

Projet DILAF

Dictionnaires Langue Africaine – Français



Méthodologie DILAF

Étape 4 : Balisage à l'aide d'expressions rationnelles

Objectif

Baliser toutes les notices à l'aide des noms d'éléments précédemment choisis et en respectant la structure définie.

Exemple :

La notice

"a [áa] harafi na biyu a cikin haruffan hausa. *Sunan Ado yana farawa da a.*
Far.: voyelle a"

est transformée en :

```
<article id="a1"><kalma>a</kalma><furici>áa</furici><rukunin_ma_ana><ma_ana>
harafi na biyu a cikin haruffan hausa.</ma_ana><misali>Sunan Ado yana farawa da
a.</misali><makwatanci>voyelle a</makwatanci></rukunin_ma_ana></article>
```

plus lisible sous la forme indentée :

```
<article id="a1">
  <kalma>a</kalma>
  <furici>áa</furici>
  <rukunin_ma_ana raga="a1">
    <ma_ana>harafi na biyu a cikin haruffan hausa.</ma_ana>
    <misali>Sunan Ado yana farawa da a.</misali>
    <makwatanci>voyelle a</makwatanci>
  </rukunin_ma_ana>
```

</article>

Outils

Éditeur de texte : Notepad++

Éditeur XML : Exchanger XML Editor

Motivation

Baliser toutes les notices d'un dictionnaire en posant les balises une à une serait un travail monotone et harassant. L'usage d'expressions rationnelles permet le traitement en masse des notices. Toutefois, son usage nécessite un apprentissage préalable des expressions rationnelles.

Méthode

4.1 – Apprentissage des expressions rationnelles

Télécharger et étudier le fichier expressions_rationnelles.pdf.

4.2 – Prétraitement du fichier

Le fichier qui contient le dictionnaire est en fait une archive XML. Quelques manipulations sont nécessaires pour récupérer son contenu :

- changer l'extension "odt" en "zip"
- décompresser l'archive
- ouvrir l'archive
- le fichier "content.xml" contient le dictionnaire. Le renommer (par exemple : dictionnaire.xml)

Exemple :

hausa_extrait_v3.odt est sauvegardé sous le nom hausa_extrait_v4.odt.

Le nom hausa_extrait_v4.**odt** est changé en hausa_extrait_v4.**zip**

L'archive hausa_extrait_v4.zip est décompressée, ce qui fait apparaître un répertoire hausa_extrait_v4.

Dans le répertoire hausa_extrait_v4 se trouve le fichier content.xml.

Ouvrir le fichier content.xml.

Le sauvegarder sous le nom : hausa_extrait_v5.xml

4.3 – Segmentation du fichier en notices

Notepad++ permet d'ouvrir la dernière version du dictionnaire, de l'examiner, puis de la modifier.

La première vision de ce fichier peut être déconcertante : son contenu est devenu complexe et semble quasi illisible. En fait, celui-ci est structuré puisqu'il s'agit d'un fichier XML. Il est donc possible d'y repérer des balises (fermantes, ouvrantes) et leur rôle, et de s'appuyer sur ces balises pour transformer petit à petit son contenu.

La méthodologie préconise d'effectuer des sauvegardes du dictionnaire en numérotant les versions successives et de noter les transformations effectuées. Cette mémoire des traitements permet de repartir à tout moment d'une version antérieure et de refaire facilement l'ensemble des traitements.

Cette mémoire des traitements est essentielle car il est **normal**, durant ce processus de conversion, de faire des erreurs, de constater des oublis, et donc d'être obligé de reprendre les traitements à partir d'une version antérieure.

Le fichier XML, dans sa première version, ne contient que deux lignes. La première, très brève, est un en-tête XML. La seconde contient l'entièreté du dictionnaire. elle est donc très longue.

Il faut repérer la ou les balises qui marquent le début d'une notice afin de segmenter cette seconde ligne et d'obtenir une notice par ligne.

La partie qui précède la première notice pourra alors être ôtée.

La fin du document doit être examinée avec attention pour ôter les balises surnuméraires.

Exemple :

Un examen de `haus_a_extrait_v5.xml` permet de repérer `<text:p text:style-name="P1"><text:span text:style-name="Lexeme">` comme les balises marquant le début d'une notice.

Le fichier est segmenté en utilisant la fonction "Remplacer" :

`<text:p text:style-name="P1"><text:span text:style-name="Lexeme">` est placé dans le champ "Recherche"

`\n` est placé dans le champ Remplacer par.

La case "Mode étendu" est sélectionnée.

Ces transformations, ainsi que les suivantes, sont notées dans le fichier `transformations.odt` (consulter `transformations.pdf`).

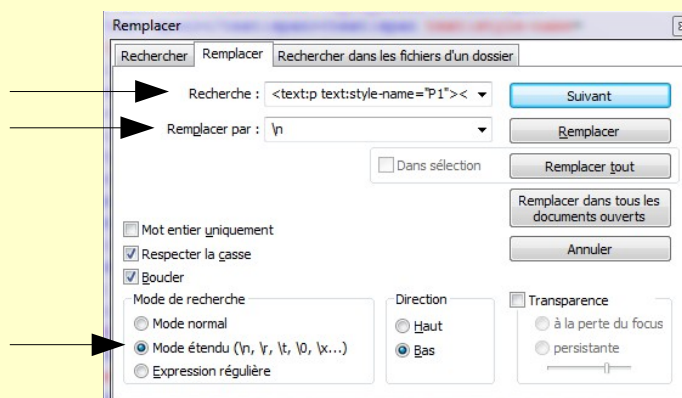


Figure 4.1 : Segmentation à l'aide de Notepad++

Les remplacements peuvent être effectués : les balises indiquées dans le champ "Recherche" sont remplacées par un retour à la ligne. Le dictionnaire est maintenant segmenté : chaque balise occupe une ligne.

La ligne entière qui précède la première notice est ôtée.(<office:document...)

Les balises suivantes, qui terminent le document, sont ôtées :
</office:text></office:body></office:document-content>

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v6.xml`

Lors de ces manipulations, il est conseillé d'effectuer prudemment un petit nombre de remplacements par quelques appuis sur le bouton Remplacer, et de vérifier que les remplacements souhaités sont bien effectués. Ensuite seulement appuyer sur le bouton Remplacer tout. Effectuer une vérification rapide du bon déroulement de cette opération.

4.4 – Balisage des notices

Les notices doivent être balisées en suivant la structure précédemment définie. Les balises sont posées à l'aide d'expressions rationnelles.

Pour réaliser ce balisage, nous allons procéder au fur et à mesure de la lecture du dictionnaire, en commençant par la première notice. Les remplacements nécessaires seront effectués grâce à une expression régulière afin de traiter potentiellement tout le dictionnaire.

Nous effectuerons trois types de balisage

- Pose des éléments de base
- Pose des éléments structurants
- Pose d'identifiants de notices et de sens.

Exemple :

Voici l'ensemble des opérations sur notre dictionnaire haoussa

* Entourer chaque notice des balises <article> et </article>

`^<text:span text:style-name="T1">(.)</text:p>$` est placé dans le champ "Recherche"

`<article>$1</article>` est placé dans le champ Remplacer par.

La case "Expression régulière" est sélectionnée.

Les remplacements sont effectués.

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v7.xml`

* Entourer chaque entrée par les balises <kalma> et </kalma>

`<article>([<]*)</text:span></text:span>` est placé dans le champ "Recherche"

`<article><kalma>$1</kalma>` est placé dans le champ Remplacer par.

La case "Expression régulière" est sélectionnée.

Les remplacements sont effectués.

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v8.xml`

* Suppression des espaces qui suivent l'entrée :

```
</kalma><text:span text:style-name="f_5f_standard"><text:span  
text:style-name="T1"> +<text:tab/></text:span></text:span> est placé  
dans le champ "Recherche"
```

`</kalma>` est placé dans le champ Remplacer par.

La case "Expression régulière" est sélectionnée.

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v9.xml`

Le détail des étapes suivantes figure dans le document `transformations.odt`. En voici le fil conducteur

* Balisage de la phonétique avec les balises `<furici>` et `</furici>`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v10.xml`

* Suppression des espaces qui suivent la phonétique

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v11.xml`

* Balisage de la définition avec les balises `<ma_ana>` et `</ma_ana>`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v12.xml`

* Suppression des espaces qui suivent la définition

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v13.xml`

* Balisage de l'exemple avec les balises `<misali>` et `</misali>`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v14.xml`

* Suppression des espaces qui suivent l'exemple

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v15.xml`

* Balisage de l'équivalent français avec les balises `<makwatanci>` et `</makwatanci>`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v16.xml`

Nous constatons que la première entrée "*a*" est entièrement balisée avec des balises de base. En revanche, ce n'est pas le cas de l'entrée suivante, "*a kalla*", pour laquelle nous procédons de la même manière.

* Balisage de la classe avec les balises `<nau_i>` et `</nau_i>`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v17.xml`

* Suppression des espaces qui suivent la classe

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v18.xml`

* Présence d'un point, qui suit la définition, et qu'il faut rattacher à la définition en le plaçant avant l'élément `</ma_ana>`.

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v19.xml`

* Suppression des espaces qui suivent la définition

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v20.xml`

La seconde notice "*a*" ainsi que "*a kalla*" sont entièrement balisées avec des balises de base. Nous observons la notice suivante : "*abacada*".

* Balisage du genre avec les balises `<jinsi>` et `</jinsi>`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v21.xml`

* Rattachement du point qui suit le genre

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v22.xml`

* Suppression des espaces qui suivent la définition

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v23.xml`

"*abacada*" est balisée avec des balises de base. Nous observons la notice suivante : "*abara*".

Dans le dictionnaire original "*abara*" porte le numéro 1 et la seconde entrée "*abara*" (qui la suit) porte le numéro 2. Ces numéros peuvent être repris pour fabriquer l'identifiant de la notice qui sera indiqué comme valeur de l'attribut "id".

* Insertion des identifiants numérotés comme valeur de l'attribut "id"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v24.xml`

* Insertion des identifiants des entrées non numérotées.

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v25.xml`

* Les apostrophes apparaissent sous la forme d'une entité `'`. Nous rétablissons des apostrophes sous la forme d'un unique caractère : `'`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v26.xml`

* Normalisation des noms des identifiants : remplacement des espaces et des apostrophes par un tiret bas (les apostrophes sont notées par l'entité `'`).

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v27.xml`

* Suppression des espaces qui suivent le numéro

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v28.xml`

* Balisage de la variante avec les balises `<yare>` et `</yare>`

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v29.xml`

* Rattachement du point qui suit la variante

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v30.xml`

* Suppression des espaces qui suivent la variante

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v31.xml`

les deux entrées "*abara*" ainsi que "*abarba*" et "*ada*" sont maintenant balisées avec des balises de base.

Mais des caractères bizarres apparaissent dans "*ada*". Un examen de dictionnaire original nous apprend que ces caractères sont des tirets.

Bien suivre la méthodologie implique de revenir à l'étape 3, d'inclure ce nouveau caractère dans le fichier notant les remplacements, et de reprendre l'étape 4 depuis le début.

Comme notre dictionnaire est ici très petit, les remplacements sont effectués dans la version courante du dictionnaire en vérifiant que chaque remplacement (il n'y en a que deux) est pertinent (il faut se reporter au dictionnaire original).

Le fichier des caractères remplacés est sauvegardé sous le nom `hausa_remplacer_unicode_v2.odt`.

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v32.xml`

Nous passons à la notice "*adabali*".

* Balisage du pluriel avec les balises `<jam_i>` et `</jam_i>`

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v33.xml`

* Rattachement du point qui suit le pluriel

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v34.xml`

* Suppression des espaces qui suivent le pluriel

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v35.xml`

"*adabali*" et "*adada*" sont maintenant balisées avec des balises de base. Nous observons la notice suivante : "*adadi*".

Cette entrée distingue deux sens qui sont numérotés et qu'il faut encadrer par des balises de bloc sémantique (`rukunin_ma_ana`) avec un identifiant de sens.

L'expression rationnelle adéquate est complexe car il s'agit d'un élément structurant qui va encadrer d'autres éléments. La difficulté est de bien placer la balise fermante du bloc sémantique.

* Balisage de blocs sémantiques numérotés avec l'élément `<rukunin_ma_ana>`.

Le numéro du sens, qui apparaît dans le dictionnaire original, est repris pour fabriquer l'identifiant du bloc sémantique (valeur de l'attribut "id").

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v36.xml`

* Suppression des espaces qui suivent un bloc sémantique

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v37.xml`

* Balisage de blocs sémantiques numérotés avec l'élément `<rukunin_ma_ana>`.

Le numéro du sens est repris pour fabriquer l'identifiant du bloc sémantique.

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v38.xml`

Les notices "*adadi*", "*adaka*", "*adalci*" et "*adali*" sont balisées avec des éléments de base. Nous observons la notice suivante : "*adana*".

* Balisage de l'information morphologique avec l'élément `<siffolin_kalma>`.

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v39.xml`

* Rattachement du point qui suit l'information morphologique

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v40.xml`

* Suppression des espaces qui suivent l'information morphologique

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v41.xml`

"*adana*" est balisée avec des éléments de base. Passons à la notice suivante : "*adananne*".

* Suppression des espaces qui suivent la phonétique

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v42.xml`

* Suppression des espaces qui suivent la classe

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v43.xml`

* Suppression des espaces qui suivent la définition

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v44.xml`

* Suppression des espaces qui suivent l'exemple

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v45.xml`

* Balisage des formes au féminin avec l'élément `<mace>`.

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v46.xml`

* Rattachement du point qui suit le féminin

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v47.xml`

* Suppression des espaces qui suivent le féminin

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v48.xml`

* Suppression des espaces qui suivent le pluriel

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v49.xml`

* Suppression des espaces qui suivent le féminin

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v50.xml`

Les notices sont balisées avec des éléments de base jusqu'à "*ba'a*". Nous observons la notice suivante : "*ba'abzine*".

* Suppression des espaces qui suivent le genre

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v51.xml`

Toutes les notices sont balisées avec des éléments de base mais certaines sont incomplètes : il manque encore le bloc sémantique.

Celui-ci doit être posé avec prudence car il englobe des éléments de base. Son identifiant doit également être construit. Nous procédons en commençant par la

première notice.

"adana" est balisée avec des éléments de base. Passons à la notice suivante : "adananne".

* Pose du bloc sémantique dans les notices de même structure que "a"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v52.xml`

Le bloc sémantique de la première notice est balisé. Nous passons à la notice suivante.

* Pose du bloc sémantique dans les notices de même structure que la seconde notice "a"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v53.xml`

Les trois premières notices sont bien balisées. Nous passons à la notice suivante "abacada". Le genre, balisé <jinsi> doit être déplacé hors du bloc sémantique.

* Pose du bloc sémantique dans les notices de même structure que "abacada"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v54.xml`

Traitement de la notice suivante, "abara". Le genre, balisé <jinsi>, et la variante, balisée <yare>, doivent être déplacés hors du bloc sémantique. Conformément à la liste des éléments de base, le genre sera placé avant la variante.

* Pose du bloc sémantique dans les notices de même structure que "abara"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v55.xml`

Traitement de la deuxième notice "abara". Le genre, balisé <jinsi> doit être déplacé hors du bloc sémantique.

* Pose du bloc sémantique dans les notices de même structure que la deuxième notice "abara"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v56.xml`

Traitement de la notice suivante, "abarba". Le genre, balisé <jinsi> doit être déplacé hors du bloc sémantique.

* Pose du bloc sémantique dans les notices de même structure que "abarba"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v57.xml`

Traitement de la notice suivante, "ada". Le genre, balisé <jinsi> doit être déplacé hors du bloc sémantique.

* Pose du bloc sémantique dans les notices de même structure que "ada"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v58.xml`

Traitement de la notice suivante, "adabali". Le genre, balisé <jinsi>, et le pluriel, balisé <jam_i>, doivent être déplacés hors du bloc sémantique. Conformément à la liste des éléments de base, le genre est laissé devant le pluriel.

* Pose du bloc sémantique dans les notices de même structure que "adabali"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v59.xml`

La prochaine notice sans bloc sémantique balisée, "adana", est traitée. L'information morphologique, balisée `<siffofin_kalma>` doit être déplacée hors du bloc sémantique.

* Pose du bloc sémantique dans les notices de même structure que "adana"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v60.xml`

La prochaine notice sans bloc sémantique balisée, "adananne", est traitée. Le pluriel, balisé `<jam_i>`, et le féminin, balisé `<mace>`, doivent être déplacés hors du bloc sémantique. Conformément à la liste des éléments de base, le pluriel sera placé avant le féminin.

* Pose du bloc sémantique dans les notices de même structure que "adananne"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v61.xml`

La seconde notice "ba" sans bloc sémantique balisée, est traitée. La variante, balisée `<yare>` doit être déplacée hors du bloc sémantique.

* Pose du bloc sémantique dans les notices de même structure que la seconde notice "ba"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v62.xml`

La prochaine notice sans bloc sémantique balisée, "ba'abzine", est traitée. Le genre, balisé `<jinsi>`, le féminin, balisé `<mace>`, et le pluriel, balisé `<jam_i>`, doivent être déplacés hors du bloc sémantique. Conformément à la liste des éléments de base, le pluriel sera placé avant le féminin.

* Pose du bloc sémantique dans les notices de même structure que "ba'abzine"

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v63.xml`

Toutes les notices sont balisées.

4.5 – Balisage du dictionnaire

Il est nécessaire d'encadrer l'ensemble des notices par un unique élément `<dilaf>`. Pour un dictionnaire bilingue, celui-ci possède quatre attributs :

- *src* (pour source) note la langue source du dictionnaire
- *trg* note la langue cible (en cas de dictionnaire bilingue)
- *version* note le numéro de version
- *datecreation* note la date de création de cette version du dictionnaire

* Ajout des balises de fin et de début du dictionnaire

Le dictionnaire est sauvegardé sous le nom `hausan_extrait_v64.xml`

4.6 – Vérification du balisage

4.6.1 – Document bien formé

Un document XML est bien formé si ses balises respectent la syntaxe XML : les balises ouvertes sont bien fermées, il n'y a pas d'éléments enchâssés, etc.

Ouvrir le dictionnaire avec Exchanger XML Editor. Cet éditeur vérifie que le document est bien formé. Si nécessaire, il signale les éventuelles erreurs.

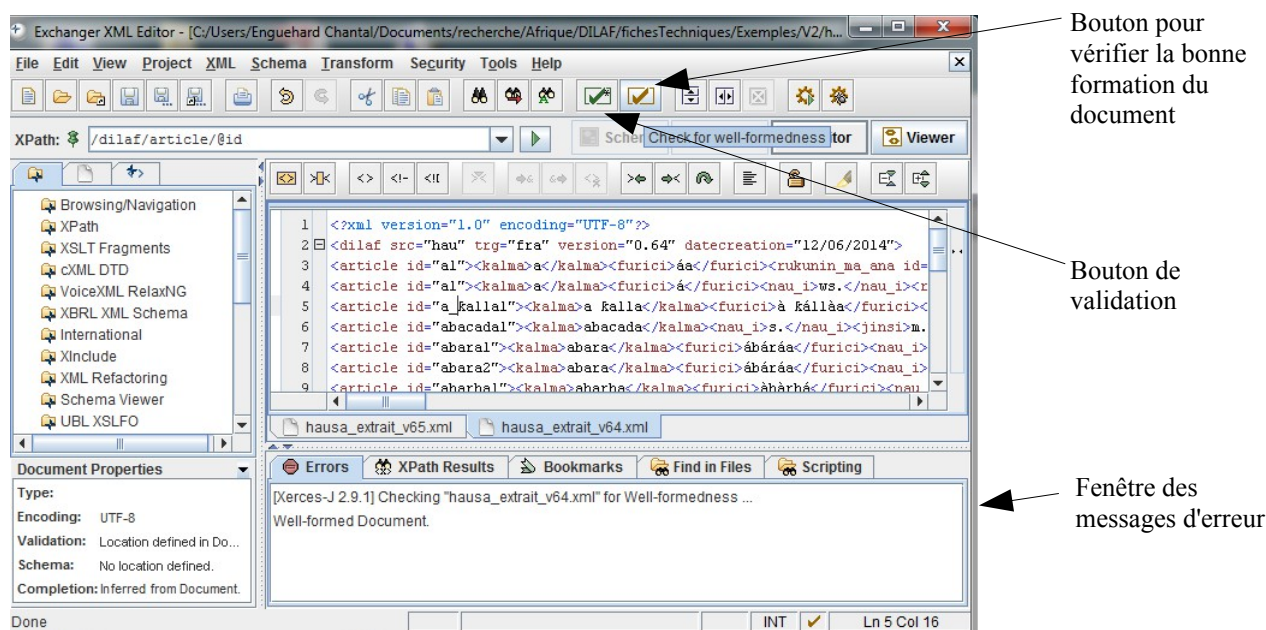


Figure 4.2 : Exchanger XML Editor

* Ouverture du dictionnaire : aucune erreur ne s'affiche
Le dictionnaire haus_a_extrait_v64.xml est bien formé.

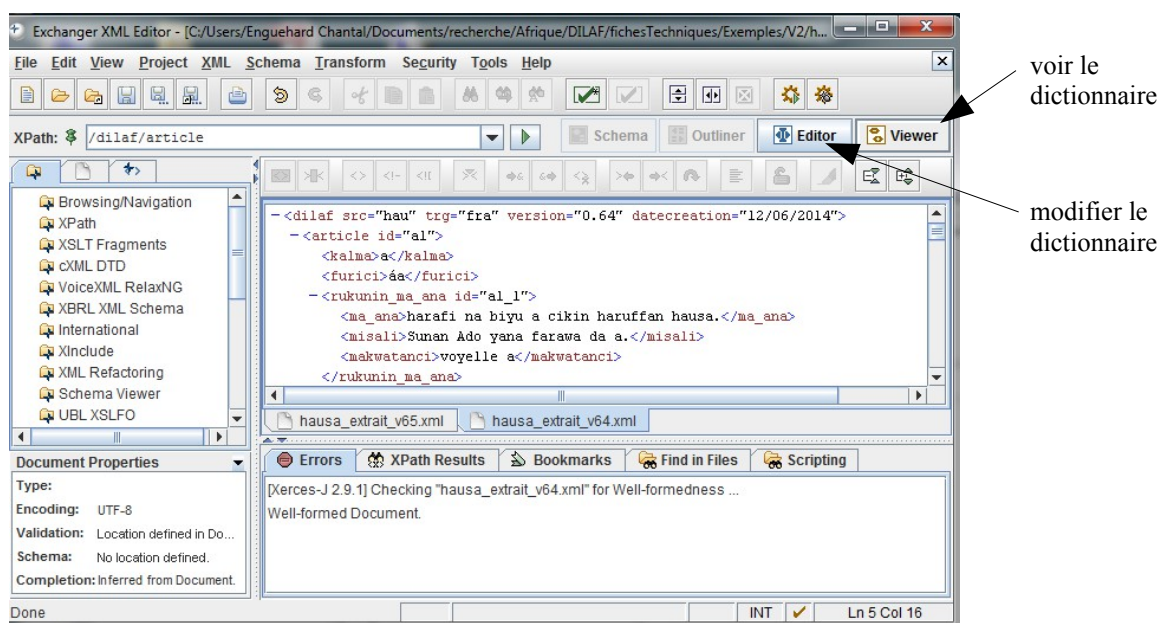


Figure 4.3 : Exchanger XML Editor

Si le dictionnaire est bien formé Exchanger XML Editor permet de le visualiser sous une forme indentée. Celle-ci est beaucoup plus lisible que la forme non indentée. Il suffit de cliquer sur le bouton "Viewer".

Pour modifier le dictionnaire, il faut cliquer sur le bouton "Editor".

4.6.2 – Document valide

Un document XML est valide si ses balises respectent la structure qui a été précédemment définie (étape 3).

Cette structure doit être notée sous une forme compréhensible par un ordinateur, appelée DTD (Document Type Definition). Celle-ci sera utilisée pour vérifier automatiquement que le document est conforme à cette structure.

Un [tutoriel](#) explique comment faire et utiliser une DTD.

Exemple :

Nous avons établi la liste des parties composant une notice. Il faut y ajouter que l'ensemble des notices est contenu dans un dictionnaire. Nous créons donc l'élément <dilaf> qui rassemble toutes les notices. Cet élément possède quatre attributs src, trg, version et datecreation permettant respectivement de noter la langue source, la langue cible (car le dictionnaire est bilingue), le numéro de version et la date de dernière modification de cette version.

Cette structure peut être exprimée dans une DTD que nous établissons par étapes.

* Définition du contenu de l'élément <dilaf> qui contient des articles :

```
<!ELEMENT dilaf (article)*>
```

* Définition des attributs de l'élément <dilaf>

```
<!ATTLIST dilaf src CDATA #REQUIRED>
```

```
<!ATTLIST dilaf trg CDATA #REQUIRED>
```

```
<!ATTLIST dilaf version CDATA #REQUIRED>
```

```
<!ATTLIST dilaf datecreation CDATA #REQUIRED>
```

Une notice est composée d'une vedette suivie de sa phonétique. Éventuellement sont indiqués une classe, un genre, une forme au pluriel, une forme au féminin, une information morphologique, une variante et un ou plusieurs blocs sémantiques.

* Définition du contenu de l'élément <article> :

```
<!ELEMENT article (kalma, furici, nau_i?, jinsi?,  
jam_i?, mace?, siffolin_kalma?, yare?,  
rukunin_ma_ana*)>
```

* Définition de l'attribut id de l'élément <article>

```
<!ATTLIST article id ID #REQUIRED>
```

Chacun des éléments composant un article doit maintenant être défini. La plupart ne contiennent qu'une information.

* Définition des éléments de <article> :

```
<!ELEMENT kalma (#PCDATA)>
```

```
<!ELEMENT furici (#PCDATA)>
```

```
<!ELEMENT nau_i (#PCDATA)>
```

```
<!ELEMENT jinsi (#PCDATA)>
```

```
<!ELEMENT jam_i (#PCDATA)>
```

```
<!ELEMENT mace (#PCDATA)>
```

```
<!ELEMENT siffolin_kalma (#PCDATA)>
```

```
<!ELEMENT yare (#PCDATA)>
```

Le bloc sémantique contient une définition, un exemple d'usage et un équivalent français

* Définition du contenu d'éléments de <rukunin_ma_ana> :

```
<!ELEMENT rukunin_ma_ana (ma_ana, misali, makwatanci)>
```

* Définition de l'attribut id de l'élément <rukunin_ma_ana>

```
<!ATTLIST rukunin_ma_ana id ID #REQUIRED>
```

* Définition des éléments de <rukunin_ma_ana> :

```
<!ELEMENT ma_ana (#PCDATA)>
```

```
<!ELEMENT misali (#PCDATA)>
```

```
<!ELEMENT makwatanci (#PCDATA)>
```

Cette première version de la DTD sera ajustée au fur et à mesure de la validation du document.

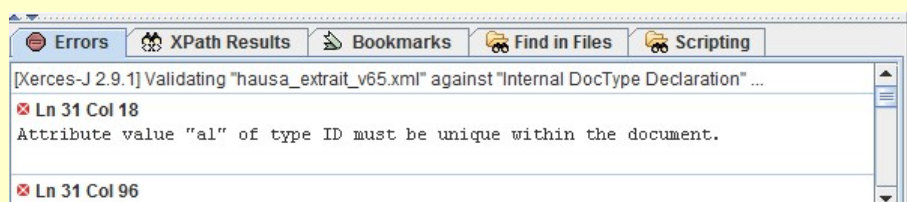
La DTD est ajoutée au document. Elle est nommée `dilaf` car l'élément `<dilaf>` contient tous les autres éléments.

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v65.xml`

Exchanger XML Editor peut vérifier si le document est bien conforme à la structure définie dans sa DTD. Pour ce faire, il faut cliquer sur le bouton de validation (voir figure 4,2).

* Clic sur le bouton de validation

Une erreur s'affiche : l'identifiant d'article "`a1`" présent à la ligne 31 existe déjà.



Effectivement, c'est le deuxième identifiant d'article qui est nommé "`a1`", or un identifiant doit être unique. Il est corrigé en "`a2`".

Il peut être sauvegardé sous un nouveau nom en cliquant sur File > Save As ...

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v66.xml`

* Clic sur le bouton de validation

Une erreur s'affiche : l'identifiant de sens "`a1_1`" présent à la ligne 31 existe déjà. Il est corrigé en "`a2_1`".

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v67.xml`

* Clic sur le bouton de validation

Une erreur s'affiche : l'article de ligne 33 n'est pas conforme à la DTD car il ne contient pas de phonétique. La DTD est modifiée pour rendre la phonétique optionnelle (ajout d'un point d'interrogation après `furici` dans la définition de l'élément `article`)

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v68.xml`

* Clic sur le bouton de validation

Une erreur s'affiche : l'article de ligne 35 n'est pas conforme à la DTD car son bloc sémantique ne contient pas d'exemple. La DTD est modifiée pour rendre l'exemple optionnel dans la définition de l'élément `rukunin_ma_ana`.

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v69.xml`

* Clic sur le bouton de validation

Une erreur s'affiche pour l'entrée "`adadi`" car ses deux blocs sémantiques contiennent des informations qui ne devraient pas s'y trouver : le genre et le pluriel. Comme le pluriel n'est indiqué que pour le deuxième sens, l'entrée doit être scindée en deux entrées identifiées par "`adadi1`" et "`adadi2`" et contenant chacune un unique bloc

sémantique.

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v70.xml`

* Clic sur le bouton de validation

Une erreur s'affiche pour la deuxième entrée "*ba*" car son identifiant "*ba1*" existe déjà. Il est corrigé en "*ba2*", l'identifiant du bloc sémantique "*ba1_1*" est corrigé en "*ba2_1*".

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v71.xml`

* Clic sur le bouton de validation

Une erreur similaire s'affiche pour la deuxième entrée "*babbake*". Les identifiants sont corrigés.

Le dictionnaire est sauvegardé sous le nom `hausa_extrait_v72.xml`

* Clic sur le bouton de validation

Le dictionnaire est maintenant valide.