

Algorithme pour la génération de documents personnalisés structurés Optimisation par fusion des classes d'utilisateurs

Vincent Ranwez, Sylvie Ranwez

▶ **To cite this version:**

Vincent Ranwez, Sylvie Ranwez. Algorithme pour la génération de documents personnalisés structurés Optimisation par fusion des classes d'utilisateurs. NîmesTIC' 2001, Relation homme-système : complexe ?, Dec 2001, Nîmes, France. edutice-00000382

HAL Id: edutice-00000382
https://edutice.hal.science/edutice-00000382

Submitted on 3 Mar 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algorithme pour la génération de documents personnalisés structurés

Optimisation par fusion des classes d'utilisateurs

Vincent Ranwez[1] et Sylvie Ranwez[2]

[1]*LIRMM, 161, rue ADA , F-34 392 Montpellier Cedex 5*

ranwez@lirmm.fr

[2]*LGI2P, EMA/Site EERIE, Parc scientifique G.Besse, F-30 035 Nîmes cedex1*

Sylvie.Ranwez@ema.fr

RESUME : *Les nouvelles technologies de l'information et de la communication, favorisent l'envoi de documents à un grand nombre de personnes (plusieurs milliers). Les publicitaires ont depuis longtemps compris qu'en s'adressant à trop de personnes en même temps le message risque d'être dépersonnalisé et donc perdre de son intérêt et de son impact. Il semble alors souhaitable de personnaliser – au moins en partie – le contenu du document afin d'en augmenter l'attrait et la pertinence. Pour cela on peut utiliser plusieurs critères pour définir différents profils de destinataires et utiliser des méthodes de composition automatique. Si ces critères sont suffisamment fins, on risque alors d'avoir presque autant de documents à générer que d'utilisateurs.*

Dans cet article nous proposons une manière efficace de contourner ce problème en utilisant la structure du document. En effet deux utilisateurs peuvent être différents suivant un critère C, mais si ce critère n'est pertinent que pour une sous-partie du document il suffit de générer de manière spécifique à ce critère uniquement cette partie du document. Si pour chaque partie d'un document on définit une partition des utilisateurs en classe d'équivalence, il suffit de construire chaque partie du document une fois par classe d'utilisateur et non une fois par utilisateur. Dans cet article nous présentons un algorithme efficace pour définir les classes d'utilisateurs correspondant à une partie d'un document, à partir des critères pertinents pour cette partie.

MOTS CLES : *Documents adaptatifs, compositions automatique, modélisation de l'utilisateur, classification, algorithme de tri*

ABSTRACT: *The New Technologies of Information and Communication – NTIC – encourage the mass mailing of documents (to several thousands of people). Advertisers know that sending the same message to too many people at the same time increases the risk of the message becoming impersonal, and thus losing its interest and impact. It would therefore seem beneficial to personalize the message in order to increase its appeal and relevance. This personalization can be achieved by using various criteria to characterize the recipients. Based on these profiles, documents that are more appropriate to each kind of profile can be automatically generated. However if the criteria are too specific there is a risk of producing almost as many different documents as there are recipients.*

In this paper we propose a way to use the structure of the document to avoid this problem. Two users may be different with respect to a criterion C, but if this criterion is only relevant to one subpart of the document the system can produce the same document for both users except for this subpart, which will be specific for each of them. If users are grouped into equivalence classes for each part of the document, the system can then generate one subpart for each class instead of one subpart per user. In this paper we present an efficient algorithm for defining user classes for one part of a document, using only the criteria that are relevant to that part.

KEY WORDS: *Adaptive documents, automatic composition, user modeling, classification, sort algorithm.*

1. Introduction

Le développement des Nouvelles Technologies de l'Information et de la Communication favorise, pour les entreprises et les institutions, la diffusion d'informations en grande quantité et pour un vaste public. Le risque de ces diffusions à grande échelle est alors que le message soit dépersonnalisé et perde ainsi de son intérêt et de son impact. C'est pourquoi il paraît indispensable de bénéficier de techniques de personnalisation des documents : c'est ce que nous appelons la composition automatique de documents personnalisés [RAN 00].

La modélisation de l'utilisateur est au centre de ce processus de composition. Cependant, dans le cas d'envoi massif d'informations, il peut s'avérer utile de mettre en évidence différentes classes d'utilisateurs afin de minimiser les traitements. Ces classes sont déterminées pour chaque partie du document à partir des différents critères qui caractérisent un utilisateur.

Après avoir positionné notre étude par rapport aux travaux existants, nous présentons les principes de la composition automatique de documents personnalisés. Nous proposons ensuite un algorithme de fusion des classes d'utilisateurs correspondant à des classes pertinentes pour une partie du document. Cet algorithme optimise le temps de traitement de l'information puisqu'il permet pour chaque partie de réduire le nombre de fois où elle est composée.

2. Etat de l'art et contexte de notre étude

La personnalisation de document pour un ou plusieurs utilisateurs suppose l'existence d'une représentation exploitable de ces utilisateurs par le système qui compose le document.

2.1. Modélisation de l'utilisateur

Les techniques de modélisation de l'utilisateur font l'objet de nombreuses recherches, par exemple dans les domaines des interfaces en langage naturel ou des applications pédagogiques. Dans ce dernier l'adaptation à un niveau de connaissance est primordiale pour favoriser le transfert d'information. On y distingue essentiellement deux sortes de modèles : overlay et buggy.

Le modèle overlay, ou par recouvrement [WEN 87][MUR 98], utilise la représentation de la connaissance d'un domaine pour modéliser celle de l'utilisateur. Ce modèle se construit au fur et à mesure que l'utilisateur parcourt un document, en mémorisant les partie visitées. On peut également donner des précisions sur l'état des connaissances : binaires (su, pas su), qualitatives (bon, moyen, pauvre) ou quantitatives (la probabilité qu'il connaisse ce concept est de…) [BRU 98]. Ce type de modèle est le plus simple à mettre en œuvre.

Le modèle buggy consiste à déterminer les comportements incorrects d'un apprenant face à un système, faire un diagnostic d'erreurs, et générer un nouveau cours qui tente de corriger les erreurs [GOU 00].

D'autres méthodes, comme la reconnaissance de plan (séquence ordonnée d'actions) ou l'apprentissage symbolique, sont présentées dans [PY 98]. L'utilisation des réseaux bayésiens est décrite dans [GON 98].

Au-delà d'une volonté de motivation et d'optimisation de transfert d'information, les applications adaptatives ont un impact économique important. Les différentes applications présentées dans [WAH 89] en atteste. Contrairement aux systèmes coopératifs que cet article décrit, nous ne nous intéressons pas aux erreurs de compréhension du domaine qui après diagnostic permettent de modifier les informations présentées (utilisé surtout dans les systèmes experts), mais nous nous concentrons sur ses caractéristiques personnelles et ses objectifs.

Dans notre approche nous nous inspirons essentiellement du modèle overlay [CRA 00]. La connaissance d'un utilisateur est contenue dans un vecteur qui mentionne les concepts du domaine qui ont été validés, acquis. Ce vecteur comporte également d'autres caractéristiques personnelles, comme le sexe de la personne, son age, son niveau d'étude, ainsi que ses préférences concernant la présentation des informations (audio, vidéo, système d'exploitation, partage d'écran, etc.) et ses objectifs (recherche pointue ou généraliste).

Nous ne présentons pas ici les problèmes soulevés par la saisie du modèle utilisateur. Précisons néanmoins qu'une partie des caractéristiques est renseignée directement par la sélection d'un stéréotype. Certaines de ces caractéristiques pouvant être modifiées comme dans [RIC 89].

2.2. Domaine d'application : l'envoi de résumés d'actualité personnalisés

La volonté marquée de développer l'accès rapide et pour tous aux informations est particulièrement intéressante dans le domaine de l'Information en ligne. En effet les NTIC offrent une alternative intéressante aux médias traditionnels en permettant à l'utilisateur de s'informer sur l'actualité en fonction de ces préférences. De plus l'auditeur-spectateur[1] bénéficie des mêmes avantages qu'un journal télévisé, puisqu'il peut disposer de vidéos, d'enregistrements, de coupures de presse, de photos, etc.

L'intérêt de la presse écrite pour le nouveau média qu'est l'Internet et la nécessité d'adapter le contenu et la présentation des informations à l'utilisateur sont largement évoqués dans [IKS 00].

Dans la suite, nous étudions essentiellement la composition de documents personnalisés, à partir de documents électroniques préexistants.

2.3. Contexte de l'étude

Les méthodes de composition automatique présentées dans [CRA 98,00] et [RAN 00] se situent dans le contexte particulier où la composition a pour but de fournir un document personnalisé à un utilisateur donné. Dans cet article, le contexte est élargi à la composition automatique d'un document pour un grand nombre de personnes. Cette situation se retrouve, par exemple, dans le cas de l'envoi en grand nombre de résumés d'actualité. Ces travaux sont en lien direct avec le projet PROMPTER de création de résumés personnalisés de rencontres sportives [RAN 01]. Cependant notre approche est généralisable, par exemple, à la composition de bande annonces, de publicités personnalisées ou des cours à distance.

Ce contexte précis impose d'optimiser l'étape de composition. En effet, l'envoi en nombre de documents personnalisés implique la génération d'un grand nombre de documents. L'optimisation du temps de composition consiste alors à 'factoriser' certaines parties du document spécifiques à une classe d'utilisateurs.

[1] L'utilisateur d'un tel système est tour à tour auditeur, lecteur, spectateur des informations qui lui sont transmises.

3. Composition automatique de documents personnalisés

Les principes de la composition automatique de documents adaptatifs et leur mise en œuvre sont présentés dans [RAN 00]. Cette approche utilise les ontologies du domaine comme base à la qualification des segments documentaires, appelés Briques d'Information (BI) [RAN 99]. Ces ontologies sont également utilisées dans le processus même de la composition [CRA 00].

Nous avons définit la composition comme étant la structuration d'un ensemble de Briques d'Information, réalisée pour un être humain, avec l'objectif d'en faire émerger une plus grande valeur sémantique.

Le processus de composition peut être divisé en quatre phases distinctes, qui font l'objet des quatre sections suivantes :

- Recherche d'Information ;
- Sélection - Filtrage ;
- Organisation ;
- Assemblage.

3.1. L'Organisation

Les quatre étapes de la composition documentaire ne sont pas forcément réalisées dans l'ordre présenté ci-dessus. En effet, l'étape d'organisation des segments peut être réalisée à posteriori. Connaissant les différentes BI qui doivent apparaître dans le document, on détermine le rôle qu'elles peuvent jouer et on définit un ordre entre ces BI de façon à respecter une trame narrative pertinente [RAN 00]. Néanmoins, l'étape d'organisation peut également être le point de départ de la composition et c'est elle qui initie alors l'étape de recherche d'informations. Connaissant la structure finale du document, on recherche les informations, en fonction du rôle, i.e. de la place qu'elles occupent, dans le document final. On peut utiliser pour cela les grammaires formelles [ROI 99].

Dans le cas d'envoi de "news" personnalisées, plusieurs grammaires peuvent être utilisées, la composition suivant l'une d'elles, en fonction des préférences des utilisateurs. Ainsi chaque grammaire détermine l'ordre et la structure du document souhaité. Un utilisateur peut préférer l'actualité internationale, nationale ou locale en premier. Un autre peut désirer un regroupement de toute l'actualité politique (internationale, nationale et locale) avant de voir l'actualité sociale. La grammaire suivante (fortement simplifiée par soucis de clarté) permet de définir une structure couramment utilisée pour un résumé d'actualité[2]. Cette structure est présentée dans Fig. 1.

G_{JL} = {V_T, V_N, S, P} avec :
V_T = ensemble des rôles[3] utilisés dans un journal en ligne ;
V_N = {Internationale, Nationale, Locale, Culture, Sport};

S = {News}

P est défini par les règles R0 à R5.

```
R0 : <News> ::= <Une> <Internationale><Nationale>
                <Locale> <Culture> <Sport>
R1 : <Internationale> ::= <Politique> <Economie>
                          <Société>
R2 : <Nationale> ::= <Politique> <Economie>
                     <Société>
R3 : <Locale> ::= <Politique> <Economie><Société>
R4 : <Culture> ::= <Cinéma> <Musique>
                   <Littérature> <Théâtre>
R5 : <Sport> ::= <Football> <Tennis> <Athlétisme>
```

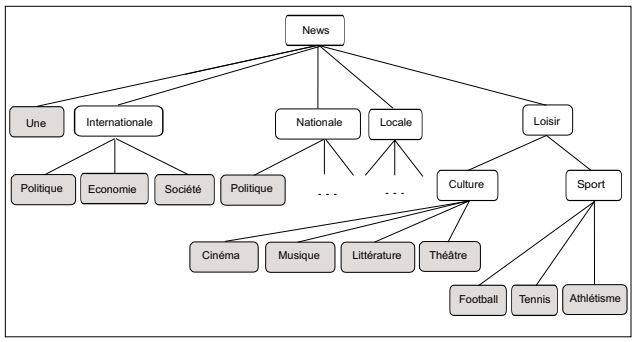


Fig. 1 : Structure d'un résumé d'information construit à partir de la grammaire formelle G_{JL}

3.2. La recherche d'information

L'étape de recherche d'information consiste à déterminer les BI candidates pour faire partie d'un document. Ce choix est réalisé en fonction des objectifs et grâce à la description sémantique du contenu des BI. Plus performante qu'une simple recherche par mots-clés, la recherche par la sémantique du contenu permet dans un contexte particulier, définit par l'ontologie du domaine, d'éviter les problèmes liés à la polysémie des mots et d'affiner les recherches.

Dans notre cas précis, les objectifs sont fixés par l'annonceur. La structure du document est alors connue (c.f. étape d'organisation), et c'est en fonction des différentes parties du document que va s'effectuer la recherche.

3.3. La sélection / filtrage

La sélection consiste à filtrer uniquement les informations nécessaires pour un profil utilisateur donné. C'est à ce niveau-là que les temps de traitement peuvent s'avérer coûteux puisqu'il s'agit de déterminer le degré de pertinence, de chaque BI en fonction de certains critères de l'utilisateur. Ensuite, suivant le temps disponible de celui-ci, un seuil est mis en place de façon à ne sélectionner que les BI pertinentes et d'une durée acceptable [CRA 98].

3.4. L'assemblage

Les techniques d'assemblage permettent d'insérer des transitions entre les BI de façon à créer un liant qui facilite la lecture. Par exemple si deux BI qui ont comme support médiatique la vidéo se succèdent, on peut spécifier qu'il faut insérer un fondu enchaîné par le noir entre les deux.

[2] C.f. http://fr.news.yahoo.com ou http://www.lemonde.fr/

[3] Voir la notion de rôle dans [RAN 00]

Nous ne détaillerons pas ces techniques ici. Dans un premier temps, en effet, nous nous contentons de présenter séquentiellement les BI sans insérer de transition.

4. Gérer des classes d'utilisateurs

Nous montrons l'intérêt de gérer des classes d'utilisateurs et nous décrivons le problème algorithmique que pose la gestion de ces classes. Nous présentons ensuite la manière dont des approches classiques permettent de résoudre ce problème puis nous proposons une approche spécifique à ce problème qui permet de le résoudre en un temps linéaire par rapport à la taille des données.

4.1. De l'intérêt de définir des classes d'utilisateurs

Afin d'éviter de composer plusieurs fois les mêmes parties d'un document, on peut regrouper les utilisateurs ayant exactement le même profil. Plus le nombre de critères utilisés pour décrire les utilisateurs est important plus la personnalisation du document est pertinente. Evidemment il devient alors de plus en plus rare d'avoir des utilisateurs ayant exactement le même profil.

Le nombre de critères à prendre en compte pour décrire un utilisateur dépend aussi de la complexité et de la taille du document à produire. Ainsi si l'on souhaite composer un résumé de l'actualité sportive uniquement pour deux utilisateurs ayant le même profil, il suffit de composer une seule fois ce résumé. Par contre s'ils souhaitent un résumé complet de l'actualité (contenant entre autres un résumé sportif) on utilisera des critères supplémentaires. Il est alors fort probable que leurs profils ne soient plus exactement les mêmes et que le système automatisé compose deux résumés distincts. La partie spécifique à l'actualité sportive sera donc composée deux fois alors que pour cette partie précise, les critères pertinents étant identiques pour les deux utilisateurs c'est la même composition qui est réalisée. Pour éviter cela on peut utiliser le fait qu'à chaque étape de la composition seul un sous-ensemble des critères utilisés pour décrire l'utilisateur est pertinent. Pour chaque partie composée, chaque feuille de l'arbre de la Fig. 1, il est possible de spécifier le sous-ensemble des critères pertinents. Dans l'exemple ci-dessus cela permet de ne composer qu'une fois le résumé sportif.

4.2. Problématique

Dans le cas où un seul critère C est pertinent pour une partie, tous les utilisateurs ayant la même valeur pour ce critère peuvent être regroupés. Ainsi le critère C définit des classes d'utilisateurs. Le nom de ces classes n'a pas de signification particulière, il ne sert qu'à les distinguer. Pour fournir à chaque utilisateur le document approprié, il suffit alors de faire la composition pour un représentant de chaque classe et de réutiliser le document obtenu pour les autres utilisateurs appartenant à la même classe.

De manière plus générale, chaque critère C_1 définit une partition des utilisateurs que l'on suppose connue. Pour chaque critère on choisit un représentant par classe d'utilisateurs, et pour chaque utilisateur on indique son représentant. L'exemple détaillé ci-dessous illustre la manière dont il est possible de stocker l'ensemble de ces informations en utilisant deux vecteurs pour lesquels l'indice *i* représente l'utilisateur *i* :

Critère C_1

Représentant	1	2	1	1	5	2
Classe	2	3	2	2	1	3
Indice	*1*	*2*	*3*	*4*	*5*	*6*

Ces deux vecteurs permettent par exemple de savoir que l'utilisateur 1 est dans la classe 2 et que son représentant est l'utilisateur 1 (lui-même). De même l'utilisateur 6 est dans la classe 3 et son représentant est l'utilisateur 2. Si ce critère C_1 est le seul pertinent pour une partie, celle-ci ne sera composée que 3 fois.

Supposons que pour une autre étape de composition il y ait deux critères pertinents, le critère C_1 ci-dessus et le critère C_2 suivant :

Critère C_2

Représentant	1	2	1	1	4	2
Classe	1	2	1	3	3	2

Dans ce cas on peut remarquer que les utilisateurs 1 et 3 sont dans la même classe pour C_1 (classe 2) et pour C_2 (classe 1), une seule composition suffit donc pour ces deux utilisateurs. Par contre l'utilisateur 4 est dans la même classe pour C_1 mais pas pour C_2. Deux utilisateurs sont dans la même classe s'ils sont dans la même classe pour C_1 et pour C_2 (ils ont la même signature).

Critère C_{12} (Critère C_1 + Critère C_2)

Représentant	1	2	1	4	5	2
Classe	1	2	1	3	4	2

4.3. Les solutions existantes

Dans ce qui suit nous décrivons un algorithme permettant d'obtenir les classes et les représentants de deux critères à partir des classes et représentants de ces deux critères pris séparément. La complexité de cet algorithme est $O(n)$, n étant le nombre d'utilisateurs. On peut alors considérer le résultat obtenu comme la description d'un nouveau critère et utiliser le même algorithme pour prendre en compte un troisième critère. On procède ainsi jusqu'à prendre en compte tous les critères pertinents pour la partie en cours de composition. Ainsi, si une partie met en jeux p critères on obtient les classes d'utilisateurs correspondantes en utilisant $(p-1)$ fois notre algorithme de complexité $O(n)$, soit une complexité totale de $O(np)$.

Une approche classique pour obtenir les utilisateurs équivalents pour un ensemble de critères consiste à définir un ordre sur les valeurs de chaque critère (par exemple leur numéro de classe) et un ordre sur les critères, puis à trier les utilisateurs en fonction de ces ordres. Une fois triés, les utilisateurs d'une même classe sont consécutifs. Si l'on procède ainsi pour obtenir les classes

correspondantes au critère C_{12} il faut trier les 6 couples de valeurs associés aux 6 utilisateurs U_1..U_6 :
$U_1(2,1)$, $U_2(3,2)$, $U_3(2,1)$, $U_4(2,3)$, $U_5(1,3)$et $U_6(3,2)$.

Une fois ces couples triés on obtient la liste suivante : $U_5(1,3)$, $U_1(2,1)$, $U_3(2,1)$, $U_4(2,3)$, $U_2(3,2)$, $U_6(3,2)$ et en parcourant une fois cette liste il est possible de déterminer les classes d'utilisateurs. La comparaison de deux éléments dépend du nombre de critères pris en compte *O(p)* et le nombre de comparaisons de l'algorithme de tri dépend du nombre d'utilisateurs à trier *O(nlog(n))* la complexité totale de cette approche est donc *O(nplog(n))*.

Une autre approche classique consiste à utiliser un arbre préfixe pour stocker les critères associés à chaque utilisateur. Dans ce cas si l'on utilise un tableau pour gérer l'ensemble des classes qui peuvent être associées à un critère et que l'on note *d* le nombre maximal de valeurs que peut prendre un critère on obtient un algorithme en *O(npd)*. Si l'on reprend l'exemple précédent en supposant que *d=3* on obtient l'arbre de Fig. 2.

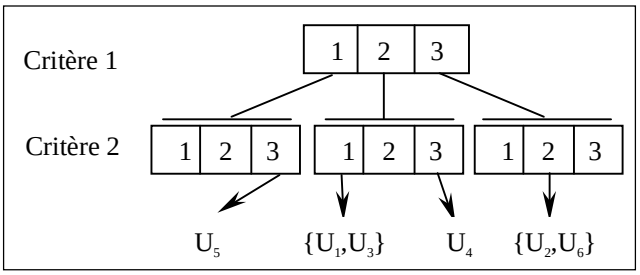


Fig. 2 : Utilisation d'arbre préfixe

Dans le cas général, on sait seulement que $d \leq n$, donnant dans le pire des cas une complexité en $O(n^2p)$. Si l'on utilise une liste au lieu d'un vecteur on réduit l'espace mémoire nécessaire, mais pas la complexité. Cependant en utilisant des structures de données plus adaptées que les listes ou les vecteurs (arbres équilibrés) on peut réduire la complexité à *O(nplog(n))* [GUS 97].

4.4. Une solution en temps linéaire

Dans le cas particulier où les valeurs à trier sont bornées par le nombre *n* d'éléments à trier, il est possible de les trier en *O(n)*. Pour cela on utilise deux vecteurs annexes dont la taille est égale à la valeur du plus grand élément à trier. Ce tri classiquement appelé tri par comptage, utilise le premier vecteur annexe pour compter le nombre d'éléments ayant une certaine valeur et le second pour compter le nombre d'éléments inférieurs ou égaux à une certaine valeur. Si l'on note V le vecteur à trier, V_{eq} et V_{le} les deux vecteurs annexes (*eq: equal* et *le: less or equal*) on obtient le vecteur trié V_T de la manière suivante :

Tri_comptage

```
Pour i=1 à n {Veq[i] = Vle[i] = 0 ;}
Pour i=1 à n {Veq[V[i]]= Veq[V[i]]+1 ;}
Vle[1] = Veq[1];
Pour i=2 à n {Vle[i]= Veq[i] + Vle[i-1] ;}
Pour i=1 à n {VT[Vle[i]]=V[i]; Vle[i]=Vle[i]-1;}
```

Pour prendre en compte deux critères C_1 et C_2 on peut attribuer à chaque couple de numéros de classes (k_1,k_2) une valeur unique $k_1(n+1)+k_2$. En effet il ne peut pas y avoir plus de classes que d'utilisateurs et k_1 et k_2 sont donc compris entre 1 et n. Cependant $k_1(n+1)+k_2$ est borné par $n(n+1)+n$. Utiliser le tri par comptage dans ce cas nécessiterait de gérer des tableaux de cette taille et la complexité du tri serait alors $O(n^2)$.

L'algorithme en *O(n)* que nous proposons utilise le tri par comptage sur un seul des critères, par exemple C_1. Les utilisateurs étant triés suivant ce critère, on les considère successivement suivant cet ordre.

Pour obtenir C_{12} à partir de C_1 et de C_2, on utilise trois vecteurs annexes (classe1, classe12 et ref12) pour associer aux éléments de référence de C_2 les informations suivantes :

- la dernière classe de C_1 associée à cette classe de C_2
- le numéro k_{12} de la classe correspondante dans C_{12}
- l'utilisateur de référence pour cette classe dans C_{12}

Les utilisateurs étant classés en fonction de C_1, lorsque l'on considère un nouvel utilisateur de classe (k_1,k_2) on recherche l'utilisateur de référence pour k_2. Si la classe associée est k_1, on positionne les valeurs des vecteurs de C_{12} pour cet utilisateur grâce aux informations associées à l'utilisateur de référence k_2.

Puisque les utilisateurs sont classés en fonction de C_1, si la classe associée à k_2 est différente de k_1 on est sûr que le couple (k_1,k_2) n'a jamais été rencontré. On crée alors une nouvelle classe k_{12} dont l'utilisateur courant est la référence et l'on met à jour les informations de l'élément de référence de k_2. En notant V_U l'ensemble des utilisateurs, VC_X le vecteur indiquant les classes des utilisateurs pour le critère X et VR_X le vecteur indiquant l'utilisateur de référence pour le critère X, l'algorithme est le suivant :

Fusion_classes

```
Trier VU avec Tri_comptage suivant C1;
C12=1 ;
Pour i=1 à n
 {classe1[i]=-1; classe12[i]=-1; ref12[i]=-1;}
Pour i=1 à n
  {u= VT [i] ;
  k1 = VC1[u] ; k2=VC2[u] ; ref2=VR2[u] ;
  si (classe1[ref2] != k1)
  alors{ classe1[ref2]=k1 ;
         classe12[ref2]=k12 ;
         ref12[ref2]=u ;
         k12=k12+1 ; }
VC12[u]=classe12[ref2] ;
VR12[u]=ref12[ref2] ; }
```

5. Conclusion et perspectives

Les applications personnalisées sont de plus en plus souvent au centre des préoccupations des annonceurs, de tout type, et les travaux concernant les techniques de personnalisation sont nombreux. Pourtant très peu mentionnent l'optimisation de cette personnalisation. Or dans le cas de génération pour un grand nombre d'utilisateurs, il est indispensable de diminuer le temps de traitement de l'information afin de satisfaire un maximum de lecteurs en un minimum de temps.

L'approche que nous avons présentée dans cet article consiste à 'factoriser' certaines étapes de la composition suivant différentes classes d'utilisateurs. Cela permet de réduire le temps de traitement de l'information lors de la composition. Cela permet également de restreindre l'espace mémoire nécessaire pour stocker l'ensemble des documents générés, dans le cas où les documents sont accessibles sur un serveur donné au lieu d'être envoyés en temps réel.

Cette approche peut encore être affinée en l'utilisant non pas pour la composition d'une partie de document, mais pour chaque étape de cette composition.

6. Références

[BRU 98] P. BRUSILOVSKY. « Methods and Techniques of Adaptive Hypermedia ». In Adaptive Hypertext and Hypermedia, P. Brusilovsky et al. (Eds.), Kluwer Academic Publishers, Netherlands, pp. 1-43, 1998.

[CRA 98] M. CRAMPES, J.P. VEUILLEZ, S. RANWEZ. « Adaptive Narrative Abstraction ». Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia, Hypertext'98, Pittsburgh, Pennsylvania, June 20-24, pp. 97-105, 1998.

[CRA 00] M. CRAMPES, S. RANWEZ. « Ontology-Supported and Ontology-Driven Conceptual Navigation on the World Wide Web ». Proceedings of HT'00, the 11th ACM Conference on Hypertext, San Antonio, Texas, 2000.

[GON 98] C. GONZALES, P.H. WUILLEMIN. « Réseaux baysiens en modélisation d'utilisateurs ». Dans Modélisation de l'Apprenant, revue Science et Techniques Educatives Vol. 5, N°2, Hermès, pp. 123-140, juin, 1998.

[GOU 00] G. GOUARDERES, A. MINKO et L. RICHARD. « Traçabilité de l'usager avec un simulateur ». Dans les actes de la conférence internationale sur l'Ingénierie des systèmes et les NTIC, NîmesTIC'2000, pp. 308-321, Nîmes, France, 11-13 septembre 2000.

[GUS 97] D. GUSFIELD. « Algorithms on Strings, Trees and Sequences », Cambridge University Press, ISBN 0521585198, p. 554, August 1997.

[IKS 00] S. IKSAL. « Composition dynamique d'articles thématiques pour des journaux en ligne ». Dans les actes de la conférence internationale sur l'Ingénierie des Systèmes et les NTIC, NîmesTIC'2000, Nîmes, France, pp. 409-415, 11-13 Septembre 2000.

[MUR 98] T. MURRAY. « Authoring Knowledge Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design ». Journal of the Learning Sciences, Vol 7, N° 1, pp. 5-64, 1998. http://www.cs.umass.edu/~tmurray/papers/JLSEon/JLS96.html

[PY 98] D. PY. « Quelques méthodes d'intelligence artificielle pour la modélisation de l'élève ». dans Modélisation de l'Apprenant, revue Science et Techniques Educatives Vol. 5, N°2, Ed. Hermès, pp. 123-140, juin, 1998.

[RAN 99] S. RANWEZ and M. CRAMPES. « Conceptual Documents and Hypertext Documents are two Different Forms of Virtual Documents ». Proceedings of the Workshop on Virtual Documents, Hypertext Functionnality and the Web, WWW'8, Toronto, Canada, pp. 21-27, May 10-15th 1999. http://www.cs.unibo.it/~fabio/VD99/

[RAN 00] S. CHABERT-RANWEZ. « Composition Automatique de Documents Hypermédia Adaptatifs à partir d'Ontologies et de Requêtes Intentionnelles de l'Utilisateur ». Thèse de doctorat en informatique, Université de Montpellier II, décembre 2000.

[RAN 01] S. RANWEZ, F. BARBARE, N. CHANAL « PROMPTER, Compte rendu d'avancement N°1». Rapport interne CR-LGI2P, Ecole des Mines d'Alès, mars 2001.

[RIC 89] E. RICH. « Stereotypes and user modeling ». User Models in Dialog Systems, A. Kobsa and W. Wahlster Eds., Springer verlag, Berlin. pp. 35-51, 1989.

[ROI 99] C. ROISIN. « Documents Structurés Multimédia ». Mémoire en vue de l'obtention d'une habilitation à diriger les recherches, Institut National Polytechnique de Grenoble, 22 septembre 1999. http://www.inrialpes.fr/opera/people/Cecile.Roisin/Welcome-en.html

[WAH 89] W. WAHLSTER, A. KOBSA. « User Models in Dialog Systems : Introduction ». User Models in Dialog Systems, A. Kobsa and W. Wahlster, Eds., Springer verlag, Berlin. p. 2-33, 1989.

[WEN 87] E. WENGER. « Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication and Knowledge », Los Altos (CA), Morgan Kaufmann Publishers, p. 486, ISBN 0-934613-26-5, 1987.