Commentaires et spécifications pour l'éditeur d'annotations

Ce fichier contient deux parties d'une part les commentaires de la version courante, d'autre part une spécification des parties qui restent à implémenter (ou à modifier)

Remarques sur la version 7.1 de l'éditeur

Historique

Modifications majeures effectuées depuis la 7.0

- Correction des nombreux bugs introduits dans la 7.0
- Refonte du format de sauvegarde des préférences
- Implementation de l'insertion automatique de balises
- Option «□ouveau document□ dans le menu File

Modifications majeures effectuées depuis la 6.1

- Exportation des documents en RTF et HTML
- Impression des documents
- L'éditeur d'attribut est directement actif lors de l'insertion d'une balise
- Début d'implémentation d'une gestion contextuelles des balises applicables
- Boite de confirmation avant quitter l'éditeur
- Nouvelle boite de gestion des préférences
- Insertion d'une image d'accueil

Modifications majeures effectuées depuis la 6.05

- Amélioration des préférences
- Possibilité de ne pas faire apparaître la balise fermante dans le code XML

Modifications majeures effectuées depuis la 6.0

- Intégration au source des nouvelles icônes
- Suppression des ralentissements lors du chargement d'un fichier de texte
- Ajout d'un fichier de préférences

Modifications majeures effectuées depuis la 5.22

- La sauvegarde du fichier annoté est effectuée dans un format "XML" (au lieu de binaire)
- Début d'implémentation de boutons de gestion de style dans la barre d'icônes
- Dimensionnement de la fenêtre au démarrage (la barre de titre est visible)
- Suppression de "l'inverse vidéo" systématique pour visualiser la balise courante
- Le scroll semble marcher correctement (le texte est entièrement visible)
- Correction du bug sur le <CR> mal pris en compte dans la zone XML
- Correction du bug sur la non remise à jour des attributs lors de l'application d'une balise
- Commande QUIT dans le menu FILE

Liste des remarques

Les différentes remarques sont étiquettées de la manière suivante :
- BUG : un problème à corriger
- MOD : une modification du comportement de l'éditeur
- IMP : implémentation d'une nouvelle fonctionnalité
- HYP : une possibilité d'implémentation future qu'il faut spécifier

Les nouvelles remarques (ou modifications) par rapport au fichier précédent sont marquées d'un trait dans la marge droite … Les remarques précédemment effectuées et toujours pas prises en compte (ou discutées par mail) sont marquée d'un trait rouge !

Remarques générales

Avant d'implémenter (ou de modifier) une fonction de l'éditeur, penser à se mettre à la place de l'utilisateur final et imaginer des scénarios d'utilisation du système ...

Avant d'implémenter de nouvelles fonctionnalités dans l'éditeur bien vérifier
- que TOUS les bugs précédents ont été corrigés
- que les spécifications proposées sont bien suivies

Fonctions générales

Gestion des préférences de l'éditeur

MOD : La modification de la police associée au balise dans la zone XML («□ag body□) modifie les styles associés à toutes les balises. C'est génant car on perd tout les styles. Il faut que mla modification ne porte QUE sur les balises de la zone XML.

MOD : La modification de la police associée à «□nnotated text□ modifie les styles associés à toutes les balises. Comme ci-dessu c'est génant, la modification ne doit porter que sur les balises qui utilisent le style par défaut

IMP : Dans la partie «□ptions□ des préférences, il serait bien de pouvoir paramétrer le nombre de caractères que l'on affiche pour les «□ag body□ et les «□ample□ext□. Par exemple pour tronquer les «□ag body□ à 30 caractères et les «□ample□ext□ à 50 …

BUG : Le chemin d'accès courant («□orking folder□) n'est pas pris en compte dans un certain nombre de fonction d'entrée-sortie notamment le chargement d'un document de texte. *OK, semble-être un bug lié à ma JVM car en changeant d'OS c'est OK□!*

Zone de texte (haut à gauche)

BUG : lorque l'on insert du texte (manuellement ou par copier/coller) on observe une dérive des balises en aval (les pointeurs de début et fin ne sont pas mis à jour). Il me semblait que ce problème avait été résolu dans une version précédente de l'éditeur, mais j'ai retester la 6.1 et le problème est déjà présent.□Bref à corriger …

PS□pourrais-tu me communiquer les références de l'API du composant que tu utilises pour implmenter la zone de texte dans l□éditeur□

(IMP : TRES IMPORTANT) : Lorsque l'on fait un copier/coller, il faut que les balises incluses soient elles aussi copiées. En fait dans l'application en génomique l'utilisateur est amené à copier plusieurs fois chaque paragraphe de manière à le "baliser" selon toutes les interprétations biologiques qui sont possibles.

MOD : Prendre en compte **l'ensemble** des spécifications de la manière dont devrait être

effectuée la gestion des balises dans l'éditeur (*cf. seconde partie de ce document*) ... En particulier le fait que si deux zones contigues ont les mêmes étiquettes il faut que les deux zones fusionnent.

Zone XML (bas)

Il semble que cette partie soit bourrée de bugs dans cette version de l'éditeur

BUG : Lorqu'on sélectionne une étiquette XML, le scroll sur la zone de texte n'est toujours pas correct à 100% la partie haute sélectionnée reste en dehors de la fenêtre

BUG: Lorsque l'on déplace le curseur d'insertion dans la zone texte il faut sélectionner la ligne correspondante du code XML

BUG (régression) : Sur certain système (Linux et maintenant aussi sous OSX) la zone du bas à gauche n'est pas synchronisée du tout avec le scroll sur le texte (A TESTER ...).

BUG (Invraisemblable) : lorsque je change l'état de la check-box «□nding balise□ dans les préférences je perds l'ensemble de mes styles. Testé 2 fois□ Franchement je ne vois pas le rapport entre les eux choses, vérifie si tu as aussi ce bug□!!

Zone édition d'attributs

MOD : Le bouton d'effacement d'une étiquette (le nommer "Delete current Tag") ne doit pas être inclu dans la Scroll-list contenant les attributs éditables mais être placé en dessous (et néanmoins toujours dans la zone d'édition des attributs bien évidemment).

IMP : Lorsque la valeur d'un attribut est un URL il serait intéressant qu'un simple clique sur le nom de l'attribut déclenche le lancement d'un navigateur et le chargement de la page désignée par cet URL. Le but de cette fonction est de permettre «□n retour à la source□ lorsque la partie de texte en cours d'annotation est ambiguë (A DISCUTER)

Sauvegarde et Rechargement des document annoté

IMP : Sauvegarder les styles en adoptant soit un format texte "ad-hoc" soit mieux en utilisant le format adopté pour les feuilles de styles CSS ... *Faire le choix de manière à faciliter l'implémentation d'une commande d'impression (cf. «□nction générales□)*

IMP : Il faudrait que le format de sauvegarde soit toujours le même que le format XML de la DTD. Cela permettrait de s'affranchir de la commande "Export XML". Les informations supplémentaires à inclure (cf ci-dessous) pourraient apparaître en début de fichier sous la forme de commentaires afin de respecter le format.

MOD : (IMPORTANT) Concernant la sauvegarde des fichiers annotés, il serait bien de *ne plus inclure la définition explicite* des styles dans le fichier (voir la discussion spécifique sur la gestion des feuilles de styles). Par contre il FAUT sauvegarder la liste des styles (des fichiers) qui étaient chargés dans l'interface de manière à les recharger automatiquement.

Exportation en HTML

MOD : La gestion des «□euilles de styles□ généré par la biliothèque JAVA est curieuse … Prepéter à chaque fois leur description dans le tag <SPAN>□et en ajoutant en prime des <U> pour gérer le soulignement) ? Bref plutôt que des horreurs du genre□

```
<p class=default>
      <span style="color: #0000ff; font-size: 12pt; font-family: Arial">
        <u>P2- In contrast, both the E213G and E213A Spo0A variants
          showed decreased binding and completely abolished
          transcriptional activation of #spoIIa# and #spoIIE#, while the R214G
```

```
          and R214A variants completely abolished both DNA binding
          and transcriptional activation.
        </u>
      </span>
    </p>
```

Il faudrait plutôt implementer une «□raie□» utilisation des feuilles de styles, dans laquelle

1) on déclare les styles dans le HEADER et
2) on les utilise dans le BODY …

```
<style type="text/css" media="screen"><!--
.abstract  { color: blue; font-style: italic; font-family: Arial; text-decoration:
underline }
--></style>

</head>
<body bgcolor="#ffffff">
<p>
      <span class="abstract">P2- In contrast, both the E213G and E213A Spo0A
variants  showed decreased binding and completely abolished transcriptional
activation of #spoIIa# and #spoIIE#, while the R214Gand R214A variants completely
abolished both DNA binding and transcriptional activation.
     </span>
 </p>
```

Pour l'annecdote notons que le code actuel fait planter certains browsers□!

De même dans la sauvegarde des préférences pourquoi utiliser des formats persos de feuilles de styles plutôt que celui proposé par les CSS. C'est «□ coup sur□» aller au devant de problèmes (moins de compatibilité et plus de code à écrire).

Gestion des styles

Non implémenté. *Voir les* ***spécifications du menu style*** … Dans un premier temps implémente au moins la structure du menu même si il n'y a pas de fonctions attachées, c'est important d'avoir une idée précise d'une interface avant d'implémenter ses fonctions …

Par ailleurs, il faudrait aussi faire la boite de dialogue de gestion des styles avec les deux onglets pour régler l'aspect des styles et la gestion de la completion …

MOD : La gestion des paramètres de completions semble correcte (mais je ne l'ai pas testé intensivement). Par contre au niveau visuel, plutot que faire «□pparaître□» les options, il vaut mieux que les options «□on actives□» soit affichées en grisée (j'imagine qu'il y a la notion de widget actif et inactif en JAVA□?)

IMP : ne pas oublier dans la sauvegarde d'un style de sauver les paramètres «□'aspects□» et ceux du «□omportement□».

Validation du code XML

IMP : Il serait bien de pouvoir faire une validation partielle du code XML par rapport à la balise courante. Par exemple l'utilisateur sélectionne une partie de texte déjà balisée (via la fenêtre inférieure gauche) puis lance la validation de cette seule balise (et de celles qu'elle contient). Cela permettrait de vérifier simplement la cohérence locale

Gestion des préférences

Il faut bien séparer les différents aspects à paramétrer dans l'éditeur. On peut en distinguer aujourd'hui au moins quatre :

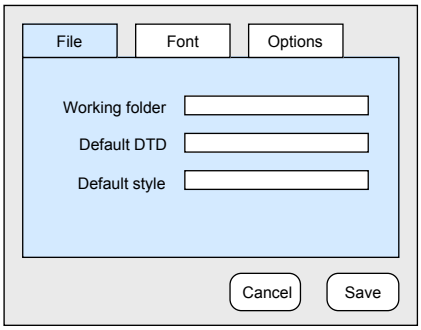
- Les fichiers par défaut : le dossier courant, la DTD et la feuille de styles associée
- Les polices par défaut : affichage des parties non dépendantes des feuilles de style
- Les couleurs par défaut : couleurs de polices et fond proposées dans les styles
- Les options d'affichages□par exemple la check box de suppression de balises de fin

Concernant les couleurs il n'est pas prioritaire de permettre une édition directe et on se contentera pour en modifier la liste d'éditer le fichier de préférence "xmljava.conf".

Par contre il semble nécessaire de structurer la boîte "Préférences" à l'aide "d'onglets" (je ne connais pas la dénomination en JAVA : Tab ?) de manière à faciliter la tâche de l'utilisateur (et du programmeur car on simplifie les problèmes de mise en place). Pour l'instant il y en aura trois : "File", "Font" et "Options". Voici la description de chacun :

* *Pour l'onglet File :*

On indique le nom du dossier de travail par défaut (le chemin d'accès utilisé par les boîtes de lecture/sauvegarde) et le nom de la DTD par défaut et de la feuille de style par défaut qui lui est associé. Ces différents champs peuvent bien sur rester vide auquel cas il ne se passe rien ...



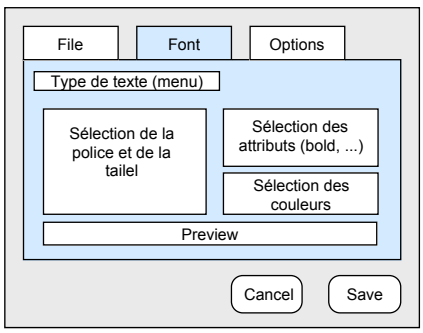
Par la suite, si besoin d'autres champs de configuration de fichier par défaut pourront être ajouté à cette boîte …

* *Pour l'onglet Font :*

Les préférences de styles de textes doivent porter sur les 4 aspects suivants :

- Pour la fenêtre supérieure gauche (le texte)
  - Police par défaut du texte non annoté ("Plain text")
  - Police par défaut du texte annoté ("Annotated text")
- Pour la fenêtre inférieure gauche (le balisage)
  - Police par défaut des balises ("Tag body")
  - Police par défaut des "fragments de phrases" ("Sample text")

Au niveau de la boite de dialogue le plus simple est d'introduire un menu «□op-up□ contenant les quatre type de textes mentionnés ci-dessus avec en dessous le composant de sélection de police. C'est lui qui permet de sélectionner la police en cours de définition. Par ailleurs il serait raisonnable que le sélecteur de police utilisé ici soit le même que celui qui est décrit pour les ***feuilles de styles*** (gain de cohérence pour l'interface et gain en coût de développement).



* *Pour l'onglet Options :*

Pour l'instant il ne contient que la « check-box » contrôlant l'apparition des balises de fin dans la zone d'affichage du code XML …

Implémentation du menu « Style »

Le menu « style » correspond à un renommage du menu « balise »

Ce menu doit contenir les commandes suivantes (avec des icônes spécifiques) :

- New style : création d'un nouveau style en se basant sur le style par défaut et ajout d'un nouveau bouton dans la barre des icônes. Idéalement chaque nouveau style doit avoir un nom (qui sera utilisé lors de la sauvegarde) et ce nom doit être utilisé pour étiqueter le bouton correspondant dans la barre des icônes.

- Load style : charge un fichier de style précédemment sauvé et ajoute un automatiquement un nouveau bouton dans la barre des icônes (à la fin) dont l'étiquette est le nom du fichier (sans l'extension du fichier bien évidemment)

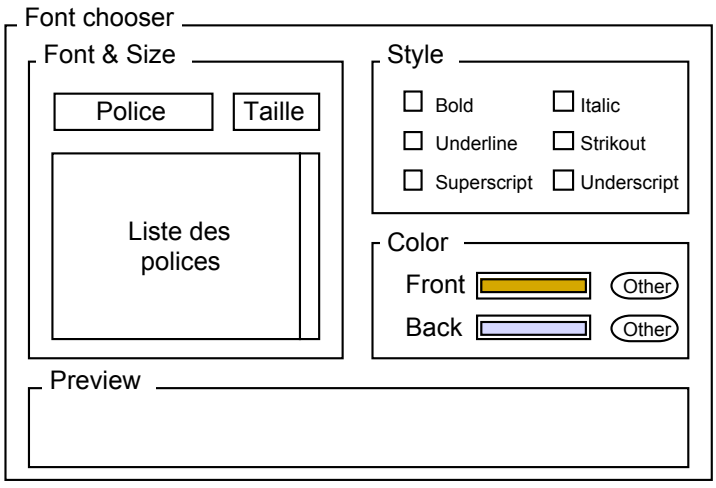
- Save current style : Sauve le style couramment utilisé (actif). Encore une fois la sauvegarde doit s'effectuer en utilisant un format « non binaire» (par exemple en utilisant du XML avec des CSS).

- Delete current style : Supprime le style courant (mais pas le fichier), on utilise alors le style précédent dans la liste (ou le dernier si on est début de liste), on doit toujours laisser au moins un style dans la barre (style par défaut par exemple) ...

- Edit Style : Appelle la boîte d'édition de style décrites ci-dessous

Le style courant, c'est-à-dire le style qui est *effectivement utilisé* pour colorer les balises, doit être signalé d'une manière ou d'une autre (boutons actif / passif ?) dans la barre des icônes (actuellement on ne sait pas lequel est actif et de plus ils ne sont pas nommés ...)

La commande « Edit style » appelle une boîte de dialogue qui permettrait de supprimer les boutons de configuration qui sont actuellement associés à chaque balise dans l'éditeur. La structure de cette boîte de dialogue pourrait être la suivante (voir aussi la discussion sur l'insertion automatique des balises) :



Dans le choix des couleurs, le bouton « Other » à droite de chaque menu (front et back) permettrait l'accès à un "color chooser" (j'imagine qu'il doit y en avoir un standard dans les bibliothèques JAVA ?) permettant à l'utilisateur de s'affranchir des couleurs prédéfinies dans le fichier "préférences". La taille des polices peut être sélectionnée à l'aide d'un simple menu pop-up contenant les valeurs classiques (7 8 9 10 11 12 13 14 16 18 20 22 24).

Remarque sur le format de sauvegarde des feuilles de style

Pour sauver les styles on pourrait avoir un format XML du genre :

```
<style name ="nom du style">
  <balise nom="une_balise">
     <CSS code="le codage CSS de la balise"/>
     <Completion action ="" insert="" begin="" end="" />
  </balise>

  ....

</style>
```

Où les attribut de la balise «Completion» correspondent aux options suivantes

- action = nothing|complete
- insert = tag|section|document|expression
- begin = "expression régulière"
- end = "expression régulière"

Comportement du Cut/Copy/Paste

Problèmes:

- *on ne peut pas copier un bloc »balisé » n'importe ou dans le texte*
- *il faut que le couper/copier agisse sur des balises*

A rédiger…

Organisation générale du menu

Le but de cette partie est d'indiquer le contenu exacte de chacun des items de la barre de menu de l'éditeur. Dans la mesure ou la gestion d'icône semble compliquer plus qu'autre chose le développement du logiciel on peut imaginer les supprimer toutes (à discuter) … Par ailleurs la barre d'outils ne doit contenir que des commandes souvent utilisées (à définir), des informations générale (numéro de ligne), et les styles disponibles …

Quelques points généraux sur le nommage de l'intitulé d'une commande de menu

- Une commande commence par une majuscule
- Lorsque la commande fait apparaître une BD elle se termine pas «□.□
- Les commandes sont regroupées en groupes homogènes séparés par un séparateur

FILE
New document

Load DTD …
Load text document …
Load annotated document …

Save annotated document …
Export XML document … (cette commande devrait disparaître si tout est XML)
Export HTML document …
Export RTF document …

Print …

Quit

EDIT
Cut
Copy
Paste
Select-all

Test DTD compatibility (validation soft …)
Validate DTD (validation complete …)

Preferences …

STYLE
New style …

Load style …
Save current style …

Delete current style
Edit styles …

Stratégie d'insertion des balises

Initialement le texte que l'on charge dans l'éditeur est dépourvu de balise, c'est l'annotateur qui les introduit une à une dans l'ordre qu'il souhaite. Il est néanmoins souhaitable d'effectuer un certain nombre de contrôles ou d'opérations automatiques afin de faciliter la tâche de l'utilisateur et lui éviter des erreurs.

Contrôles du balisage

Dans la version actuelle de l'éditeur la seule manière de vérifier que le balisage effectué par l'utilisateur est correct vis-à-vis de la DTD est d'utiliser « a posteriori » un validateur. Or certaines situations devraient être évitables « a priori ». Par exemple, si l'utilisateur effectue deux balisages successifs d'une partie de texte avec un recouvrement, on obtient aujourd'hui un code XML incohérent (indépendamment de la DTD) :

1) Tag A : « un exemple de phrase » : un exemple <A> de phrase </A>
2) Tag B : « un exemple de phrase » : <B> un exemple <A> de </B> phrase </A>

Plusieurs solutions sont possibles pour éviter/corriger une telle situation :

- soit en décalant les étiquettes précédemment existantes (ici <A>) de manière à faire disparaître le chevauchement des balises,
- soit en décalant une des étiquettes en conflit de manière à obtenir une inclusion stricte (A dans B ou l'inverse).
- soit en dupliquant le texte (ici le mot « de » apparaîtrait dans les deux zones),
- soit plus simplement en signalant une erreur

Le comportement à adopter pourrait être, au moins dans les deux premiers cas, partiellement automatisé en prenant en compte la structure de la DTD :

- Si A est plus spécifique[1] que B <B> un exemple <A> de phrase </A> </B>
- Si B est plus spécifique que A <A> <B> un exemple de phrase </A> </B>
- Si A et B sont sans relation <B> un exemple de </B> <A> phrase </A>

Dans ce dernier cas (A et B sans relation) si les zones sélectionnées étaient exactement les mêmes (mêmes positions de début et fin) on pourrait éventuellement supprimer le balisage A et le remplacer entièrement par celui de B.

Autre point, il serait très souhaitable que l'éditeur adopte plus largement lorsque cela à un sens *le comportement classique des TdT*. Ainsi, deux zones contiguës ayant le même balisage devraient être automatiquement fusionnées. Par exemple:

1) Tag A : « un exemple de phrase » : un exemple <A> de phrase </A>
2) Tag A : « un exemple de phrase » : un <A> exemple de phrase </A>
~~un <A> exemple </A><A> de phrase </A>~~

[1] Plus spécifique signifiant que dans la structure de la DTD la balise A peut apparaître dans une balise de type B a une profondeur quelconque.

De même lorsque la balise qui est appliquée sur la zone sélectionnée est la même que la balise que l'on trouvait antérieurement (et que cette balise ne peut être appliquée récursivement), on peut modifier simple ment la taille de la zone concernée.

1) Tag A : « un exemple de phrase » : un exemple <A> de phrase </A>
2) Tag A : « un exemple de phrase » : <A> un exemple de phrase </A>
~~<A> un <A> exemple de phrase </A></A>~~

Enfin, une autre manière de faciliter l'introduction des balises serait de n'activer[2] à tout instant que les balises qui sont « potentiellement » acceptables, c-à-d celles qui sont « plus spécifiques[3] » dans la DTD que la balise « courante » qui est explicitée par la position du curseur ou celle de la zone sélectionnée (en prenant en compte l'exception précédemment mentionnée ou les zones sélectionnées et balisées sont exactement les mêmes). Pour chaque balise, cette liste de dépendances peut être facilement calculée une fois pour toute lors de la lecture de la DTD …

Ces options sont évidemment à discuter plus amplement en fonction de la complexité de leur mise en place. Pour les implémenter il faut probablement disposer d'une API minimale du genre associée à une liste (ou arbre) triée des positions de balises :

GetGlobalBalise (selection) → Balise englobant immédiatement la sélection
GetBeginningBalise (selection) → Liste des balises commençant dans la zone sélection et se finissant à l'extérieure de celle-ci
GetEndingBalise (selection) → Liste des balises finissant dans la zone sélection et se commencant à l'extérieure de celle-ci
GetIncludedBalise (selection) → Liste des balises incluses dans la sélection
GetMostSpecific(balise) → Renvoie les balises plus spécifiques de la DTD

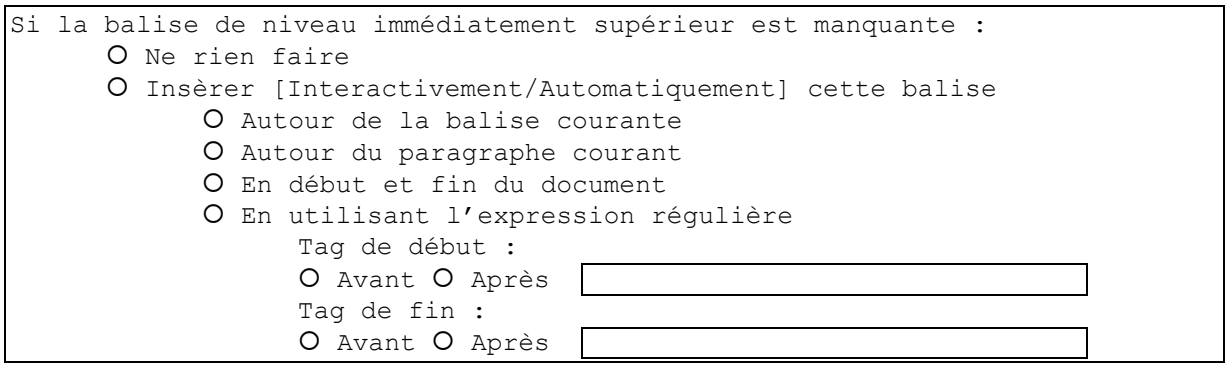
Il est clair que lorsque la zone sélectionnée est importante ces méthodes peuvent conduire à des temps de calcul importants. Une manière de limiter ce problème serait par exemple de borner la taille des listes renvoyées ce qui ne devrait pas créer de différence notable de fonctionnement du système du point de vue de l'utilisateur.

Insertion (semi)-automatique des balises

Dans la DTD que nous utilisons (mais ce n'est pas spécifique), le marquage d'un agent d'interaction, d'une cible, d'une interaction … sont assez simples à effectuer pour l'utilisateur (biologiste) puisqu'elles consistent à affecter un sens biologique aux éléments du texte. Par contre, l'introduction des balises de structuration plus générales (par exemple ABSTRACT, SENTENCE, …) est nettement moins intuitive et elles risquent d'être assez systématiquement oubliées ce qui conduira à devoir effectuer de nombreuses corrections. Par ailleurs, on constate que certaines balises sont sémantiquement « couplées » dans la DTD : par exemple, une balise d'agent d'interaction <A> est forcément englobée dans un fragment <AF>.

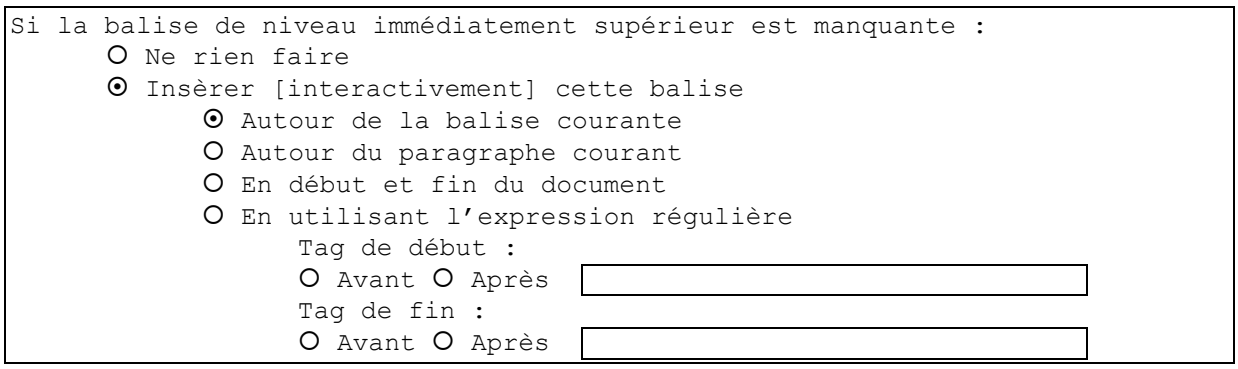
[2] Deux solution sont possibles : soit clairement de désactiver les balises non autorisées, soit de les tagger (icône) sans les interdir de manière particulière de manière à signaler qu'une incohérence existe entre les deux balisages.
[3] La mise en place d'une option complémentaire qui consisterait à n'activer que des balises « plus générales » que l'ensemble des balises contenue dans la zone sélectionnée est également envisageable mais risque d'être plus couteuse en temps.

Pour résoudre ces différents problèmes[4], il semble nécessaire de pouvoir associer aux feuilles de style des directives (optionelles) permettant d'insérer automatiquement les balises de niveau supérieur (englobantes) lorsqu'elles sont absentes dans le document. Or, cette information n'est « qu'en partie » déductible de la DTD, car si celle-ci permet de connaître les balises englobantes, elle ne permet pas de savoir OU elles doivent être introduite dans le document. Pour y parvenir voici les options que l'on pourrait associer avec la feuille de style associée à une balise.



On trouve deux séries d'options imbriquées (radio-boutons) dans la boîte de dialogue. Tout d'abord pour décider si l'absence de l'étiquette de niveau supérieur déclenche une action ou non, puis si ce n'est pas le cas pour décider où sera insérée (de manière entièrement automatique ou avec une validation explicite de l'utilisateur) la balise supérieure manquante. La dernière option permet d'effectuer une recherche des points d'insertion à l'aide d'une expression régulière (la recherche du point d'insertion de la balise ouvrante s'effectuant vers le début du document).

Ainsi, l'exemple ci-dessus est la configuration de la feuille de style associée à une balise <An> de manière à ce qu'on ajoute interactivement la balise <Afn> qui doit l'englober autour de la balise <An> introduite en premier.



Dès lors, grâce à un tel mécanisme si l'annotateur sélectionne la partie "centrale" de l'agent (balise <An>) d'une interaction avant d'avoir décrit la partie "large" (balise <Afn>) il serait possible d'ajouter automatiquement cette dernière dans le texte.

En pratique, cette gestion de l'insertion des balises manquantes doit être effectuée de manière récursive jusqu'à ce qu'on arrive à la racine de la DTD ou jusqu'à ce que l'on retombe sur une balise qui était effectivement prévue par la DTD. Ainsi si l'utilisateur charge un document et commence directement à introduire une balise <A1> dans une phrase quelconque, cela

[4] A court terme on peut également envisager d'ajouter automatiquement les balises les plus générale via un « préprocessing » externe des fichiers texte.

déclenchera les insertions suivantes en cascade (pour peu que les feuilles de styles aient été correctement configurées) :

- Ajout de la balise <AF1> autour de <A1>
- Ajout de la balise <INTERACTION> autour du paragraphe courant
- Ajout de la balise <SECTION> autour de <INTERACTION>
- Ajout de la balise <ANNOTATED-DOCUMENT> en début et fin de document

On pourra utiliser une boîte de dialogue de ce type pour modifier les paramètres graphiques et sémantiques associés à chacune de balises de a DTD. L'onglet Style contiendrait le « font chooser » précédemment décrit et l'onglet Completion les options de configuration de l'ajout automatique de styles