

## 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 6.1 de l'éditeur

### Historique

#### 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

#### 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 6.05

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

### Liste des remarques

Les différentes remarques sont étiquetées 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) sont marquées d'un trait dans la marge droite ...

#### 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 ...

Pensez à regarder les spécifications décrites dans la seconde partie de ce document

## Fonctions générales

**IMP** : il serait utile de pouvoir imprimer un texte «Balisés» en gardant le code de couleur. Il y a plusieurs solutions pour cela (notamment (A DISCUTER):

- Soit la classe JAVA utilisée éditer le texte supporte l'impression (simple)
- Soit le système est capable d'exporter une version HTML (avec les bonnes balises de couleurs ou mieux des feuilles de styles CSS2) du texte en cours d'édition (A VOIR).

**BUG** : Assez souvent (lors de l'abandon d'une tentative de chargement de fichier notamment) le curseur de la souris se transforme (montre, ou sablier selon les OS) en indiquant qu'il y a une tâche en cours. L'interface reste fonctionnelle, mais c'est troublant pour l'utilisateur. Est-ce un problème de mauvais rafraîchissement ou plus gênant un thread «Fantôme»

## Gestion des préférences de l'éditeur

**MOD** : Déplacer le menu préférence dans le menu Edit (à la fin)

**IMP / MOD** : Terminer l'implémentation de la boîte général "Préférence" permettant de sélectionner les choix par défaut pour le système (*voir les remarques sur le format de sauvegarde des fichiers annotés et les spécifications d'implémentation*)

**IMP / MOD** : Le format du fichier de préférence est à définir plus clairement. Il y a des attributs "frontcolor{R,G,B}" est ce que cela veut dire qu'un attribut "backcolor{R,G,B}" est également prévu ? Une syntaxe plus inspirée du HTML semblerait préférable (on peut par ailleurs considérer que la liste des couleurs de "fond" et "police" est la même)

<fontcolor = "code RGB en hexa" /> par exemple "#8b0000"

**BUG** : Lors du premier appel de la boîte de préférence la «check-box» indiquant l'affichage des balises de fin n'est pas cochée alors que ces balises sont bel et bien visibles ....

## Zone de texte (haut à gauche)

(**IMP** : **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.

**IMP** : Il serait bien de remplacer "l'inverse vidéo" par une sélection "normale" (celle qu'on utilise avec la souris) .... Pas prioritaire.

**HYP** : Prendre en compte la spécification de la manière dont devrait être effectuée la gestion des balises dans l'éditeur (*cf. seconde partie de ce document*) ...

(**HYP** : A SPECIFIER) : l'insertion de caractères indiquant le début et fin de balisage de manière à désambigüer l'étiquetage lorsque les zones sont recouvrantes.

## Zone XML (bas)

**IMP – BUG** : Lorsque l'on "scroll" sur le texte après avoir sélectionné une balise XML dans la fenêtre inférieure gauche il faut remettre le curseur d'insertion en début de la zone de texte correspondante dans la fenêtre supérieure gauche. Actuellement elle reste à l'emplacement où elle était avant la sélection et comme le curseur de texte disparaît, on ne sait plus où il est !

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

**BUG** : L'indentation des balises XML lorsque la balise de fin est invisible n'est pas correcte

(lorsque les balises de fin sont présentes c'est OK).

**BUG**: Lorsque l'on déplace le curseur d'insertion il faut montrer (sélectionner) le code XML correspondant à la balise courante dans le texte (je ne sais pas si c'est un problème nouveau lié à la gestion de l'apparition ou non de la balise de fin dans la zone XML ou une régression par rapport à la version précédente de l'éditeur)

**BUG** : Lorsque l'on applique une balise dans la zone de texte, le code XML correspondant n'est pas automatiquement sélectionné dans la zone XML. (Il faut explicitement recliquer dans la zone de texte pour mettre à jour la sélection).

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

### Zone édition d'attributs

**IMP** : 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 «[Un 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. «[Fonction 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. De même il FAUT sauver le nom (du fichier) de la DTD utilisée.

Le rechargement d'un fichier annoté conduit donc l'éditeur à charger (si ce n'est pas déjà le cas !) la DTD associée à ce fichier et ses feuilles de styles. Si la DTD n'est pas trouvée dans le dossier courant cela provoque un message d'erreur. Notons que d'autres informations de sauvegarde du contexte pourraient être sauvé dans le fichier comme le nom du style actif et la position du curseur dans la fenêtre de texte.

**BUG** : Dans certain cas, le fichier XML produit est vide (problème sous Linux mais pas OSX)

### Menu FILE

**IMP** : Ajouter une commande "New document" dont l'effet est de créer un document vide (on vide la mémoire) tout en laissant la DTD et les feuilles de styles intactes.

**BUG** : Pour la commande QUIT, il est raisonnable de faire apparaître une boîte de dialogue demandant si l'on souhaite (si besoin) sauvegarder le fichier en cours d'annotation ...

## Gestion des styles

**BUG** : Le chargement d'un nouveau fichier de style ne remet pas à jour l'affichage des zones de texte qui étaient déjà balisées dans le document courant.

**IMP** : Terminer l'implémentation partielle (au niveau de l'interface) qui a été effectuée des boutons de styles (commande "Save this style"). Voir les *spécifications du menu style* ...

**BUG** : Sous certains systèmes (Linux mais pas OSX) les boutons de modification des styles (fenêtre en bas à droite) n'apparaissent pas. Se reporter au document "INTERFACE-INSERTION" pour la description des modifications à apporter visant à supprimer ces boutons.

**BUG** : Dans les menus permettant de choisir les couleurs la couleur "courante" n'apparaît pas dans le menu (le menu reste toujours blanc quelle que soit la couleur actuelle).

## 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

## Spécifications de l'interface d'annotation

### 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

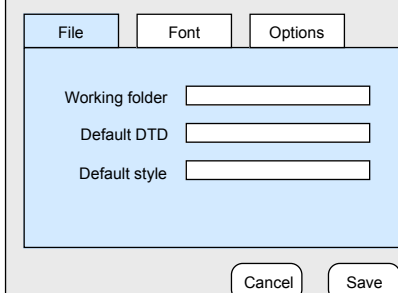
Concernant les couleurs il n'est pas prioritaire de permettre une édition directe et on se contentera pour en modifier la liste d'éditer le fichier de préférence "xmljava.conf".

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 ...



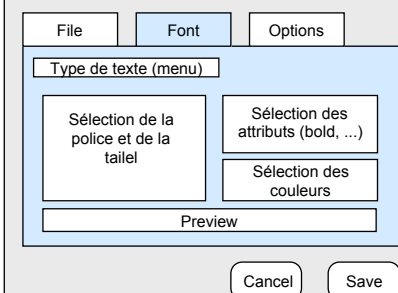
#### \* Pour l'onglet Font :

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

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

NOTE : L'implémentation actuelle de "Annotated text" conduit à modifier l'ensemble des styles associés aux balises ce qui est pratique mais dangereux (risque d'écrasement des styles déjà définis). On va appeler cela une "feature" du système :-)

Au niveau de la boîte de dialogue le plus simple est d'introduire un menu «Pop-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 « **Textuel** » par exemple celui des CSS qui paraît le plus intéressant.
- Delete current style : Supprime le style courant (mais pas le fichier), ☐ n'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 :

Font chooser

Font & Size

Police Taille

Liste des polices


Style


☐ Bold ☐ Italic

☐ Underline ☐ Strikout

☐ Superscript ☐ Underscript

Color

Front  Other

Back  Other

Preview

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).

## 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<sup>1</sup> 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>~~

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 simplement la taille de la zone concernée.

---

<sup>1</sup> Plus spécifique signifiant que dans la structure de la DTD la balise A peut apparaître dans une balise de type B à une profondeur quelconque.



- 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<sup>2</sup> à tout instant que les balises qui sont « potentiellement » acceptables, c-à-d celles qui sont « plus spécifiques<sup>3</sup> » 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 commençant 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>.

Pour résoudre ces différents problèmes<sup>4</sup>, il semble nécessaire de pouvoir associer aux feuilles de style des directives (optionelles) permettant d'insérer automatiquement les balises de

<sup>2</sup> Deux solution sont possibles : soit clairement de désactiver les balises non autorisées, soit de les tagger (icône) sans les interdire de manière particulière de manière à signaler qu'une incohérence existe entre les deux balisages.

<sup>3</sup> 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.

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



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.

Si la balise de niveau immédiatement supérieur est manquante :		
<input type="radio"/>	Ne rien faire	
<input type="radio"/>	Insérer [Interactivement/Automatiquement] cette balise	
<input type="radio"/>	Autour de la balise courante	
<input type="radio"/>	Autour du paragraphe courant	
<input type="radio"/>	En début et fin du document	
<input type="radio"/>	En utilisant l'expression régulière	
	Tag de début :	
<input type="radio"/>	Avant <input type="radio"/> Après	<input type="text"/>
	Tag de fin :	
<input type="radio"/>	Avant <input type="radio"/> Après	<input type="text"/>

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.

Si la balise de niveau immédiatement supérieur est manquante :		
<input type="radio"/>	Ne rien faire	
<input checked="" type="radio"/>	Insérer [interactivement] cette balise	
<input checked="" type="radio"/>	Autour de la balise courante	
<input type="radio"/>	Autour du paragraphe courant	
<input type="radio"/>	En début et fin du document	
<input type="radio"/>	En utilisant l'expression régulière	
	Tag de début :	
<input type="radio"/>	Avant <input type="radio"/> Après	<input type="text"/>
	Tag de fin :	
<input type="radio"/>	Avant <input type="radio"/> Après	<input type="text"/>

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 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

