStartTag {
  shift; # discarding 1st arg
  $currentTag = shift @_;
} # handleStartTag

sub handleChar {
  shift;
  my $txt = shift;

  if ($currentTag eq "u") {
    print "\tSTRING: ***$txt***\n";
  }
} # handleChar
```

```
<u who='0' n='1'>
   Bonjour <pause/>
   Je voudrais...
</u>
```

1. <u> → handleStartTag
   $currentTag <- "u"
2. *Bonjour,* → handleChar
   $currentTag eq "u" : vrai → affichage

Restriction aux balises <u>

Quand on reçoit du texte (déclenchement de l'événement qui appelle la procédure associée), « *comment je peux savoir que je suis dans un balise <u> ?* »
→ Gestion d'une variable globale.
Première possibilité : stocker le nom de la balise courante.

```
my $currentTag = "";

sub handleStartTag {
  shift; # discarding 1st arg
  $currentTag = shift @_;
} # handleStartTag

sub handleChar {
  shift;
  my $txt = shift;

  if ($currentTag eq "u") {
    print "\tSTRING: ***$txt***\n";
  }
} # handleChar
```

```
<u who='0' n='1'>
   Bonjour <pause/>
   Je voudrais...
</u>
```

1. <u> → handleStartTag
   $currentTag <- "u"
2. *Bonjour,* → handleChar
   $currentTag eq "u" : vrai → affichage
3. <pause/> → handleStartTag
   $currentTag <- "pause"

Restriction aux balises <u>

Quand on reçoit du texte (déclenchement de l'événement qui appelle la procédure associée), « *comment je peux savoir que je suis dans un balise <u> ?* »
→ Gestion d'une variable globale.
Première possibilité : stocker le nom de la balise courante.

```
my $currentTag = "";

sub handleStartTag {
  shift; # discarding 1st arg
  $currentTag = shift @_;
} # handleStartTag

sub handleChar {
  shift;
  my $txt = shift;

  if ($currentTag eq "u") {
    print "\tSTRING: ***$txt***\n";
  }
} # handleChar
```

```
<u who='0' n='1'>
   Bonjour <pause/>
   Je voudrais...
</u>
```

1. <u> → handleStartTag
   $currentTag <- "u"
2. *Bonjour,* → handleChar
   $currentTag eq "u" : vrai → affichage
3. <pause/> → handleStartTag
   $currentTag <- "pause"
4. *Je voudrais* → handleChar
   $currentTag eq "u" : faux → pas d'affichage

Restriction aux balises <u>

Deuxième possibilité : gérer un booléen *« on est dans une balise u »*

```
my $uTag = 0;

sub handleStartTag {
  shift; # discarding 1st arg
  my $tagName = shift @_;
  $uTag = 1 if ($tagName eq "u");
} # handleStartTag

sub handleEndTag {
    shift;
    my $tagName = shift;
    $uTag = 0 if ($tagName eq "u");
} # handleEndTag

sub handleChar {
  shift;
  my $txt = shift;

  print "$txt\n" if ($uTag);
} # handleChar
```

Restriction aux balises <u>

Deuxième possibilité : gérer un booléen *« on est dans une balise u »*

```
my $uTag = 0;

sub handleStartTag {
  shift; # discarding 1st arg
  my $tagName = shift @_;
  $uTag = 1 if ($tagName eq "u");
} # handleStartTag

sub handleEndTag {
    shift;
    my $tagName = shift;
    $uTag = 0 if ($tagName eq "u");
} # handleEndTag

sub handleChar {
  shift;
  my $txt = shift;

  print "$txt\n" if ($uTag);
} # handleChar
```

```
<u who='0' n='1'>
   Bonjour <pause/>
   Je voudrais faire uneréservation pour
Londres
</u>

=>

Bonjour Je voudrais faire uneréservation pour
 Londres
```

Afficher le locuteur

```
my $uTag = 0;
my $currentSpeaker;

sub handleStartTag {
  shift; # discarding 1st arg
  my $tagName = shift @_;
  my %attributes = ();

  while ($#_ > 0) {
    my $attr  = shift @_;
    my $value = shift @_;
    $attributes{$attr} = $value;
  } # while ($#_ > 0)

  if ($tagName eq "u") {
    $uTag = 1;
    $currentSpeaker = $attributes{"who"};
  } # if ($tagName eq "u")
} # handleStartTag
```

```
sub handleChar {
  shift;
  my $txt = shift;

  if ($uTag) {
    print "[ $currentSpeaker ]: ";
    print "$txt\n";
  }
} # handleChar
```

Afficher le locuteur

```
my $uTag = 0;
my $currentSpeaker;

sub handleStartTag {
  shift; # discarding 1st arg
  my $tagName = shift @_;
  my %attributes = ();

  while ($#_ > 0) {
    my $attr  = shift @_;
    my $value = shift @_;
    $attributes{$attr} = $value;
  } # while ($#_ > 0)

  if ($tagName eq "u") {
    $uTag = 1;
    $currentSpeaker = $attributes{"who"};
  } # if ($tagName eq "u")
} # handleStartTag
```

```
sub handleChar {
  shift;
  my $txt = shift;

  if ($uTag) {
    print "[ $currentSpeaker ]: ";
    print "$txt\n";
  }
} # handleChar
```

```
[ O ]: Air France bonjour
[ C ]: bonjour je voudrais
       faire uneréservation pour
[ C ]:  Londres
[ O ]: excusez-moi
```

Uniquement les <u> contenant des pauses

On veut afficher le texte de :

```
<u who='O'>Bonjour<pause/>
          Je voudrais...</u>
```

Mais pas celui de :

```
<u who='O'>Bonjour,
          Je voudrais...</u>
```

SAX : gestion événementielle

- Quand on arrive sur une balise <u>, on ne peut pas savoir s'il y a une balise <pause/> après
- Quand on arrive sur une balise <pause/>, on ne peut pas « remonter » à la partie de texte située avant dans la balise <u>.

Uniquement les <u> contenant des pauses

On veut afficher le texte de :

```
<u who='O'>Bonjour<pause/>
          Je voudrais...</u>
```

Mais pas celui de :

```
<u who='O'>Bonjour,
          Je voudrais...</u>
```

SAX : gestion événementielle

- Quand on arrive sur une balise <u>, on ne peut pas savoir s'il y a une balise <pause/> après
- Quand on arrive sur une balise <pause/>, on ne peut pas « remonter » à la partie de texte située avant dans la balise <u>.
- $\rightarrow$ nécessité de stocker le texte dans un *buffer*
- on l'affiche si on a rencontré une pause (on en garde la trace dans une variable booléenne
- on l'affiche quand on a la totalité du texte (balise fermante </u>)

Uniquement les <u> contenant des pauses

```
my $uTag = 0;
my $currentSpeaker;
my $txtBuffer;
my $containsPause;

sub handleStartTag {
  shift;
  my $tagName = shift @_;
  # attributes handling here

  if ($tagName eq "u") {
    $uTag = 1;
    $currentSpeaker = $attributes{"who"};
    $txtBuffer = "";
    $containsPause = 0;
  } # if ($tagName eq "u")
  elsif ($tagName eq "pause") {
    $containsPause = 1;
    $txtBuffer .= " [PAUSE] ";
  } # elsif ($tagName eq "pause")
} # handleStartTag
```

```
sub handleChar {
  if ($uTag) {
    shift;
    my $txt = shift;
    $txtBuffer .= $txt;
  } # if ($uTag)
} # handleChar

sub handleEndTag {
  shift;
  my $tagName = shift;

  if ($tagName eq "u") {
    $uTag = 0 ;

    if ($containsPause) {
      print "[ $currentSpeaker ]: ";
      print "$txtBuffer\n";
    } # if ($containsPause) {
  } # if ($tagName eq "u")
} # handleEndTag
```

Dernière étape : gérer les subtilités

Uniquement les <u> contenant des pauses...

Solution adoptée : gérer un booléen « *on est dans une balise u* »

```
my $uTag = 0;

sub handleStartTag {
  shift;
  my $tagName = shift @_;
  $uTag = 1 if ($tagName eq "u");
} # handleStartTag

sub handleEndTag {
    shift;
    my $tagName = shift;
    $uTag = 0 if ($tagName eq "u");
} # handleEndTag

sub handleChar {
  shift;
  my $txt = shift;

  print "$txt\n" if ($uTag);
} # handleChar
```

Dernière étape : gérer les subtilités

Uniquement les <u> contenant des pauses. . .

Solution adoptée : gérer un booléen « *on est dans une balise u* »

```
my $uTag = 0;

sub handleStartTag {
  shift;
  my $tagName = shift @_;
  $uTag = 1 if ($tagName eq "u");
} # handleStartTag

sub handleEndTag {
    shift;
    my $tagName = shift;
    $uTag = 0 if ($tagName eq "u");
} # handleEndTag

sub handleChar {
  shift;
  my $txt = shift;

  print "$txt\n" if ($uTag);
} # handleChar
```

Cas particulier : les tours de parole imbriqués

```
<u who="C" n="37">j'aurai pas
  le temps d'aller dans une
  agence <u who="O" n="38">bon
  alors<pause/>arrivez un petit
  peu plus tôt à
  l'aéroport hein</u>
  <u who="C" n="38">d'accord</u>
  <who="O" n="39">pour prendre
  votre billet </u>
</u>
```

Dernière étape : gérer les subtilités

Uniquement les <u> contenant des pauses. . .

Solution adoptée : gérer un booléen *« on est dans une balise u »*

```
my $uTag = 0;

sub handleStartTag {
  shift;
  my $tagName = shift @_;
  $uTag = 1 if ($tagName eq "u");
} # handleStartTag

sub handleEndTag {
    shift;
    my $tagName = shift;
    $uTag = 0 if ($tagName eq "u");
} # handleEndTag

sub handleChar {
  shift;
  my $txt = shift;

  print "$txt\n" if ($uTag);
} # handleChar
```

Cas particulier : les tours de parole imbriqués

```
<u who="C" n="37">j'aurai pas
  le temps d'aller dans une
  agence <u who="O" n="38">bon
  alors<pause/>arrivez un petit
  peu plus tôt à
  l'aéroport hein</u>
  <u who="C" n="38">d'accord</u>
  <who="O" n="39">pour prendre
  votre billet </u>
</u>
```

Dans `handleEndTag` :

```
$uTag = 0 if ($tagName eq "u");
```

Dernière étape : gérer les subtilités

Uniquement les <u> contenant des pauses. . .

Solution adoptée : gérer un booléen « *on est dans une balise u* »

```
my $uTag = 0;

sub handleStartTag {
  shift;
  my $tagName = shift @_;
  $uTag = 1 if ($tagName eq "u");
} # handleStartTag

sub handleEndTag {
    shift;
    my $tagName = shift;
    $uTag = 0 if ($tagName eq "u");
} # handleEndTag

sub handleChar {
  shift;
  my $txt = shift;

  print "$txt\n" if ($uTag);
} # handleChar
```

Cas particulier : les tours de parole imbriqués

```
<u who="C" n="37">j'aurai pas
  le temps d'aller dans une
  agence <u who="O" n="38">bon
  alors<pause/>arrivez un petit
  peu plus tôt à
  l'aéroport hein</u>
  <u who="C" n="38">d'accord</u>
  <who="O" n="39">pour prendre
  votre billet </u>
</u>
```

Dans `handleEndTag` :

```
$uTag = 0 if ($tagName eq "u");
```

→ perte de « *d'accord pour prendre votre billet* »

Dernière étape : gérer les subtilités

Gestion du niveau de profondeur (imbrication) de balise <u> par une variable globale entière

```
my $uDepth = 0;

sub handleStartTag {
    $uDepth++  if ($tagName eq "u");
}

sub handleEndTag {
  if ($tagName eq "u") {
    $uDepth--;

    if (($uDepth == 0) and $containsPause) {
      print "$txtBuffer\n";
    } # if (($uDepth == 0) and $containsPause)
  } # if ($tagName eq "u")
} # handleEndTag
```

Dernière étape : gérer les subtilités

Gestion du niveau de profondeur (imbrication) de balise <u> par une variable globale entière

```
my $uDepth = 0;

sub handleStartTag {
    $uDepth++  if ($tagName eq "u");
}

sub handleEndTag {
  if ($tagName eq "u") {
    $uDepth--;

    if (($uDepth == 0) and $containsPause) {
      print "$txtBuffer\n";
    } # if (($uDepth == 0) and $containsPause)
  } # if ($tagName eq "u")
} # handleEndTag
```

```
sub handleChar {
  shift;
  my $txt = shift;

  $txtBuffer .= $txt if ($uDepth > 0);
} # handleChar
```