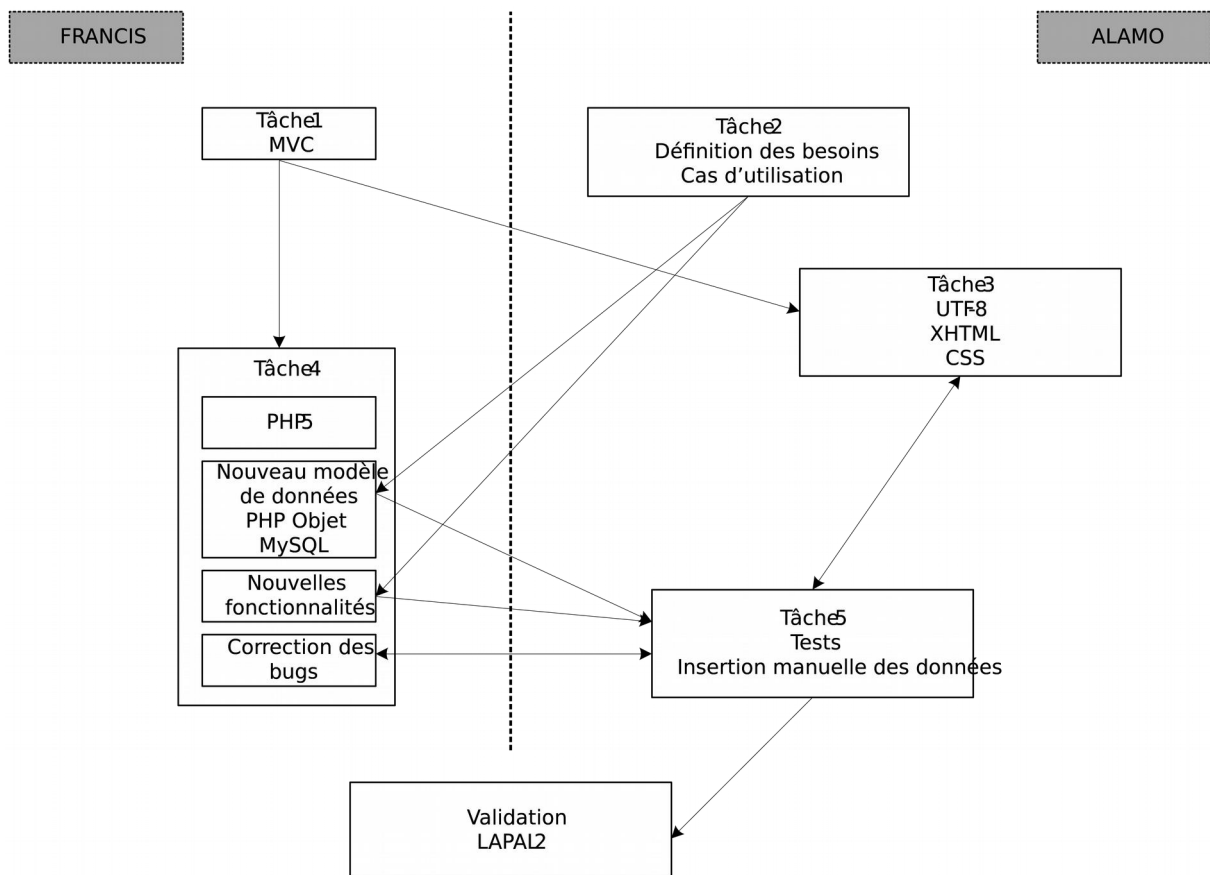


I.1. Vue générale

Le schéma suivant représente les différentes tâches et leurs dépendances :



2. Description des tâches

2.1. Modèle MVC : [Cette étape - effectuée - aboutira à la version LAPAL 2](#)

modification de la structure de l'application selon le modèle MVC (Modèle-Vue-Contrôleur). Cette tâche est indispensable avant de pouvoir effectuer tout nouveau développement de l'application. Il s'agit de bien séparer :

- Les fonctions php
- Les éléments communs à chaque page (balise head, menu, logo, css, images, etc.)
- Les pages du site, ne contenant plus que des appels logiques à des fonctions php et des inclusions des fichiers communs

La tâche sera répartie en 5 sous-tâches :

2.1.1. Structure générale de l'application, menus et authentification

2.1.2. Fonctionnalités de visualisation

2.1.3. Fonctionnalités de création

2.1.4. Fonctionnalités de modification

2.1.5. Fonctionnalités de suppression

Cette étape est-elle achevée ?

Bug à corriger : la suppression d'un schéma ne doit plus entraîner la suppression des sous-schémas - ni des contraintes qui leur sont associées- qu'il utilise ;

La suppression d'un schéma ne devrait pas entraîner la suppression des sous-schémas - ni des contraintes qui leur sont associées- qu'il utilise ;

Actuellement (dans Lapal 2) tous les sous-schémas - et les contraintes associées - utilisés par le schéma supprimé sont supprimés (ce qui ne devrait pas être possible) ; néanmoins un message notifie, avant validation, les sous-schémas - et les contraintes - sur le point d'être supprimés ;

d'autre part la suppression d'un sous-schéma ne supprime pas les schémas l'utilisant, bien que ceux-ci ne fonctionnent plus (possibilité d'insertion d'un message d'alerte ?)

En complément de cela - et en remplacement des majuscules/minuscules (distinction des schémas principaux (orphelins) des schémas utilisés par d'autres schémas) - peut-on imaginer la mise en place d'un compteur indiquant si S utilise des sous-schémas -et combien- et s'il est utilisé comme sous-schéma - et par combien ?

2.2. Définition des besoins pour LAPAL 3

description des cas d'utilisation de lapal3 dans un futur proche et éloigné. Cela permettra d'optimiser la définition du nouveau modèle de données. Ensuite, à partir de cette analyse, nous pourrons définir quelles éventuelles nouvelles fonctionnalités (ie. Fonctionnalités non disponibles dans lapal1) implémenter tout de suite dans lapal2. Cette tâche peut démarrer dès maintenant - plus l'analyse sera approfondie, plus lapal2 répondra aux exigences actuelles et futures des ses utilisateurs.

2.3. Mise à jour de la présentation : il s'agit de mettre à jour les feuilles de style, l'encodage des caractères et le standard html. Cette tâche pourra démarrer dès que le modèle MVC (tâche 1) sera implémenté, et n'interférera pas avec les développements futurs, puisque les scripts php et les pages html auront été séparées.

2.4. Refonte de la partie applicative : cette tâche ne peut démarrer qu'une fois la tâche 2 terminée. Elle consiste à :

-Convertir le code php3 en php5.

-Définir un nouveau modèle de données, puis créer les classes php5 correspondantes et une nouvelle base de données MySQL.

-Mettre à jour les fonctions php pour qu'elles utilisent ces nouvelles classes et cette nouvelle base de données.

-Développer de nouvelles fonctions php correspondant aux nouvelles fonctionnalités à éventuellement implémenter (jusque-là, tout le travail n'aura donc consisté qu'à redévelopper les fonctionnalités existantes de lapal1)

-Corriger les bugs et améliorer les scripts durant la phase de test.

2.5. Phase de tests : il s'agit de tester lapal2, faire un retour de bugs et de demandes d'améliorations. Cette phase servira également à insérer les données dans la nouvelle base de données (les données de lapal1 ne seront pas récupérées automatiquement).

3 Définition des besoins et modèle de données pour LAPAL 3 - Etape 2

3.1 Etat des lieux - LAPAL 2

Un des buts de l'ALAMO est la génération automatique de textes littéraires, étant donné certaines contraintes. LAPAL2 est un langage-auteur de générations de textes littéraires avec contraintes, exploitable sur Internet. Une première version de LAPAL2 a fait l'objet d'un mémoire d'étudiants de maîtrise informatique de Paris XIII 1999-2000 (LAPAL2000).

LAPAL2000 a été écrit en PHP3 associé à MySQL. Les schémas et les contraintes reposent sur des structures arborescentes (voir ci-dessous)

3.1.1. Généralités.

La structure de base commune à toutes les productions, appelée, **schéma**, est, en premier lieu, une arborescence, caractérisée par

- l'existence d'une racine r, sommet unique sans prédécesseur.

- la propriété : pour tout sommet x de l'arborescence, il existe un chemin et seul ayant r pour origine et x pour extrémité.

Les sommets n'ayant pas de successeurs sont appelés feuilles, les autres : noeuds.

Une arborescence peut être définie récursivement.

L'auteur construira un tel schéma ou en modifiera un existant déjà.

L'arborescence ne sera pas complètement figée : l'auteur pourra prévoir des possibilités d'omission, de permutation ou d'itération de sous-arborescences (voir plus loin).

A chaque feuille de l'arborescence sera associé un **corpus** spécifique où s'effectueront les choix, au cours des étapes de la production déclenchée par le lecteur. Ces corpus sont propres au schéma et doivent être saisis par l'auteur lors de la construction de ce schéma ou sont au contraire des corpus externes déjà disponibles. Les corpus sont composés d'items individuels auxquels sont associés des attributs ou d'extraits de corpus existants.

Les **contraintes** assurent la cohérence syntaxique et sémantique du texte à produire. Elles s'expriment sous forme de relations entre des attributs associés aux éléments textuels.

Exemples :

1) Le schéma est un quatrain composé de quatre vers. La rime est un attribut d'un vers. Une contrainte est : " le premier et le dernier vers ont même rime".

2) Le schéma est une phrase : PH-> suj/ve/comp, suj ->item, ve->item, comp ->det/adj/nom,

det->item, adj ->item, nom->item.

Les items ont une valeur et des attributs (genre, nombre) sur lesquels portent des contraintes d'égalité.

3.1.2. Fonctionnement général de LAPAL (version 2)

Les principes généraux de la construction d'un schéma par l'utilisateur de LAPAL

3.1.2.1. Les schémas

La "racine" d'un schéma en est l'élément fondamental : "sonnet" s'il s'agit d'un sonnet, "discours" s'il s'agit d'un discours, etc.. On peut cependant donner des noms différents à un schéma et sa racine : cela permet de définir différents modèles de sonnets (ou de discours).

Un élément constitutif d'une structure de schéma peut

- se décomposer en sous-éléments

- être un élément de base ("non décomposable"), à choisir dans un corpus (qui peut n'avoir qu'un seul élément).

La définition d'un schéma de création d'écriture comporte donc les étapes suivantes : définition du nom du schéma (ce nom figurera dès lors dans le catalogue) ;

1) définition des structures du schéma (définition de la racine, règles de décomposition des éléments, mode de détermination des éléments non décomposables) ;

2) définition du système d'attributs du schéma (saisie des attributs associés à cet élément du schéma puis entrée des contraintes d'attributs associées aux règles de décomposition) ; cette étape peut être supprimée, si l'auteur précise que son schéma ne comporte pas de contraintes d'attributs (on a alors une génération uniquement combinatoire)

et saisie des items imposés et des corpus spécifiques , si le schéma en comporte.

3) Après chaque étape, on peut revoir (et modifier) tout ce qui a été précédemment défini; on peut aussi décider d'interrompre la définition du schéma (qui sera sauvegardé à l'état incomplet et pourra donc être repris à tout moment).

Dès lors que les fonctions « modification » et « suppression » fonctionnent à chacune des étapes - schéma, corpus, attributs, contraintes - la conception globale du schéma est « évolutive », elle n'est en rien figée en début de la réalisation (construction du schéma) ; le schéma doit pouvoir - tout au long de sa construction, et après - être révisable ;

3.1.2.2. Les Corpus

Les seuls corpus possibles dans la version LAPAL 2 sont ceux créés par l'utilisateur lui-même, ou repris (et éventuellement modifiés) parmi ceux antérieurement créés dans LAPAL ; ils sont dits internes en opposition à ceux importés dans le système à partir de dictionnaire ou d'une base de données textuelles : liste d'expressions toutes faites, de fragments de dialogues, syntagmes, unités narratives, épisodes, etc.- appelés ; pour cela appelés externes

3.1.2.3. Attributs et Contraintes

Les types d'attributs actuellement possibles sont les suivants : verbal(énumération), textuel, numérique, logique.

Les types de contraintes utilisables dans la version LAPAL sont des contraintes dites simples (relation entre des attributs) en opposition aux contraintes dites conditionnelles, contraintes de la forme :

SI <contrainte> ALORS <contrainte> SINON <contrainte conditionnelle>

Les contraintes simples actuellement utilisables sur les attributs sont des contraintes d'égalité ou de différence (avec plus ou moins de succès)

dysfonctionnement de la contrainte différence : elle semble fonctionner comme la contrainte égalité - ne pourrait-on pas concevoir la contrainte « différence » (sur (les attributs ou) les éléments des corpus) comme naturelle (par défaut (sans indication contraire)) : les contraintes « différence » -

3.2. Perspectives - Définition des besoins et modèle de données pour LAPAL 3

3.2.1. Les axes de développement vers LAPAL 3

3.2.1.1. Un premier axe de travail consiste :

1) ou bien en l'amélioration de cette version LAPAL2000, en particulier l'augmentation de la diversité des types de contraintes possibles, l'amélioration de l'ergonomie et de l'interface graphique, l'ajout d'un manuel en ligne.

Il serait important de simplifier la circulation de l'attribution des contraintes des constituants situés sur des branches différentes de l'arborescence.

Il y a là un problème d'héritage des attributs qui mérite une étude théorique. Pour ce problème et pour d'autres autres, on devrait exploiter la notion de "classes" et la notion d'héritage dans les langages orientés objet.

2) ou bien en s'engageant sur une voie différente avec d'autres langages :

On pourrait envisager des représentations non arborescentes qui donnent d'autres informations que la structure : les réseaux ATN, graphes auxquels sont associés des conditions de changement d'état (conditions pour emprunter un arc : genre, nombre,...) et des actions suivant ce changement (construction de structures syntaxiques partielles par exemple) ; les réseaux sémantiques.

Ces représentations pourraient éventuellement apparaître comme constituants d'une structure arborescente.

Dans les deux cas, il semble indispensable, avant la programmation, de définir un langage de contraintes, également utilisable par l'auteur (celui qui écrira les programmes de production).

3.2.1.2 Un deuxième axe de développement concerne :

l'introduction, dans le logiciel, de nouvelles fonctionnalités linguistiques et littéraires : automatiser la gestion des contraintes usuelles (accords en genre et en nombre, aspects du verbe, etc.) et donc permettre l'utilisation en ligne de quelques-uns des outils grammaticaux et sémantiques ; organiser la manipulation et l'exploitation de bases de données textuelles se situant à un niveau hiérarchique supérieur au niveau lexical, des phrases ou expressions toutes faites du "langage cuit" jusqu'aux épisodes stéréotypés.

Il faut donc d'abord étudier l'existant dans le domaine des lexiques, dictionnaires, analyseurs syntaxiques actuellement disponibles. Quelles sont les possibilités d'obtenir des informations concernant l'orthographe, la phonétique, la sémantique.

Ensuite définir des normes de présentation de ces dictionnaires et de ces analyseurs adaptées au fonctionnement du logiciel et écrire des programmes d'adaptation de bases de données externes dans le format désiré.

Pour faciliter les importations et exportations on peut envisager de représenter en XML les corpus, dictionnaires et textes à produire, et utiliser un TEI (Text Encoding Initiative). Il faut alors un éditeur XML (exemple : XENA écrit en java ou wordperfect).

3.2.1.3. Un troisième axe de développement concerne :

la mise en œuvre, dans l'utilisation du logiciel, de concepts et de techniques multimodales, en organisant la gestion de données de type image ou de type son, en définissant aussi, à cette occasion, les nouveaux types de contraintes susceptibles de contribuer, à côté des contraintes linguistiques et littéraires usuelles, à la production assistée des œuvres.

Dans une organisation "orientée objet", l'introduction de constituants de type image ou son ne pose pas de problèmes de principe. Il faut cependant prévoir, lors de la spécification de l'entité "Objet LAPAL de base", les formats et les attributs propres à ces nouvelles structures de données. En particulier, les relations de contraintes devront s'ouvrir à des rapports spatiaux et temporels nouveaux.

Dans ce qui suit, nous rappelons les notions de base de LAPAL : structures, contraintes, corpus et établissons le MENU qui sera à la disposition de l'utilisateur du logiciel.

Pour la clarté des explications, nous ferons appel à trois personnages qui entrent en jeu dans LAPAL, appelés informaticien, auteur, lecteur.

L'informaticien écrit le logiciel LAPAL. L'auteur écrit des programmes en utilisant LAPAL. Le lecteur exécute ces programmes. L'utilisateur peut être un ou plusieurs de ces personnages.

Que veut faire l'auteur ? Des programmes qui vont générer des phrases, des poèmes, des scénarios, ou tout autre texte défini par des contraintes. L'auteur choisit les contraintes qui vont définir et structurer ces productions.

3.2.2. Précisions sur la constitution des SCHEMAS, CORPUS, CONTRAINTES - objectifs de LAPAL 3

3.2.2.1. Objectifs sur les schémas

le schéma doit pouvoir - tout au long de sa construction, et après - être révisable ; mais le retour sur la définition des objets (attributs, corpus, schémas, contraintes) -leurs modifications- pose alors la question de l'hérédité : la rupture d'une certaine procédure ordonnant la construction du schéma (définition des attributs - construction du corpus- structure du schéma - définition des contraintes) peut rompre les liens entre les objets (attributs, corpus, schémas, contraintes) assurant la cohérence du schéma

Il serait nécessaire d'ajouter à la possibilité de permuter des constituants, dans une règle de décomposition (et ceci à un niveau quelconque de l'arborescence des constituants) la possibilité d'omettre certains de ces constituants.

Possibilité pour chaque attribut, contrainte, corpus, sous-schéma d'affichage du schéma (ou des schémas) auquel il/elle se rattache ;

Possibilité de visualiser les propriétés de chacun des attributs, contraintes, corpus, schémas : auteur, date, éléments associés (corpus, attributs, contraintes, schémas et sous-schémas)

3.2.2.2. Objectifs sur les Corpus

Les corpus pourront être externes (c'est-à-dire importés dans le système à partir de dictionnaire ou d'une base de données textuelles : liste d'expressions toutes faites, de fragments de dialogues, syntagmes, unités narratives, épisodes, etc.) ou internes (c'est-à-dire créées par l'utilisateur lui-même). L'importation - hors création de schéma - d'un corpus externe comportera nécessairement une phase de formatage (sous le contrôle de l'utilisateur) permettant d'associer aux items du corpus la suite des valeurs des attributs qui leur sont attachés et qui seront utilisés au moment de la mise en action des contraintes, c'est-à-dire dans la phase de production de l'œuvre.

Dans une phase déjà avancée de la mise au point de *LAPAL3*, ce processus de production pourra présenter un aspect récursif : dans ce cas certaines feuilles de l'arborescence pointeront vers des bases de données composées elles-mêmes de sous-arborescences susceptibles, à leur tour, de donner lieu à une production de sous-ensembles de l'œuvre finale.

On pourra distinguer aussi corpus textuel et corpus fonctionnel (c'est-à-dire où la valeur textuelle de l'item sélectionné doit être calculée en fonction d'autres attributs). Cette dernière possibilité devrait être exploitée dans le cadre d'une automatisation partielle de la gestion des accords syntaxiques et des accords sémantiques élémentaires.

Possibilité d'incorporation des productions dans un corpus à définir (avec formatage)

3.2.2.3. Objectifs sur les attributs

3.2.2.3.1.-Extension des types d'attributs utilisables : verbal, textuel, numérique, logique, typographique, autre...

3.2.2.3.2.-Les contraintes dites simples (relation entre des attributs) pourront être de différents types :

- indépendantes du type des attributs : égalité, inégalité, appartenance (des valeurs des attributs à une liste définie)

- ensemblistes : attributs compatibles (ayant au moins un élément commun), attributs incompatibles (deux à deux disjoints), le premier est l'ensemble de suivants, le premier est la réunion des suivants

- numériques : le premier vaut 1 de plus (ou de moins) que le suivant, le premier est la somme des suivants.

3.2.2.3.3. -Mises en place de contraintes dites conditionnelles, de la forme :

SI <contrainte> ALORS <contrainte> SINON <contrainte conditionnelle>

Exemple : Si les attributs "sexe" de PERSONNAGE 1 et "sexe" de PERSONNAGE 2 sont égaux, ALORS "style" de SCENARIO est à choisir parmi "roman scout" et "drame homosexuel".

3.2.2.3.4.-Le bon fonctionnement de LAPAL3 dépendra pour une part essentielle de la mise en œuvre du jeu des contraintes, donc du système des attributs disponibles et de la palette des relations que ces attributs devront vérifier au moment des sélections successives d'items. Tout comme dans le cas des corpus, les éléments constitutifs du jeu des contraintes pourront être choisis dans des listes préparées à l'avance ou, au contraire, spécifiées directement par l'utilisateur.

Les attributs externes (ou préétablis) seront empruntés aux domaines de la grammaire ou à des généralisations naturelles : genre, nombre, personne, champ thématique, aspect, caractéristique rhétorique ou stylistique, etc. Leur valeur pourra être celle-là même, convenablement abrégée, que leur donne l'usage : masc., fem., sing., pers1, prés-ind., etc. Les attributs créés par l'utilisateur pourront être identifiés et recevoir des valeurs choisies de façon à en traduire efficacement la signification et l'emploi.

Les relations externes (préétablies) seront, classiquement :

identique à

différent de

compatible avec

Dans le cas d'attributs numériques on pourra évidemment utiliser les relations ensemblistes ou algébriques usuelles : $<$, $>$, $+$, $-$, \times et définir des relations de compatibilité ou d'incompatibilité spécifiques.

Ici encore, les relations définies par l'utilisateur se référeront de façon logique aux particularités des attributs pour lesquels ils seront pertinents. Ils devront correspondre à des opérations disponibles (donc déjà programmées) dans le système.

Dans la programmation du logiciel, les contraintes ne devront pas être vérifiées après l'exploration combinatoire mais en cours de cette exploration.

3.2.2.4. Précisions sur les contraintes

On doit distinguer les contraintes déjà imposées par la structure du schéma et celles qu'on y ajoute. Prenons quelques exemples.

suite d'éléments choisis dans les corpus :

normalement, donnée par le parcours (de Tarry) des feuilles de l'arborescence. On peut prévoir des priorités dans l'algorithme général pour les fils d'un nœud, sinon, l'auteur doit les préciser (comme des poids par exemple, d'où contraintes supplémentaires d'inégalité)

Exemples (on suppose que les priorités sont incluses):

Phrase. Contraintes supplémentaires : accords (égalité)

Scénario (voir ALP¹, p.328 : 2 ou 3 corpus) ; contraintes de "réalisme" et autres.

Poèmes. Corpus de vers. Une strophe est une suite de vers, un poème une suite de strophes. Contraintes supplémentaires : rimes, nombre de pieds, etc

"le tra du des" . Un corpus (alors poids avec contrainte de supériorité) ou deux corpus. Contraintes supplémentaires : accords.

Petites annonces. Plusieurs corpus. Contraintes supplémentaires : compatibilité de catégories à définir par l'auteur.

Aphorismes (voir ALP p.315) Plusieurs corpus. Contraintes de compatibilité.

Concaténation d'expressions.

Concaténer des hémistiches pour fabriquer de nouveaux alexandrins.

Concaténer des expressions connues en les liant avec une proposition (voir ALP p 259) Ex : Attention à la marche ou crève

Fabriquer des "mots-valises"

Contraintes liées.

La contrainte sur un élément dépend des contraintes déjà appliquées ou des données

Exemples

Boule de neige (Voir ALP p 194). Poème dont chaque vers est fait d'un mot qui augmente d'une lettre à chaque vers. Le Corpus doit contenir des mots dont le nombre de lettres est un attribut (il peut y avoir d'autres attributs !). La contrainte change à chaque vers. Si C(i) est : mot de i lettres, alors C(i+1) est : mot de i+1 lettres.

Acrostiche (Voir PLS² p.103) Poème dont chaque vers commence par une lettre (ou un mot) donnée.

Pouvoir lier chaque feuille (issue d'un seul corpus) du schéma à une valeur donnée/fixée de l'attribut

Modifier un texte.

Pour pouvoir modifier un texte, il faut la possibilité de saisir ses éléments sur un schéma et ensuite d'en changer certains par tirage avec contraintes.

Exemples. Littérature itérative (Voir ALP p.85)

Remplacer des mots par leur définition, récursivement.

4. MENU

¹ Atlas de Littérature Potentielle

² Pour la Science, nov 1998

L'utilisateur de *LAPAL3* conduira son activité d'auteur en choisissant parmi les options d'un menu initial comportant :

spécification de l'organisation structurale d'un schéma de création d'œuvres (littéraires ou multimodales).

saisie (ou importation) des bases de données textuelles (ou informationnelles) convenablement choisies contenant les constituants littéraires (ou iconiques ou sonores) qui contribueront à la production d'une œuvre.

spécification des jeux de contraintes qui permettront de filtrer, parmi les assemblages tirés au hasard dans ces bases de données, les combinaisons admissibles d'items lexicaux ou autres. Ces contraintes seront relatives aux valeurs numériques ou textuelles des attributs attachés aux items en cours de sélection.

Edition – éventuellement modification – des divers constituants du schéma, en cours de spécification ou après les premiers essais d'exploitation.

Possibilité d'exportation les productions sur un site, blog, etc... (avec adresse URL des schémas)

Page interface de sélection des corpus selon leurs propriétés (attributs, auteurs, dates, etc...) ; nécessité d'indexer les corpus, attributs, schémas lors de leur création

Exploitation d'un schéma préalablement défini (c'est-à-dire production d'une œuvre satisfaisant aux contraintes imposées).

Reprise, sous éditeur, d'une œuvre produite par l'un des schémas déjà réalisés à des fins de complétion ou de mise en forme.