

CADIXE (Projet Caderige : <http://caderige.imag.fr/>)

Manuel d’utilisation (Editeur version 2.0a4 du 1/03/05)

Contact : Gilles.Bisson@imag.fr, Hassiba.Zidelkheir@imag.fr

Objectifs de l'éditeur

L’éditeur CADIXE permet d’annoter interactivement un document textuel à l’aide de balises XML décrites dans une DTD externe. Voici quelques sites présentant ces technologies :

- Présentation du XML : <http://www.brics.dk/~amoeller/XML/overview.html>
- Ecriture d’une DTD : http://www.w3schools.com/dtd/dtd_intro.asp

Contrairement à ce qui est couramment proposé dans ce type d’éditeur, l’utilisateur n’a pas à structurer directement le texte sous la forme d’un arbre XML. Il peut introduire les balises une à une dans l’ordre où il le souhaite en « surlignant » la partie du texte qui l’intéresse et en y appliquant la balise de son choix parmi celles disponibles. On reste ainsi dans une optique voisine de celle que l’on trouve avec un traitement de texte. Les zones de textes balisés sont visualisées à l’aide d’une combinaison de polices et couleurs qui sont décrites dans des « feuilles de styles » au format CSS librement configurables par l’utilisateur.

Configuration minimale

Le logiciel étant écrite en JAVA, il est exécutable sur un large nombre de plateformes : Linux, MacOS X et Windows. La machine virtuelle doit être récente : version $\geq 1.4.2$ et l’on doit disposer d’une quantité de mémoire raisonnable (≥ 256 Mo).

Installation de l'application

L’application distribuée est nommée CADIXE.jar (il ne faut pas modifier ce nom !) et **il n’est pas nécessaire** de la « décompacter ». Lors de l’exécution, le logiciel construit automatiquement (si elle n’existe pas déjà) la hiérarchie des fichiers dont il a besoin pour travailler dans le dossier courant. Elle prend la forme suivante :

- CADIXE.jar : l’exécutable en Java
- /resources : les ressources (icônes, labels ...) de l’application
- /cadixe-preferences : les configurations de l’éditeur
 - cadixe.properties : fichier des préférences par défaut
- /formats
 - /models : modèles de documents
 - /dtd : fichiers des DTDs
 - /styles : feuilles de styles au format CSS
- /documents : le dossier contenant les documents utilisateurs
- /scripts : le dossier des scripts (non actif dans cette version)

Les emplacements et les noms de dossiers qui sont notés en vert sont modifiables via le paramétrage de l’application (cf. partie « gestion des préférences ») ou la ligne de commande. Il est ainsi possible de configurer précisément l’éditeur de manière à ce que, par exemple, les DTDs et Styles soient globaux à tous les utilisateurs et que les documents et les préférences restent locaux. Pour récupérer les informations issues d’une version précédente, il suffit de les remettre dans les dossiers correspondants (dtd, styles, documents, ...).

Lancement de l'application

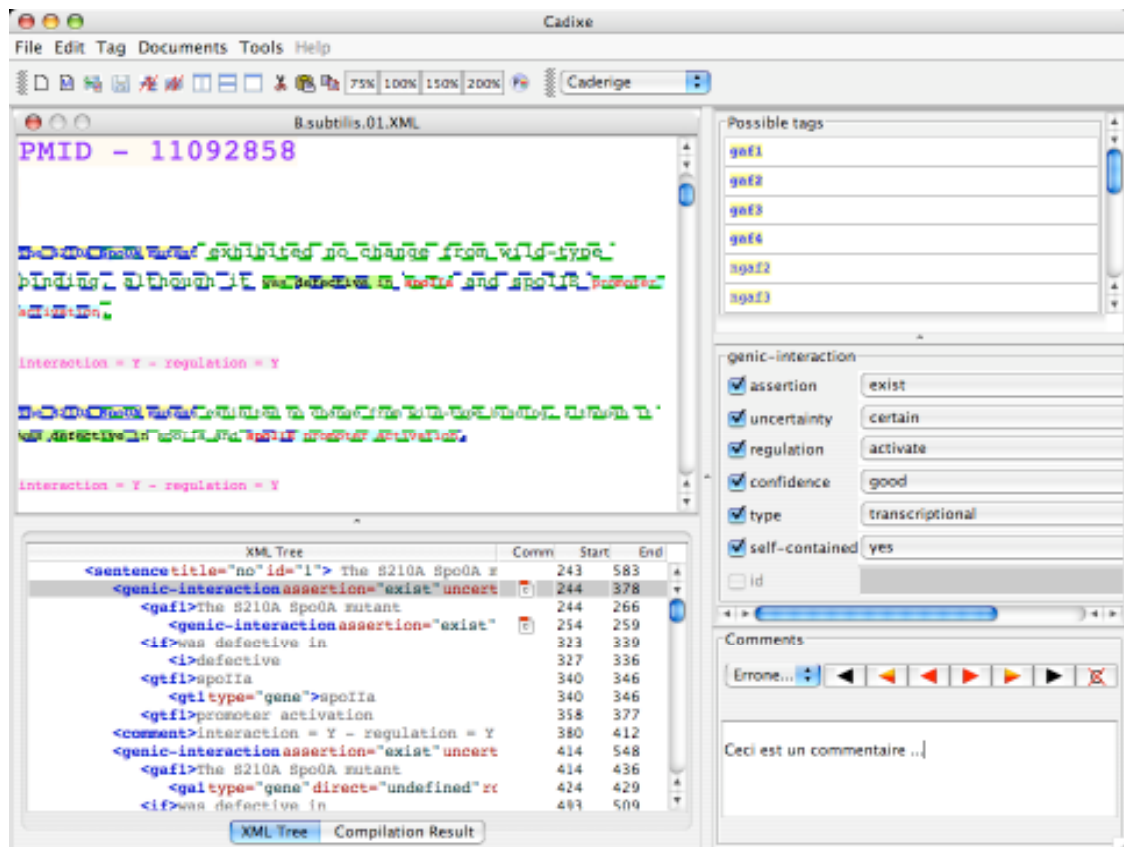
Le lancement de l'éditeur sous UNIX (ou Linux) s'effectue à partir d'un terminal (Xterm). Il faut se placer au niveau de l'application puis exécuter la ligne de commande suivante :

```
java -jar CADIXE.jar
```

Si l'application n'est pas au niveau du dossier courant (pwd), on peut aussi la lancer en indiquant simplement le chemin d'accès. Il n'est plus nécessaire de mettre un argument supplémentaire `-root` comme dans les versions antérieures V1.xx.

```
java -jar /Volumes/Memnos/editeur/version 2/CADIXE.jar
```

Sous d'autres systèmes (OS X notamment, Windows, ...) il est également possible, et plus simple, de lancer l'application en double-cliquant sur l'exécutable `CADIXE.jar`. Après quelques instants, la fenêtre principale de l'éditeur apparaît ... L'application étant écrite avec Swing, l'aspect de la fenêtre s'adapte en fonction du système hôte (la copie ci-dessous est faite sur MacOS X) et selon les paramètres de lancement qui ont été utilisés.



Les paramètres de la ligne de commande

Lors du lancement de l’éditeur, on peut spécifier dans la ligne de commande les options suivantes permettant de modifier le comportement par défaut de l’éditeur :

-userhome path : par défaut l’éditeur stocke ses dossiers de travail à l’endroit où se trouve l’application, cet argument permet d’indiquer un autre chemin d’accès pour le fichier de préférence et ainsi, en fonction des informations qu’il contient (cf. « gestion des préférences »), amener le système à gérer ses fichiers à un autre emplacement. Par exemple, sous Unix, on pourra mettre dans le script de lancement de l’éditeur un chemin contenant la variable \$HOME si l’on veut que la gestion des données soit effectuée par rapport au compte de l’utilisateur.

Lors de l’exécution, les dossiers qui sont nécessaires à l’application (Cadixe-preferences, Formats, ...) et qui n’existent pas encore (c’est à priori le cas lors du premier appel) sont automatiquement copiés à partir du dossier de l’application. Il est ainsi facile de « distribuer » une même configuration initiale aux utilisateurs qu’ils pourront ensuite personnaliser. Voici un exemple d’appel (attention, dans ce cas le dossier Cadixe doit avoir déjà été créé) :

```
java -jar /CADIXE.jar -userhome $HOME/Cadixe
```

-prefs name : indique le fichier de configuration (qui contient les préférences) qui est à charger dans le dossier « Cadixe-preferences » du dossier de travail courant. Par défaut le fichier qui est chargé est « cadixe.properties ». On peut avoir donc différentes configurations (par exemple correspondant à des chemins différents sur les dossiers de travail) et utiliser l’une ou l’autre lors du lancement de l’éditeur.

-load path : charge le fichier qui est indiqué automatiquement lors du lancement de l’éditeur. Le fichier peut-être un fichier texte ou un fichier annoté (au format XML). Lorsqu’il s’agit d’un fichier texte, il est chargé en utilisant la DTD et les styles qui sont définis dans le modèle par défaut (cf. « gestion des préférences »).

-name string : Ouvre un nouveau document (qui est construit à partir du modèle par défaut) avec comme nom celui qui est passé en paramètre.

-model path : Indique le modèle par défaut qui est à utiliser, à la place du modèle par défaut défini dans les préférences, lorsque l’on charge un document textuel (–load) ou que l’on crée un document (–name).

Description de la version courante

La version 2.0a4 est la quatrième version « alpha » de l’éditeur. Voici les nouveautés.

Nouveautés de la version 2.0a4 par rapport à la version 2.0a3

- Implémentation des balises de type « EMPTY » qui sont des balises comme <Tag/> qui n’englobent pas de texte. Elle sont repérées dans le document à l’aide d’icônes que l’utilisateur peut spécifier dans le fichier de style.
- Implémentation des menus permettant la création de modèles à partir du document courant et le chargement des feuilles de styles.
- Ajout de la ligne : `<?xml-stylesheet type="text/css" href="xxx.css"?>` dans l’entête des fichiers ce qui permet d’avoir une visualisation du fichier annoté par l’intermédiaire d’un navigateur WEB via la feuille de style courante « xxx.css ».

Fonctionnalités manquantes de la 2.0a4

Il manque encore certaines fonctions dans l’éditeur. Elles seront implémentées dans les mois qui viennent, merci de nous indiquer vos priorités !

- La gestion des commandes Undo/Redo.
- Accès à une documentation en ligne à partir de l’éditeur
- Des commandes d’édition permettant de rallonger/raccourcir une balise.
- L’appel à des scripts externes pour faire des annotations automatiques.
- L’exportation au format HTML des documents annotés.
- La possibilité de piloter l’éditeur via des commandes http (load, close, ...).
- La complétion automatique des balises comme dans les versions 1.xx

Bugs connus de la 2.0a4

- Lorsque la palette des commentaires est incrustée à droite de l’éditeur, la visualisation de la prise en compte des modifications d’états dans la zone XML ne se fait qu’après le click souris suivant dans l’interface.
- Lorsque la fenêtre de l’éditeur est petit et que toutes les palettes sont affichées simultanément, certaines parties de la zone des commentaires et de la recherche sont « cachées » et non accessibles.
- Les opérations de suppression de balises sont lentes sur les documents longs.

Historique des versions

Nouveautés de la version 2.0a3 par rapport à la version 2.0a2







- Implémentation de la validation du code XML. Contrairement à ce qui se faisait dans les versions 1.xx, la validation permet maintenant de retrouver en une seule passe l’ensemble des balises manquantes, ce qui accélère la correction. Lorsque l’on sélectionne une erreur l’éditeur sélectionne automatiquement la balise incriminée, et les messages d’erreurs ont été revus de manière à améliorer la compréhensibilité.
- Correction de plusieurs bugs de la version 2.0a2
 - Correction des fonctions de recherche et activation du Replace
 - Correction de la gestion des boîtes de remplissage des attributs
 - Correction d’erreurs multiples dans le copier/coller,
 - Correction de la fonction « delete » (suppression par la touche « Suppr »).
 - Correction de l’affichage lorsque l’on applique une balise englobante
 - Correction et extension du présent manuel ☺

Nouveautés de la version 2.0a2 par rapport à la version 2.0a1

- Implémentation des fonctions de recherche. Il est ainsi possible de faire des recherches et des remplacements sur le texte du document annoté, mais également (nouveau **v20**) de faire des recherches dans les annotations pour retrouver une balise, un commentaire ou une valeur d’attribut particulière.
- Implémentation de la gestion des commentaires sur les balises (nouveau **v20**) qui permet à l’utilisateur d’associer à toutes les balises du document des informations en texte libre expliquant ses choix d’annotation. Il est également possible d’affecter un « état » à chaque balise parmi : {validated, unvalidated, erroneous}.
- Implémentation des « styles paramétriques » (nouveau **v20**). Ils permettent d’associer plusieurs feuilles de styles à une même balise, le choix du style à afficher étant fonction de la balise ET de la valeur d’un ou de plusieurs attributs. Par exemple, il n’est plus nécessaire d’avoir des balises de type <Agent1>, <Agent2>, ... mais une seule balise <Agent> dont la feuille de style est indexée sur un attribut « Numéro_agent ». L’utilisation de cette possibilité demande toutefois une traduction des documents déjà annotés (Cf. styles paramétriques).
- Implémentation de la boîte de préférences et des paramètres associées. Il est donc possible de paramétrer l’application comme dans la version 1.16 (notamment le fait de « forcer » le renseignement des attributs lors de l’introduction d’une balise). De nouvelles possibilités de paramétrage sont en outre accessibles.
- Simplification de la récupération des documents annotés avec la version 1.xx de l’éditeur. Il suffit maintenant de mettre les DTDs et les feuilles de styles (au nouveau format) dans les dossiers correspondants de la version 2.0a2.
- Amélioration des possibilités de configuration de l’éditeur : il est ainsi possible de dire explicitement quels sont les fichiers (DTD, configuration, ...) qui sont partagés par tous les utilisateurs et ceux qui sont gérés localement.
- Implémentation d’une boîte « document information » permettant de conserver des méta-informations sur le document (nom des annotateurs, historique, ...).
- La gestion de nouveaux paramètres de lancement : -load, -name, -model, -userhome.

- Sauvegarde des préférences liées au document (zoom, spacing, position du curseur) directement dans le document annoté sous la forme de commentaires XML. Lorsque l'on recharge un document, on retrouve donc les conditions de travail précédentes.
- Modification (simplification) de la barre des menus
- Corrections de plusieurs bugs présents dans la version 2.0a1

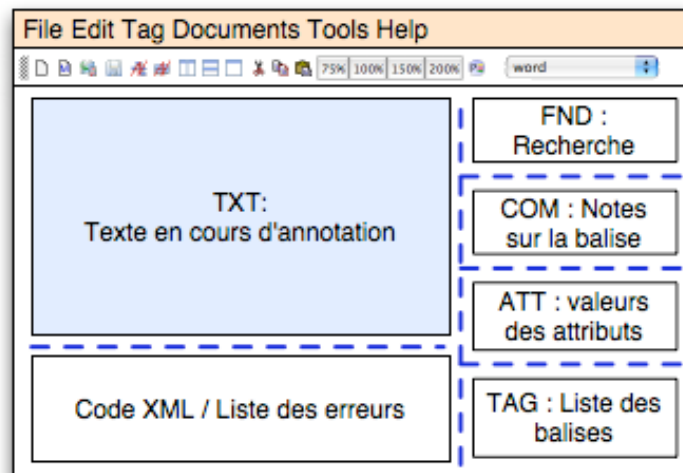
Nouveautés de la version 2.0a1 par rapport à la version 1.17

- Nouvelle implémentation
 - L'application a été *intégralement réécrite* ce qui a permis d'avoir un gain majeur en termes de vitesse, de stabilité et ce qui facilite énormément la maintenance et l'évolution de l'application. En outre, les formats de fichiers ont été revus de manière à être plus simples. Par exemple, les feuilles de styles sont au format CSS (à 98% certains attributs étant spécifiques à l'éditeur).
- Gestion multi-documents (nouveau )
 - L'éditeur permet maintenant de charger simultanément plusieurs documents annotés en mémoire, chaque document ayant sa propre DTD et ses propres styles, et d'en visualiser plusieurs simultanément ce qui facilite la comparaison des annotations.
- Copier/coller de textes et de balises (nouveau )
 - Le copier/coller prenant en compte les annotations est (enfin) implémenté, il permet également d'importer ou d'exporter du code XML. Par exemple si on copie une partie annotée et qu'on la colle dans un éditeur de texte (Emacs,...) on voit apparaître le code XML correspondant. L'inverse étant également possible on peut donc importer du code écrit à la main, ou « à terme » qui sera généré par un script externe.
- Meilleur contrôle de l'introduction des balises (nouveau )
 - Dans la version précédente, on ne pouvait insérer que des balises plus spécifiques que la balise courante, dans la V2.0 l'éditeur contrôle en outre que la balise que l'on applique sur un bloc est bien plus générale que toutes les balises incluses. Ainsi, n'est plus possible d'appliquer plusieurs fois une balise sur une même zone de texte.
- Nouvelles annotations graphiques (nouveau )
 - En plus de jouer sur les couleurs et les polices dans les feuilles de styles, il est maintenant possible de dessiner des rectangles autour des annotations ce qui enrichit la palette visuelle (un exemple est fourni avec la distribution).
- Affichage personnalisable (nouveau )
 - Pour chacun des documents en mémoire, l'utilisateur peut sélectionner le coefficient de zoom de l'affichage qu'il souhaite employer : 50%, 75%, 100%, 150% ... Il n'est donc plus nécessaire de modifier les styles pour améliorer la lisibilité en outre l'utilisateur peut modifier la taille de l'interligne.
- Personnalisation des configurations (nouveau )
 - Via les paramètres passés dans la ligne de commande lors du lancement de l'éditeur, il est possible de régler le comportement de l'éditeur de manière à décider quelles ressources (DTDs, Styles,, ...) sont locales ou globales aux utilisateurs. Cela facilite l'installation de CADIXE dans des environnements multi-utilisateurs. Par ailleurs l'éditeur "auto-installe" les dossiers dont il a besoin lors de la première exécution.

Organisation générale de l'interface

Outre la barre de menu et la ligne d'outils contenant les commandes et les indicateurs usuels, la fenêtre de CADIXE est organisée en plusieurs zones qui peuvent être redimensionnées (ou masquées) en faisant glisser les barres de séparations :

- TXT : Le (ou les) document(s) en cours d'annotation
- XML : Les étiquettes XML qui ont été introduites dans le texte
- FND : Outil de recherche dans le document et les balises
- COM : Les commentaires (ou notes) associés à la balise courante
- ATT : La valeur des attributs de la balise courante
- TAG : La liste des balises utilisables dans le contexte courant



L'affichage de la zone d'outils (toolbar) qui est située sous la barre des menus ainsi que celui des zones ATT, TAG, FND et COM peut-être activé/désactivé à l'aide des commandes dans le menu « TOOLS » de la barre de Menus. Les positions relatives des zones d'éditations à droite sont modifiables (cf partie « préférences »).

Principes élémentaires de l'annotation d'un texte

Le principe d'utilisation de l'éditeur repose sur deux étapes :

- 1) la sélection à l'aide de la souris de la partie du document (mot, paragraphe, ...) que l'on souhaite annoter,
- 2) sur l'application d'une balise à cette partie. Pour se faire, il suffit de cliquer sur l'étiquette voulue dans la liste TAG des balises. La zone ainsi distinguée apparaît alors dans le « format d'affichage » (que l'on appelle un « style ») qui est associé à la balise et une nouvelle balise apparaît dans la zone XML et la zone ATT,
- 3) sur la sélection de la valeur des attributs (éventuels) de la balise dans la zone ATT.

A un instant donné, en fonction de la position du curseur, seules les balises de la DTD qui sont *plus spécifiques* que la balise courante peuvent être insérées. De manière symétrique lorsque l'utilisateur sélectionne toute une zone du document, seules les balises qui sont *plus générales* que chacune des balises apparaissant dans cette zone peuvent être insérées.

Selon la DTD utilisée, chaque balise peut avoir un certain nombre d'attributs qui servent à préciser sa signification et qui apparaissent dans la partie ATT de l'éditeur. Leurs valeurs

peuvent être modifiées par l'utilisateur : lorsqu'elles appartiennent à un type énuméré (liste), les valeurs possibles sont sélectionnables à partir d'un menu, dans le cas contraire une simple zone d'édition permet de saisir le texte associé à l'attribut.

A tout instant la position du curseur détermine l'emplacement de la balise courante, dès lors pour examiner une annotation il suffit de placer le curseur dans celle-ci. Le nom et la liste des attributs de la balise courante s'affichent dans la partie ATT de l'éditeur.

La zone inférieure gauche de l'éditeur affiche les balises XML insérées dans le texte ainsi que la position de début et fin de chaque balise (exprimée en nombre de caractères). Elle permet de contrôler si l'annotation en cours est satisfaisante (notamment en ce qui concerne l'imbrication des balises). La navigation entre les deux fenêtres TXT et XML est « synchronisée », en d'autres termes, lorsqu'on déplace le curseur dans l'une de ces zones l'autre est automatiquement décalée de manière à visualiser la même information.

Modification d'un texte et de son balisage

Comme dans tout éditeur, l'utilisateur est libre d'insérer, copier/coller, supprimer du texte dans le document. Si la zone appartient à une balise, ses positions de début et fin sont automatiquement mises à jour, toutefois si l'ensemble des caractères appartenant à cette zone sont supprimés, la balise est automatiquement supprimée de l'arbre XML.

Pour supprimer complètement une annotation il suffit de placer le curseur dans celle-ci puis de cliquer sur (ou de choisir dans le menu « Tag ») la commande "Delete current tag" pour effacer la balise courante ou sur "Delete tag group" pour supprimer *l'ensemble des balises* de cette zone c'est à dire l'étiquette courante et toutes celles qu'elle contenait éventuellement (ce qui correspond au sous-arbre XML). Dans ce dernier cas, une boîte de dialogue demande confirmation de la commande afin d'éviter les effacements accidentels.

Il peut arriver que plusieurs balises soient appliquées sur une même zone de texte (même position de début et de fin). Dans ce cas pour indiquer à l'éditeur le niveau sur lequel on veut travailler, il suffit de cliquer dans la zone XML sur la balise concernée.

Les balises « EMPTY »

Les balises EMPTY sont des balises particulières de la DTD qui ne servent pas à annoter directement un fragment de texte mais à placer une information ponctuelle dans le document. Lorsque l'on insère une telle balise une icône est insérée dans le document afin de la matérialiser. Le nom de l'icône à introduire est donnée dans la feuille de style de la balise (cf « Gestion des feuilles de styles »). Voici un exemple d'affichage avec ce type de balise :

the region from dnaA to abrB of the **Bacillus subtilis**  and ...

Contrairement aux autres balises l'insertion d'une telle balise peut s'effectuer à l'endroit du curseur sans qu'il soit nécessaire de sélectionner une zone de texte. Toutefois lorsqu'une zone est sélectionnée, les balises EMPTY sont insérées en fin de cette zone.

Gestion des commentaires

Pour chacune des balises qui sont insérées dans le texte, l'utilisateur peut associer un commentaire lui permettant d'expliquer l'annotation qu'il a effectué et éventuellement ses doutes ou interrogations sur cette annotation. Cette possibilité est à la fois intéressante dans

le cadre d’un travail qui peut se dérouler sur plusieurs mois et/ou qui est effectué conjointement par différents experts. Les commentaires sont organisés en deux parties :

- Une partie en texte libre.
- Un « état » qui prend sa valeur parmi : {validated, unvalidated, erroneous}. La signification de l’état est purement informelle et permet juste d’indiquer le degré de certitude que l’on possède sur l’annotation.

Les balises possédant un commentaire possède dans la zone XML de l’éditeur une icône spécifique dans la colonne « Comment » dont la couleur est représentative de l’état. Il faut noter que ces commentaires sont stockés dans le document annoté (XML) dans des champs commentaires « <-- ... --> » (au sens XML du terme) qui sont « interprétés » par CADIXE. Ces ajouts ne changent donc pas la conformité du document avec la DTD. Pour ajouter (ou modifier) un commentaires dans la balise courante, il y a trois possibilités :

- Appeler la commande « Add comment » dans le menu « tag »
- Double cliquer dans la colonne « comment » dans la zone XML
- Activer l’affichage de la zone de commentaire dans le menu « Tools » et saisir le commentaire dans la zone qui est dans la partie droite de l’éditeur. Dans ce dernier cas, la commande « Add comment » est automatiquement désactivée.

Lorsque la boîte de dialogue ou l’outil permettant la saisie des commentaires est affichée, l’utilisateur peut se déplacer de commentaires en commentaires grâce aux 6 boutons qui sont pacés au-dessus du champ de saisie textuel. Les boutons « noirs » permettent de se déplacer sur le commentaire précédent/suivant, les bicolores « rouge/orange » vont sur le commentaire précédent/suivant de type « unvalidated » ou « erroneous » et les « rouges » permettent de se déplacer entre les commentaires de type « erroneous » uniquement.

Enfin le bouton « reset » (le « C » barré à droite) supprime le commentaire courant c’est à dire qu’il remet l’état à « validated » et qu’il vide le champ en texte libre.

Visualisation des documents avec un navigateur WEB

Il est possible de visualiser le contenu d’un document annoté XML à l’aide d’un navigateur Web « standard » pour peut qu’il soit capable de prendre en compte des feuilles de styles CSS (c’est le cas de FireFox, Safari, ...). En effet, lors de la sauvegarde, CADIXE ajoute automatiquement dans l’entête du fichier, la ligne ci-dessous :

```
<?xml-stylesheet type="text/css" href="XXX.CSS"?>
```

Le nom XXX.CSS représente celui de la feuille de style courante. Dès lors, en copiant cette feuille de style dans le même dossier que le document, on peut charger et visualiser les balises sous leur forme graphique dans un navigateur Web.

Attention : comme il s’agit de XML, pour qu’il y ait des retours à la ligne entre certaines balises, on doit d’ajouter dans les styles de ces balises l’attribut « display : block; ». De plus pour mettre en forme l’affichage on peut également jouer sur l’ajout de directives telles : « padding : numberpx; » et « vertical-align : numberpx; ». L’ajout de ces attributs dans les styles n’est pas une gêne pour CADIXE car il ne les interprète pas.

Gestion des fichiers (Menu FILE)

Ce menu contient les commandes de chargement de fichiers. L’éditeur CADIXE est capable de travailler sur plusieurs documents en mémoire utilisant des DTD et des styles différents.

- **New Document** : l’éditeur construit un nouveau document en prenant comme modèle le fichier modèle par défaut qui est appelé pour l’instant « default.model » et qui se trouve dans le dossier « /formats/models/ ». Un modèle est fichier qui contient les noms de la DTD et des feuilles de styles qui sont à charger dans le nouveau document. On peut modifier ces noms en modifiant le modèle dans un éditeur de texte (les versions ultérieures permettront d’effectuer cela directement de l’interface).
- **New Document from Model** : l’éditeur construit un nouveau document en laissant l’utilisateur choisir le modèle qu’il veut utiliser. Le chemin par défaut qui est utilisé pour trouver les modèles est décrit dans les préférences (cf description)
- **New document from DTD ...** : l’éditeur construit un nouveau document à partir de la DTD qui est sélectionnée. Il charge aussi la feuille de style par défaut associé (elle porte le même nom) en la créant si elle n’existe pas déjà dans le dossier des styles.

- **Load Document** : charge un document qui peut être annoté ou non. Si le document un texte non annoté la balise « racine » de la DTD est automatiquement insérée. Lorsque le document est annoté, l’éditeur va automatiquement chercher dans le dossier contenant les DTDs, celle qui avait été utilisée dans le document annoté (il n’utilise que le nom, pas le chemin). Aussi, pour assurer la (re)lecture des documents annotés avec la V1.0, il suffit de mettre les DTD utilisées par ces documents dans le dossier des DTDs (par défaut, il est situé dans « ./formats/dtd »).

Par ailleurs, lorsqu’il charge une version antérieure le système charge automatiquement une feuille de style ayant le même nom que la DTD associée au document sinon il la crée. Par exemple, si le document pointait sur une DTD « caderige.dtd » il va automatiquement charger (ou créer si besoin) une feuille de styles nommée « caderige.css » dans le dossier des styles (qui est par défaut initialisé à « /formats/styles »). Pour assurer l’utilisation des feuilles précédentes, il est par contre nécessaire de convertir manuellement l’ancien format vers le nouveau (ce qui demande 5 minutes en utilisant la fonction « Find & Replace » d’un éditeur de texte).

- **Load recent ▶** : liste des derniers fichiers qui ont été chargés dans l’éditeur.
- **Insert Text ...** : insère un fichier de texte à la position du curseur.
- **Save Document** : Sauve le fichier annoté. Contrairement à la version précédente, il n’y a plus de fichier de style associé. Le nom des styles à charger est directement inclus dans les fichiers XML ce qui est beaucoup plus simple.
- **Save As ...** : Sauve le document en demandant un nom de fichier à l’utilisateur
- **Save Document as Model** : l’éditeur crée un nouveau modèle à partir du document courant. Il utilise comme nom de fichier celui de la DTD courante. Un modèle est un fichier qui contient les noms d’une DTD et d’une ou plusieurs feuilles de styles et qui peut être utilisé comme « patron » pour créer un nouveau document dans Cadixe (Cf la commande « new document »).
- **Quit ...** : Quitte l’éditeur après confirmation.

Edition du texte (Menu EDIT)

Dans cette version de l’éditeur, le copier/coller fonctionne maintenant sur les zones annotées. Le texte annoté est copié dans le « presse-papier » sous la forme d’une chaîne de caractères contenant les balises XML avec leurs attributs, on peut donc très facilement échanger des annotations entre CADIXE et d’autres logiciels, par exemple des éditeurs de textes.

Attention : lorsque l’on « importe » d’une application extérieure une zone balisée via le « presse-papier », CADIXE ne vérifie pas la validité du balisage (au sens où il n’y a pas de validation qui est effectuée vis à vis de la DTD). Dès lors l’insertion d’une telle information peut entraîner la création d’un arbre XML incohérent.

- Cut : coupe la partie sélectionnée et les annotations éventuelles.
- Copy : copie la partie sélectionnée et les annotations éventuelles.
- Paste : colle le contenu du « presse-papier ». On ne peut logiquement coller le contenu que si le point d’insertion correspond à une balise plus générale ou identique (dans ce cas, on fusionne les zones) à celle incluse dans le presse-papier.
- Paste as text : colle le texte « brut » sans les balises ... Le texte collé est associé à la balise courante comme si les caractères avaient été entrés au clavier.
- Find ... : recherche une chaîne de caractères dans le document annoté. La recherche peut être effectuée à l’aide d’une expression régulière (REGEX) lorsque la « check-box » correspondante est activée. Ces deux sites expliquent cette syntaxe :
 - <http://www.regular-expressions.info/reference.html>
 - <http://java.sun.com/docs/books/tutorial/extra/regex/index.html>

Voici quelques directives classiques utilisées dans les expressions régulières :

.	: le point indique un caractère quelconque dans la recherche.
^ et &	: correspond respectivement au début et à la fin d’une ligne.
\d, \w, \s	: désignent respectivement un chiffre, un caractère de mot, un espace.
[]	: indique une liste de caractères (ex : [0-9] ; [a-z] ; [abcdef0123] ...).
?	: séquence de caractères répétée 0 ou 1 fois.
+	: séquence de caractères répétée 0 ou plusieurs fois.
*	: séquence de caractères répétée 1 ou plusieurs fois.
	: alternative (si l’on cherche les mots « le » ou « la » : « lella »).
{num}	: séquence de caractères répétée <i>num</i> fois exactement.
{min,max}	: séquence de caractères répétée entre <i>min</i> et <i>max</i> fois.
()	: pour délimiter des séquences (ex : « [0-9\s(protein gene)+ »).

Par ailleurs, les 4 boutons qui sont placés en bas de la boîte : « First », « Previous », « Next » et « Last » permettent de naviguer d’une occurrence à l’autre.

- Replace ... : permet de rechercher une chaîne de caractères et de la remplacer par une autre. Attention pour des raisons de cohérence le remplacement n’est possible que si la chaîne de caractères est *entièrement* incluse dans une ou plusieurs balises. Dans le cas contraire, le remplacement n’est pas possible car l’éditeur n’a aucune manière de savoir sur quelle(s) partie(s) de la chaîne il doit réappliquer les balises.

- Search Tag ... : cette commande permet de rechercher des balises dans le texte en se basant sur leur nom, le commentaire et l’état qui leur est associé et la valeur des attributs (la chaîne vide correspond à ne mettre aucune contrainte sur cette valeur). Par exemple, on peut chercher toutes les balises de type <Abstract> qui ne sont pas validées (Unvalidated) et qui contiennent un attribut ID manquant.
- Next Find/Search : réapplique la dernière commande de recherche effectuée.
- Préférences : Accès à la boîte de préférences permettant de configurer le logiciel, les options disponibles sont décrites dans la suite de ce document.

Gestion des balises (Menu TAG)

- Delete current Tag: Suppression de la balise courante (désignée par la position du curseur). Cette commande ne modifie pas les balises qui seraient incluses dans celle-ci. Par exemple si l’on supprime la balise ayant la feuille de style rouge :

Avant : `contrast, both | the E213G and E213A Spo0A variants showed`

Après : `contrast, both the E213G and E213A Spo0A variants showed`

Notons qu’il est impossible de supprimer la balise « racine » de la DTD. En effet, le but de CADIXE étant d’annoter des documents cette suppression n’a pas de sens.

- Delete Tree Tags : Suppression récursive de l’ensemble des balises dans la balise courante (désignée par la position du curseur). Cette commande affiche une boîte de confirmation car elle est susceptible de faire perdre la totalité des annotations effectuées dans un document. Par exemple, dans le cas ci-dessus, on aurait :

Avant : `contrast, both | the E213G and E213A Spo0A variants showed`

Après : `contrast, both the E213G and E213A Spo0A variants showed`

- Add comment ... : ajoute un commentaire en texte libre et un « état » (validated, unvalidated, erroneous) à la balise courante (cf description des commentaires). On peut aussi ajouter/éditer un commentaire à une balise en double-cliquant sur la colonne « comment » de la zone XML lorsque la palette des commentaires n’est pas ouverte dans la partie droite de l’éditeur.

Gestion des documents (Menu DOCUMENTS)

- Document Information ... : cette commande permet de visualiser le nom de la DTD utilisée pour l’annotation et d’ajouter des « méta-informations » sur le document afin de l’identifier complètement. Ces informations sont stockées sous la forme de commentaires dans l’entête du fichier XML. On peut indiquer notamment :
 - le titre du document,
 - le nom des annotateurs,
 - les références du document original,
 - des commentaires sur l’historique de l’annotation ...
- Zoom ▶ : indique le niveau de zoom du document courant (par défaut 100%).
- Spacing ▶ : espacement entre les lignes du document courant.
- Window organisation ▶ : permet de modifier le nombre de documents qui sont simultanément visibles dans la partie « annotation » de l’interface ainsi que leur organisation spatiale (horizontale ou verticale). Le nombre maximum de documents affiché est réglable (Cf. partie « préférences »).
- Load Style ... : charge une nouvelle feuille de styles dans le document courant. Ce style devient le nouveau style courant.
- Reload Style : recharge le style courant à partir de son fichier dans le dossier des styles. Cette commande est très utile lorsque l’on est en train de mettre au point une feuille de styles avec un éditeur externe (Cf. « Gestion des styles ») afin de visualiser

rapidement le résultat des modifications effectuées. En effet, cette commande évite d’avoir à fermer puis à recharger le document sur lequel on travaille. Le rechargement ne concerne QUE le document courant, les autres documents qui utilisent ce style ne sont pas directement modifiés.

- **Remove Style** : supprime le style courant de la mémoire (le fichier n’est pas modifié). S’il ne reste plus qu’un seul style en mémoire cette commande est ignorée.
- **Styles ▸** : ce sous-menu contient la liste des styles du document courant qui sont actuellement chargés en mémoire. On passe d’un style d’affichage à l’autre en le sélectionnant un nom de style dans cette liste.
- **Close document** : ferme le document courant (contenant le curseur d’insertion). Si le document a été modifié une boîte de dialogue demande si on veut le sauver.
- *Liste des documents* : le menu se termine par la liste des documents chargés dans l’éditeur. Dans une fenêtre (que l’on soit en mode mono ou multi-documents) pour changer de document courant il suffit d’en choisir un parmi cette liste.

Outils (Menu TOOLS)

- Les quatre premiers éléments de ce menu montre la liste (et l’état : « show/hide ») des palettes d’outils affichables dans la partie droite de l’éditeur. L’état des palettes est automatiquement stocké dans le fichier des préférences. A chaque lancement de l’éditeur l’utilisateur retrouve la configuration qu’il avait lors de la session précédente. L’ordre d’affichage des palettes est modifiable via les préférences (cf « Gestion des préférences »). Les palettes disponibles sont :
 - la liste des balises applicables,
 - les attributs de la balise courante,
 - l’outil de commentaire, *(existe aussi en boîte de dialogue, menu EDIT)*
 - l’outil de recherche. *(existe aussi en boîte de dialogue, menu TAG)*

En fonction de la tâche que l’on est en train d’effectuer, on peut utiliser une ou plusieurs palettes. Ainsi, durant l’annotation, on utilise principalement les palettes « attributs » et « balises » alors que si l’on est en train de commenter une annotation déjà réalisée ou utilisera davantage les palettes « recherche » et « commentaires ».

- **XML Validation of the Document** : cette commande lance la validation du document annoté. Contrairement aux version 1.xx de l’éditeur, la vérification indique dans le panneau inférieur gauche (« Validation result ») l’ensemble des erreurs trouvées.

Lorsque l’utilisateur clique sur une erreur de la liste Cadix positionne automatiquement le curseur sur la position incriminée dans le document. En outre, si l’on double-clique sur une erreur, l’éditeur repasse de surcroît dans le mode de visualisation du code XML (onglet « XML code »).

Dans le code XML, les balises fautives sont repérées à l’aide d’une pastille rouge ; en double-cliquant sur cette pastille on repasse dans le mode de visualisation de la liste des erreurs. Il existe trois principales sortes d’erreur après l’annotation :

- **Warning** : ces erreurs signalent généralement un problème concernant les attributs des balises (valeurs manquantes, valeurs erronée, ...)
- **Erreur simple** : c’est l’erreur la plus fréquente, dans le cas de Cadix elle correspond typiquement à l’absence d’une balise dans le code. Si ce type

d’erreur devient systématique il est préférable que l’utilisateur repasse dans le mode « *Allows to insert the most specific tag* » (cf préférences, option Tags) qui en forçant les utilisateurs à suivre la hiérarchie de la DTD évite que l’on omette trop souvent d’insérer des niveaux intermédiaires.

- Erreur fatale : ce type d’erreur stoppe la validation. Il se produit par exemple lorsqu’il manque une balise fermante dans le code. A priori ce type d’erreur ne devrait pas se produire car Cadix détecte ce problème soit lors du chargement du fichier, soit lors de l’insertion des balises lors d’un « coller ».

Aide (Menu HELP)

Dans cette version de l’éditeur, ce menu permet simplement de faire apparaître la bannière de l’application et de connaître le numéro de la version avec laquelle on travaille.

Gestion des préférences

La boîte de préférences (menu EDIT) permet de configurer le comportement général de l’éditeur CADIXE. On accède aux possibilités suivantes :

- *Default paths* : indique à l’éditeur où il peut trouver les différentes ressources telles que : le nom du modèle par défaut ainsi que les chemins d’accès pour les modèles, les DTD, les feuilles de styles, les scripts et les documents annotés.

Lorsque le chemin est relatif la position peut-être donnée soit par rapport à l’application (c’est le cas standard), soit par rapport au chemin d’accès qui a été passé dans le paramètre *-userhome* de la ligne de commande. La sélection de l’une ou l’autre de ses possibilités se fait à l’aide du « radio-bouton ».

On peut ainsi configurer l’éditeur pour que les ressources « partagées » (DTD, Models, Styles, ...) soient gérées globalement au niveau de l’application alors que les ressources « utilisateurs » (les préférences, les documents) soient rangées au niveau du dossier de travail.

- *Editor* : Permet de définir l’ordre dans lequel les outils (palettes) sont rangés dans la partie droite de l’interface. Cette option permet également de réglage des paramètres généraux de l’éditeur comme : l’ouverture automatique d’un nouveau document initialisé avec le modèle par défaut lors du lancement de l’application, le nombre maximum de documents visibles simultanément lorsque l’on est dans le mode multi-document (cf. menu DOCUMENTS), et enfin le nombre de fichiers qui sont mémorisés dans la liste de la commande « Load Recent » (menu File).
- *Annotation* : option d’annotation permettant de dire si la sélection d’une zone de texte s’effectue au niveau du caractère ou directement au niveau des mots (défaut).
- *XML zone* : configuration de l’affichage de la zone XML. On peut ainsi choisir la police et sa taille des caractères utilisés dans cette fenêtre. Il est également possible de sélectionner les couleurs et styles associées aux balises, aux attributs, aux valeurs ainsi qu’à l’extrait de texte placé à droite de la balise.
- *Attributes* : paramétrage de l’éditeur pour renseigner la valeur des attributs lors de l’insertion d’une balise et contrôler comment sont générés les attributs dans le code XML du document annoté. On a les options suivantes :
 - *Display a dialog to fill the attributes*:
Permet de choisir si, lorsqu’on introduit une nouvelle balise, une boîte de dialogue doit apparaître afin de forcer l’utilisateur à saisir la valeur des attributs. Trois options sont possibles :
 - « Never » : on utilise seulement la partie édition de la fenêtre.
 - « Always » : la boîte apparaît dès qu’il y a un attribut à renseigner.
 - « When REQUIRED » : elles apparaît s’il y a un attribut REQUIRED.
 - *Verify if REQUIRED attributes are set* :
Lorsque l’affichage de la boîte de saisie des valeurs d’attributs est actif, cette option permet de forcer l’utilisateur à toujours renseigner les attributs REQUIRED. La boîte ne se ferme alors que lorsque chacun d’eux a une valeur.

- *Allow to suppress the attributes :*

Lorsque cette option est active une « check-box » apparaît à côté de chaque attribut dans les zones d’édition fenêtre principale et boîte de saisie. Elle permet d’activer/désactiver les attributs. Un attribut désactivé ne sera pas écrit dans le code XML à moins que ce soit un attribut REQUIRED et que l’option « Always include Required attributes » soit activée. Cette option permet d’éviter d’inclure dans le code XML des attributs qui ne sont pas valués.

- *Tags* : option d’affichage de la liste de balises avec les options suivantes :

- *Allows to insert the most specific tag :*

Si l’option est désactivée (défaut) l’utilisateur ne peut sélectionner que des balises qui sont immédiatement plus spécifiques que la balise courante, en d’autres termes des balises qui sont des « filles » immédiates de la balise courante dans la DTD. Par contre, si l’option est désactivée toutes les balises plus spécifiques sont acceptables. Le mode « strict » laisse moins de liberté dans l’ordre dans lequel l’annotation peut-être effectuée, mais il est très utile pour guider ceux qui ne maîtrisent pas encore bien une DTD.

- *Display relevant tags only:*

Si l’option est activée (défaut) seules les balises qui sont effectivement applicables apparaissent dans la liste des balises. Si elle est désactivée, les balises non applicables apparaissent en grisé. Cette dernière possibilité est intéressante lorsque l’on a des DTDs petites comportant peu de balises.

- *The tags are validated by default :*

Option inutilisée dans la version 2.0a4

- *Toolbar* : configuration des commandes à mettre dans la barre d’outils. L’affichage se compose de 2 colonnes : à gauche la liste des commandes à faire apparaître dans la barre des outils (endessous des menus) et à droite la liste de toutes les commandes qui sont disponibles dans l’éditeur. L’utilisateur peut changer l’ordre d’apparition des commandes par « drag & drop » dans la liste de gauche.

Les styles paramétriques

Il est maintenant possible d’associer simultanément *plusieurs styles* à une même balise la sélection du style à utiliser s’effectuant *dynamiquement* au moment de l’affichage en fonction de la valeur prise par de l’un (ou plusieurs) de ses attributs. Cette possibilité permet notamment de limiter le nombre de balises à introduire dans la DTD lorsque l’on veut différencier visuellement plusieurs balises codant des informations ayant une même sémantique, mais se différenciant uniquement sur la valeur de leurs attributs.

Par exemple, dans la DTD produite par le projet CADERIGE, il y a actuellement plusieurs balises <AGENT 1> ... <AGENT n> qui ont été introduites de manière un peu arbitraire dans le seul but de pouvoir différencier « graphiquement » les différents agents impliqués dans une interaction. Avec les styles paramétriques, il devient possible de n’avoir qu’une seule balise <AGENT> globale ayant plusieurs styles, le style à utiliser étant dynamiquement indexé par le numéro de l’agent qui devient alors un attribut.

Ainsi, à la balise : <Agent id=1 type= "protein"> on va associer une feuille de style spécifique nommée « Agent_1 » par concaténation du nom de la balise et de la valeur de l’attribut ID et ainsi de suite avec les valeurs 2, 3, ... Lors de la concaténation effectuée pour générer le nom du style on utilise le caractère « _ » (underscore) pour séparer le nom de la balise ainsi que les différentes valeurs entre elles. Pour utiliser cette possibilité, il faut donc tout simplement créer les styles CSS correspondants dans les fichiers de styles et ajouter dans le style associé à la balise « père » (la balise <AGENT> dans l’exemple ci-dessus) l’attribut « **cadixe-index-attribute** » avec comme valeur le (ou les) attribut(s) dont la valeur pour va indiquer le nom du style¹ (cf. « Gestion des styles » pour plus de détails).

Si l’on prend un cas plus concret, par exemple celui où l’on a une balise <INTERACTION> qui possède un attribut « SENS » qui peut prendre les deux valeurs possibles ACTIVE et INHIBE. Dans le fichier de style on mettra (parmi d’autres lignes évidemment) :

- 1) Pour indiquer sur quel attribut se passe la “spécialisation” :
 - `.interaction { ... cadixe-index-attribute : sens ; ... }`
- 2) Puis on va créer les 2 feuilles de styles spécifique ci-dessous (en ajoutant au nom de la balise les valeurs possibles séparées par le caractère “_”). Ici la variation porte sur la couleur du texte mais tout est possible. Elles seront utilisées automatiquement dans l’affichage en fonction de la valeur prise par l’attribut « SENS » :
 - `.interaction_active { ... color : red ; ... }`
 - `.interaction_inhibe { ... color: blue ; ... }`

Finalement, au niveau de l’affichage, on obtiendra :

Balise présente dans le document	Affichage obtenu
<interaction sens="active"> Gene Abc </interaction>	Gene Abc
<interaction sens="inhibe"> Gene Abc </interaction>	Gene Abc

¹ Notons qu’il est relativement facile d’introduire cette nouvelle possibilité dans les fichiers qui ont été déjà annotés avec une DTD précédente en utilisant les possibilités de remplacement de chaînes de caractères offertes par les éditeurs de textes. Par exemple dans le cas ci-dessus, il suffira de substituer systématiquement dans les fichiers la chaîne : « <AGENT1 » par « <AGENT ID=1 » et cela pour chaque numéro d’agent existant.

Gestion des feuilles de styles

Contrairement à la version précédente, la configuration des différents fichiers de styles ne se fait plus via l'éditeur. En effet, les nouveaux styles sont directement codés sous forme de fichiers au format CSS (comme dans les documents au format HTML) et il est donc possible de les modifier soit à l'aide d'un éditeur de texte soit, plus simplement, par l'intermédiaire d'un éditeur de CSS. Il en existe pour toutes les plateformes (Linux, MacOS X, Windows), voici quelques possibilités, si vous en connaissez d'autres, merci de me le signaler ...

- Multiplateformes :
 - <http://www.nvu.com/>
 - http://www.westciv.com/style_master/
 - <http://www.ucware.com/juststyle/index.htm>
- Linux :
 - <http://cssed.sourceforge.net/>
- MacOS X :
 - <http://www.macrabbit.com/cssedit/>

Par défaut les styles sont placés dans le dossier « formats/styles ». Voici la liste des attributs qui sont actuellement gérées par CADIXE, les autres attributs des CSS sont pour l'instant ignorés. Le but n'est pas de les gérer tous, mais nous espérons toutefois en introduire de nouveaux dans la version finale de l'éditeur ☺. Voici un exemple d'affichage :

YxiE was induced by phosphate starvation independently of **sigma(B)** and **PhoPR**.
the *Bacillus subtilis* response regulator **Spo0A** or valine substitution

Les attributs spécifiques à CADIXE qui sont absents du standard « CSS 2.0 » commencent tous par le préfixe « cadixe- » :

Attributs standard de gestion des polices et des couleurs :

- background-color : *color* ;
- color : *color* ;
- font-family : *font name* ;
- font-style: italic ;
- font-size : *number* ;
- font-weight : bold;
- text-decoration : {underline, line-through} ;

Concernant les "boîtes" autour du texte :

- border-color : *color* ;
- border-width : *number* ;
- border-style : {solid, dotted, dashed} ;
- **cadixe-border-shape** : {rectangle} ;

Concernant la déclaration des "styles paramétriques" :

- **cadixe-index-attribute** : *attribut1, attribut2, ...* ;

Concernant les icônes à utiliser pour les balises de type « EMPTY » :

- **cadixe-empty-tag** : chemin relatif à une image de type GIF (~ 12pt), le dossier racine est (temporairement !!) le dossier « resources/icons ».

Contact

L’éditeur d’annotation Cadixe a été initialement développé dans le cadre du projet CADERIGE (<http://caderige.imag.fr/>) financé dans le cadre d’une ACI Bioinformatique. Un grand merci à l’équipe MIG de l’INRA pour son soutien actif. Merci également au SIB (Institut Suisse de Bioinformatique) qui a participé et soutenu ce développement.

Pour toutes questions, remarques, propositions d’amélioration de l’éditeur, implémentation de nouvelles fonctions, contactez : gilles.bisson@imag.fr

Si vous constatez un problème ou anomalie quelconque dans l’éditeur, afin de faciliter sa correction, il faut nous transmettre si possible les fichiers (documents, DTDs, styles) sur lesquels vous étiez en train de travailler au moment du problème ainsi qu’une description précise du problème que vous avez rencontré.