

Intégration d'ontologies pour l'accès à une bibliothèque d'hyperlivres virtuels

Ontology integration to access a virtual hyperbook library

Gilles Falquet

Claire-Lise Mottaz Jiang

Jean-Claude Ziswiler

Centre universitaire d'informatique

24, rue Général-Dufour, CH-1211 Genève 4, Suisse

Gilles.Falquet | Claire-Lise.Mottaz | Jean-Claude.Ziswiler @ cui.unige.ch

Résumé

Un hyperlivre virtuel est un document virtuel formé d'un ensemble de fragments d'information associés à une ontologie de domaine et muni de méthodes ou règles de sélection et d'assemblage. Dans cet article nous étudions le problème de l'accès et de la lecture dans une bibliothèque d'hyperlivres virtuels. Il s'agit de générer des hyperdocuments de lecture qui présentent des informations et connaissances obtenues à partir de plusieurs hyperlivres. De plus ces hyperdocuments doivent correspondre à des objectifs de lecture ou de points de vue particuliers.

L'approche que nous proposons est basée sur l'intégration des ontologies spécifiques à chaque hyperlivre et sur un langage déclaratif pour spécifier les hyperdocuments de lecture. Dans un premier temps nous montrons comment certaines techniques d'intégration d'ontologies peuvent être appliquées dans le cas du modèle d'hyperlivres. Nous présentons ensuite l'inférence de liens sémantiques à partir d'ontologies intégrées, puis l'incorporation des règles d'inférences dans le langage de spécification des hyperdocuments. Finalement nous présentons brièvement une implémentation réalisée à l'aide d'un système de vues hypertextuelles sur des bases de données.

Mots Clef

Document virtuel, ontologie, intégration d'ontologies, interface, bibliothèque numérique,

Abstract

A virtual hyperbook is a virtual document made of a set of information fragments linked to a domain ontology and equipped with selection and assembly methods or rules. In this paper, we study the problem of accessing and reading in a digital library of virtual hyperbooks. In this case it is necessary to generate hyperdocuments that present information and knowledge originating from several hyperbooks. Moreover, these hyperdocuments must fit with the reading objectives or specific point of views.

The proposed approach is based on the integration of the hyperbooks' domain ontologies and on a declarative language to specify the interface hyperdocuments. We first show how current ontology integration techniques can be applied in virtual hyperbook libraries. Then we

show how new interface documents can be generated with the integrated ontologies and how to re-use the hyperbook interface specifications. Finally we present implementation techniques based on databases and hypertext view generation systems.

Keywords

Virtual document, ontology, ontology integration, interface, digital library.

1 Introduction

1.1 hyperlivres et documents virtuels

Depuis quelques années les notions de document virtuel et de personnalisation sont étudiées selon différents angles et différentes problématiques.

La recherche sur les hypertextes a abordé plusieurs domaines qui nous intéressent directement. Des systèmes tels qu'Intermedia (Garret *et al.*, 1986) ou Storyspace (Bernstein, 2002) ont été développés essentiellement dans le but de produire de la littérature hypertextuelle, d'autres tels que KMS (Ackscyn *et al.*, 1988) ou MacWeb (Nanard *et al.*, 1993) ont pour but la gestion et le partage de connaissances. Les notions de lien, d'ancrage, de composition de nœuds, etc. ont été étudiées en détail, ce qui a conduit, entre autres, à la définition du modèle de référence Dexter (Halasz *et al.*, 1994).

Différents modèles et systèmes ont également été proposés pour intégrer sous forme d'hyperlivres la notion de livre et de publication électronique. Il peut s'agir de transformer un livre papier en hypertexte (Rada, 1990) ou en livre électronique (Landoni *et al.*, 2000), d'écrire directement sous forme électronique (hypertextuelle) (Fröhlich *et al.*, 1997) ou encore d'intégrer des documents électroniques existants (Brusilovsky *et al.*, 2002).

Le problème de la personnalisation des hypertextes a conduit à la définition de modèles et de techniques d'adaptation et d'adaptativité. (De Bra *et al.*, 1997) (Brusilovsky, 1998). La capacité d'adaptation correspond à la présentation de contenus différents ou différemment orga-

nisés, en fonction d'un profil de l'utilisateur. L'adaptativité consiste à mettre à jour automatiquement le profil de l'utilisateur en fonction de son comportement. Un exemple bien connu d'adaptativité est le changement de couleurs des liens menant à des pages Web déjà visitées. Dans (Wu *et al.*, 2001) les auteurs proposent un modèle d'hypertexte adaptatif qui comprend un modèle du domaine, un modèle de l'utilisateur et des règles d'adaptation. Le modèle du domaine est un réseau sémantique formé des concepts du domaine et de relations entre ces concepts. Ce modèle sert essentiellement à définir des règles d'adaptation, en fonction, par exemple, des concepts connus (ou maîtrisés) de l'utilisateur.

Parallèlement, de nombreux travaux ont été menés depuis une vingtaine d'années dans le domaine des documents électroniques. Ils ont conduit en particulier à la distinction entre la structuration logique des documents, exprimée dans un langage tel que SGML ou XML, et la restitution des documents sur écran, sur papier, ou autre, exprimée dans un langage de définition de styles tel que CSS ou dans un langage de transformation comme XSLT. Des applications de XML ont également été développées pour structurer des documents dans des domaines d'application particuliers : MathML (formules mathématiques), SVG (graphisme vectoriel), SMIL (contenus multimédias synchronisés), X3D (scènes de réalité virtuelle 3D), etc. La plupart de ces standards sont créés et gérés par le W3 Consortium (www.w3c.org).

Un courant de recherche se dégage actuellement autour de la notion de document virtuel personnalisable (Crampes *et al.*, 1999), (Crampes *et al.*, 1999), (Garlatti & Crampes, 2002). Les documents virtuels personnalisables sont définis comme des ensembles d'éléments (souvent appelés fragments) associé à des mécanismes de filtrage, d'organisation et d'assemblage. En fonction du profil de l'utilisateur ou de ses intentions, ces mécanismes vont produire différents documents adaptés à ses besoins. Le modèle que nous présentons s'inscrit dans cette approche, tout en utilisant la notion de publication hypertextuelle de bases de données.

Crampes et Ranwez (2000) proposent deux modèles de documents virtuels. Tous deux font appels à une ontologie de domaine pour indexer les fragments d'information (ressources). Dans le premier cas une stratégie de « chaînage arrière conceptuel » permet de créer des parcours de lecture correspondant aux objectifs du lecteur (décrits en termes de graphes conceptuels). Dans le second cas, une ontologie pédagogique définit des règles pédagogiques qui guident l'assemblage de fragments de manière à produire des documents qui respectent une approche pédagogique définie. Ces règles contraignent en particulier l'ordre d'apparition des informations dans un document. Un moteur d'inférence se charge de générer des documents qui satisfont les règles.

Garlatti et Iksal (2002, 2002b) proposent un modèle de documents virtuels très détaillé et étendu. Celui-ci est basée sur quatre ontologies qui modélisent le domaine, les méta données, l'application et l'utilisateur. Ces onto-

logies permettent une approche purement déclarative de la composition des documents. Dans cet article nous présenterons un modèle qui s'inscrit dans cette perspective.

1.2 Bibliothèque de documents virtuels

Etant donné le développement rapide des techniques dans le domaine des hyperdocuments, et des documents virtuels, il est raisonnable de penser que les bibliothèques numériques vont peu à peu intégrer non seulement des documents numériques traditionnels, structurés ou non, mais également des documents virtuels personnalisables. Il est dès lors intéressant d'explorer les nouvelles possibilités d'accès et de lecture qu'offre une bibliothèque de documents virtuels. La principale différence avec une bibliothèque numérique classique est la disparition du caractère monolithique du livre ou de l'article. La possibilité de sélectionner et d'assembler des fragments d'information provenant de différents documents virtuels ouvre de nouvelles perspectives de lecture mais pose également de nombreuses questions.

Si l'on applique le principe des documents virtuels aux bibliothèques de documents virtuels, un système de lecture devrait pouvoir composer de nouveaux documents à partir de *tous* les fragments d'information de la bibliothèque et en fonction des objectifs du lecteur. Par exemple, un lecteur souhaitant se renseigner sur la notion de récursivité devrait obtenir un document contenant une définition de cette notion, éventuellement d'autres définitions présentant des points de vue alternatifs, des exemples tirés de divers ouvrages, des exercices, etc. On peut également considérer qu'un livre virtuel, une fois inséré dans une telle bibliothèque, va s'enrichir automatiquement grâce à des fragments tirés d'autres livres (nouveaux exemples et exercices, nouveaux commentaires sur certaines notions, etc.). Ceci demande bien évidemment de s'assurer de la compatibilité sémantique des fragments avant de les réutiliser.

L'objectif est donc des fournir au lecteur de nouveaux documents qui soient sémantiquement et narrativement cohérents. Pour cela nous proposons une approche qui se base d'une part sur l'intégration des ontologies de domaine et d'autre part sur la réutilisation des spécifications des documents d'interfaces (aspect narratif).

1.3 Les ontologies et leur intégration

Il semble évident que dans le domaine des livres virtuels et a fortiori dans les bibliothèques virtuelles, l'intégration et la réutilisation des ontologies jouent un rôle majeur. Si on suppose que chaque livre virtuel possède sa propre ontologie de domaine, on a besoin d'une technique d'intégration pour créer une bibliothèque virtuelle sémantiquement cohérente. Il faut noter ici qu'il n'est pas pensable que tous les livres d'une bibliothèque utilisent la même ontologie de domaine, car soit une telle ontologie n'existe pas, soit elle ne convient que pour les concepts stables et au contour net.

La littérature concernant l'intégration est très hétérogène. Nous allons néanmoins en faire un survol et catégoriser les approches existantes.

On peut s'inspirer de (Pinto *et al.*, 1999) pour dresser une typologie des principales méthodes d'intégration. Quelques travaux font référence à la construction d'une ontologie par assemblage, extension et alignement d'ontologies de base, en dressant des liens entre les concepts. Les ontologies existantes vont persister et vont faire partie de la nouvelle ontologie. Ce processus d'intégration est souvent choisi si les différentes ontologies couvrent des domaines complémentaires. Mais dans ce cas, il n'est pas possible de modifier ou de supprimer les concepts, et le déplacement de concepts n'est pas toujours réalisable. Ces limitations deviennent particulièrement importantes au moment où on constate que la terminologie de l'ontologie n'est plus à jour ou que sa structure n'est pas appropriée aux autres ontologies. En revanche, si la forme des différentes ontologies est similaire, cette approche est relativement facile à utiliser et peut même être automatisée.

D'autres approches consistent à fusionner les différentes parties (merging / matching) en une seule ontologie cohérente. Dans une telle ontologie, on retrouve des régions des anciennes ontologies qui sont plus ou moins inchangées. Ce genre d'approche est utilisé quand les différentes ontologies à intégrer ont une forme ou un contenu très hétérogène. Le résultat est une ontologie plus homogène qu'avec une approche par assemblage, mais le processus d'intégration n'est pas facilement automatisable.

Comme exemples, on peut mentionner les projets Ontolingua et KRAFT. Ontolingua (Gruber, 1992) (Farquhar *et al.*, 1996) permet la création collaborative d'ontologies ainsi que la réutilisation d'ontologies de base grâce à une bibliothèque incluse dans le serveur. Le système contient également le langage KIF (Knowledge Interchange Format), basé sur la logique du premier ordre, qui a été développé afin de faciliter l'échange des différentes ontologies. Les opérations d'intégration sont décrites dans (Farquhar *et al.*, 1997). Il y a principalement des outils pour créer l'ontologie par assemblage des concepts existants (ontology inclusion operator) et des mécanismes pour renommer et désambiguer les termes. L'inclusion d'une ontologie dans une autre est réalisée comme suit. Tout d'abord, le vocabulaire (les symboles non-logiques : noms des relations, les fonctions et les objets constants, ainsi que les axiomes) d'une ontologie est traduit dans le langage de l'autre ontologie. Ensuite, les axiomes sont annotés dans l'autre ontologie.

Dans le projet KRAFT (Preece *et al.*, 2000), des agents (médiateurs et facilitateurs) vont effectuer des traductions parmi les différentes ontologies afin de pouvoir les fusionner. Les médiateurs proposent des services de base comme le filtrage, le triage ou l'assemblage des connaissances reçues des autres agents ; ils fusionnent les contraintes. Les facilitateurs aident à établir la communication entre les agents ; ils servent d'annuaires "Pages Jaunes". L'approche fonctionne même si les langages ne correspondent pas. Plusieurs types de mappings sont

possibles : l'approche supporte à la fois des mappings simples entre deux classes ou valeurs, mais aussi des mappings plus complexes entre des expressions composées. Mais avec cette approche très flexible la sémantique peut être modifiée, parce que l'utilisateur peut créer des mappings erronés ou des contradictions.

Quelle que soit l'approche utilisée, l'intégration des ontologies reste une tâche difficile et coûteuse. Il peut aussi arriver qu'on ne trouve pas les ontologies de base dont on a besoin ou qu'on doive utiliser des ontologies qui n'ont pas le niveau de spécialisation ou de diversité souhaité. D'autres propriétés, comme l'héritage, doivent encore être étudiées ; il n'est pas garanti que toutes les propriétés des ontologies de base soit respectées entièrement dans l'ontologie intégrée. Dans notre cas particulier, il faut encore prendre en compte le fait qu'une bibliothèque de documents virtuels peut être très évolutive. L'arrivée de nouveaux documents va nécessiter un processus d'intégration constant afin que l'ontologie reste adaptée à la bibliothèque. Pour toutes ces raisons, nous proposons d'utiliser une approche d'intégration simple, qui consiste à conserver les concepts et les liens des ontologies initiales, et à établir des relations d'équivalences entre les éléments des ontologies. Cette approche est décrite plus en détail dans la section 3.2.

Dans la suite de cet article nous proposons tout d'abord un modèle simple de documents virtuels (hyperlivres) (section 2). Puis nous décrivons une approche multi-point de vues de l'intégration d'hyperlivres dans une bibliothèque d'hyperlivres (section 3). Dans la section 4 nous montrons comment utiliser le modèle pour définir des documents de lecture "globale" dans la bibliothèque intégrée. La conclusion présente brièvement les techniques d'implémentation que nous avons utilisées pour réaliser des prototypes de livres et de bibliothèques virtuels.

2 Un modèle d'hyperlivre et de bibliothèque

Dans le modèle d'hyperlivre que nous proposons il y a une séparation nette entre la partie structurelle (représentation de l'information) et la spécification de l'interface de lecture et d'écriture (documents d'interface). La partie structurelle est elle-même composée de fragments d'information, d'une ontologie de domaine et d'une structure d'hyperlivre

Les fragments constituent le contenu informationnel de base de l'hyperlivre. Chaque fragment possède un contenu structuré, qui est un ensemble d'élément et de sous-éléments XML, et appartient à une ou plusieurs catégories. De plus, les fragments peuvent être liés entre eux par des liens typés.

La liste des catégories et des types de liens est définie dans la structure d'hyperlivre que nous décrirons au paragraphe 2.3.

2.1 Ontologie de domaine

L'ontologie de domaine donne une représentation formelle des concepts du domaine étudié ainsi que des différentes relations qui lient ces derniers; elle ne contient pas les concepts pédagogiques, narratifs et structurels. Cette ontologie est utilisée pour indexer ou qualifier les fragments.

Il existe actuellement de nombreux formalismes et langages pour représenter des ontologies. Si les formalismes du type réseau sémantique ou classe-relation sont simples à comprendre et à traiter, ils ont cependant un pouvoir d'expression qui est parfois trop limité pour exprimer de manière détaillée des définitions de concepts. De plus, on remarque que plus les descriptions de concepts sont détaillées, plus les mécanismes d'intégration d'ontologies peuvent être efficaces dans la comparaison de concepts. Dans (Falquet & Mottaz Jiang, 2000) nous avons par exemple décrit un formalisme qui étend un modèle de type classe-relation de manière à obtenir un pouvoir d'expression équivalent à certaines logiques de description tout en gardant l'aspect réseau du formalisme. Nous avons également développé un algorithme de comparaison syntaxique de concepts pour ce modèle. Ce genre de formalisme (famille des logiques descriptives) combine donc deux avantages : son aspect classe-relation qui sera fort utile pour la construction de l'interface et son pouvoir d'expressivité qui pourra être exploité à des fins d'intégration.

2.2 Structure d'hyperlivre

A l'heure actuelle il n'existe pas de consensus sur un modèle commun de document virtuel. On peut cependant constater que la grande majorité des modèles proposés comprennent une ontologie de domaine et une base de fragments. C'est au niveau de la spécification de l'assemblage et du modèle de document que l'on trouve le plus de variété (structure narrative, structure argumentative, ontologie pédagogique, moteurs d'inférence spécifiques, etc.)

Dans le modèle que nous utilisons nous séparons cette spécification en deux parties : une partie structurelle qui représente les relations existant entre fragments et entre l'ontologie de domaine et les fragments ; une partie dynamique/d'interface qui décrit l'assemblage des fragments pour produire les documents de lecture.

La structure d'hyperlivre est composée des classes représentée sur le diagramme de classes de la figure ci-dessous (on a utilisé une notation de type UML).

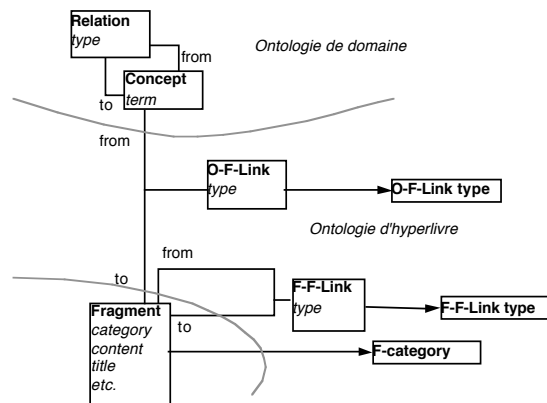


Fig. 1. Classes de la structure d'hyperlivre

La partie structurelle d'un hyperlivre est un ensemble d'objets qui sont des instances des classes de la structure d'hyperlivre. Les deux classes O-F-Link et F-F-Link sont des classes associatives (associations) qui représentent les liens entre l'ontologie de domaine et les fragments et entre les fragments eux-mêmes.

La catégorie d'un fragment indique la nature de son contenu, elle est indépendante du rôle que peut jouer le fragment vis à vis d'un concept ou d'un autre fragment. Les liens entre fragments peuvent être de plusieurs natures – structurels (fragment composé de sous-fragments) – argumentatifs (liens de type argument, positions, contradiction, ...) – narratifs/rhétoriques (renforcement, élaboration, résumé, ...)

Il faut noter que si le schéma de classes est fixe, les instances vont varier selon les hyperlivres. En particulier, chaque hyperlivre peut avoir ses propres catégories de fragments, types de liens et rôles.

Le typage des liens entre concepts et fragments permet de qualifier un fragment non seulement en indiquant de quoi il parle mais également de préciser de quelle nature est sa relation avec le concept. Les types de liens que l'on trouve fréquemment sont, entre autres :

- *instance (exemple, illustration)* : le fragment décrit une instance particulière du concept
- *définition* : le fragment contient une définition textuelle (ou autre) du concept
- *propriété* : le fragment décrit une propriété du concept
- *référence (utilise)* : le fragment fait référence à ce concept (il est nécessaire de connaître le concept pour comprendre le fragment)

Ces liens typés jouent un rôle crucial dans l'inférence de liens pertinents entre fragments lors de la génération des documents d'interface. L'idée étant de remplacer les liens directs entre fragments par des liens inférés à partir des liens entre les fragments et l'ontologie et des relations de l'ontologie. La figure suivante montre deux liens dérivés obtenus en parcourant des liens ontologie-fragment et des relations entre concepts.

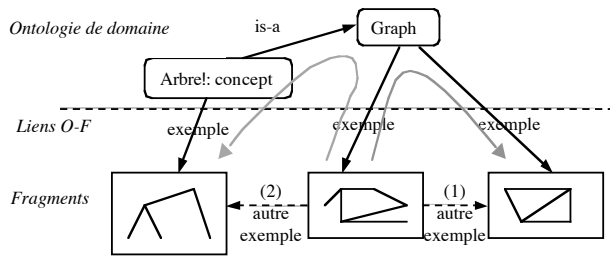


Fig. 2. Inférence de liens à travers l'ontologie de domaine

Etant donné que la structure d'hyperlivre et l'ontologie de domaine ont une structure de graphe, on peut exprimer une grande variété de règles d'inférence sous forme d'expressions de chemins, comme nous le verrons ci-dessous.

2.3 Modèle d'interface

L'interface de lecture d'un hyperlivre est un hypertexte formé de nœuds et de liens hypertextuels dérivés à partir de la monographie suivant une spécification. Le modèle possède essentiellement deux catégories de liens hypertextuels : les liens de référence (liens hypertextuels habituels) et les liens d'inclusion qui permettent de construire des documents hiérarchiques complexes en incluant des nœuds les uns dans les autres.

Une spécification d'interface est composée d'un ensemble de schémas de nœuds qui peuvent être instanciés pour produire les nœuds et liens réels de l'interface. Un schéma de nœud est composé

- d'une sélection (qui définit quels objets de la monographie choisir pour constituer le nœud)
- d'une définition de contenu (quels attributs utiliser et comment les arranger dans le nœud)
- de spécifications de liens de référence et d'inclusion

La composition des documents d'interface est donc spécifiée à la fois par les relations existant dans l'hyperlivre (partie structurelle de la composition) et par les règles de sélection et d'assemblage exprimées dans les schémas de nœuds (partie dynamique et inférentielle).

La sélection des fragments à assembler se fait en évaluant une ou plusieurs expressions de chemin, qui fournissent des n-uplets d'objets appartenant aux classes de la structure d'hyperlivre, puis en sélectionnant parmi ces n-uplets ceux qui satisfont une condition globale.

Une expression de chemin est une séquence $E_1 \dots E_n$ où chaque E_i est un élément de chemin qui peut prendre l'une des formes suivantes :

$C_i v_i [s_i]$
 $-(C_i v_i [s_i]) \rightarrow$
 $\leftarrow (C_i v_i [s_i]) -$

où C_i est un nom de classe de la structure d'hyperlivre, v_i est un nom de variable et s_i est une expression booléenne (éventuellement vide) sur les attributs de la classe C_i . Pour les formes $-(\dots) \rightarrow$ et $\leftarrow (\dots) -$ C_i doit être l'une des classes associatives (*O-F-Link*, *F-F-Link* ou *Relation*)

Par exemple :

Concept $c \rightarrow (O-F-Link \ k \ [type="exemple"]) \rightarrow$
 Fragment f

L'interprétation de cette expression est composée de tous les triplets (c, k, f) tels que c est une instance de la classe *Concept*, k est une instance de *O-F-Link*, f est une instance de *Fragment*, k lie c à f ($k.from = c$ et $k.to = f$) et k est de type *exemple*.

Formellement, une assignation $(v_1=x_1, \dots, v_n=x_n)$ satisfait une expression de chemin $E = E_1 \dots E_n$ sur un hyperlivre si

- x_i est un objet de la classe C_i ($i=1, n$)
- pour chaque E_j de la forme $-(C_j v_j) \rightarrow$ on a $x_j.from = x_{j-1}$ et $x_j.to = x_{j+1}$ (l'objet x_j associe x_{j-1} à x_{j+1})
- pour chaque E_j de la forme $\leftarrow (C_j v_j) -$ on a $x_j.to = x_{j-1}$ et $x_j.from = x_{j+1}$

Etant donné qu'il n'existe qu'une association entre deux classes dans le schéma, on peut sans ambiguïté omettre les noms d'associations. De plus on utilisera le raccourci syntaxique

$C_1 v_1 \rightarrow "NomType" \rightarrow C_2 v_2$

pour signifier

$C_1 v_1 \rightarrow (A \ v_{1,2} \ [type="NomType"]) \rightarrow C_2 v_2$

où A est l'unique association existant entre C_1 et C_2 .

Par exemple, le chemin suivant associe un fragment f à un fragment g s'ils sont tous les deux des exemples d'un même concept c .

Fragment $f \leftarrow ("exemple") -$ Concept c
 $-("exemple") \rightarrow$ Fragment g

Le chemin ci-dessous associe quand à lui f à g si g est un exemple de c ou d'un sous-concept de c (la notation $*$ représente la clôture transitive d'un lien)

Fragment $f \leftarrow ("exemple") -$ Concept $c \leftarrow "est-un" -$ Concept $d \rightarrow ("exemple") \rightarrow$ Fragment g

Nous pouvons maintenant présenter, à l'aide d'exemples, la syntaxe des schémas de nœuds.

Exemple 1. Le schéma suivant sélectionne tous les fragments liés à un concept C donné, le contenu d'une instance est formé

- d'un élément `<title>` qui contient le terme principal associé à C
- pour chaque fragment F lié à C par un lien L : le type du lien, le titre et le contenu du fragment.

```
node fragments_lies[C]
  <title> "Fragments liés à ", C.term
```

```

</title> ,
{<subtitle> L.type, ": ", F.title
</subtitle>
<text> F.content </text>
}
from Concept C -(O-F-Link L)-> Fragment F

```

La présentation finale d'un nœud instance de ce schéma sera déterminée par des feuilles de style XSLT et CSS.

Exemple 2. Cet exemple montre comment spécifier un assemblage de fragments de manière à obtenir un document d'interface sémantiquement consistant et obéissant à un objectif pédagogique. Les schémas de nœud suivants spécifient un document (les marques d'éléments XML ont été omises) qui présente

- le contenu d'un fragment *f*
- des liens vers tous les concepts directement liés à celui-ci
- des liens d'expansion vers les concepts nécessaires à la compréhension de ce fragment

Les liens d'expansions permettent d'inclure le contenu d'un nœud dans un autre à la demande (quand l'ancre est activée).

```

node fragment_et_contexte[f]
  f.content ,
  include concepts_liés[f] ,
  include concepts_nécessaires[f]
from

node concepts_nécessaires[f]
  { expand href def_concept[c](c.term) }
from Fragment f <-"utilise"- Concept c

node def_concept[c]
  c.term ,
  { include fragment_et_contexte[f] }
  href fragments_liés[c] ("fragments liés") ,
from Concept c <-"definition"-> Fragment f

node concepts_liés[f]
  { href def_concept[c](l.type c.term) }
from Concept c
-(O-F-Link l [type /= "utilise"])-> Fragment f

```

Cet exemple illustre la construction par "chaînage arrière" d'un document. Le lecteur utilise l'expansion de liens pour faire apparaître les définitions nécessaires à sa compréhension tant qu'il n'arrive pas sur des concepts connus.

3 Une approche multi-point de vue de l'intégration

L'objectif étant de créer des documents d'interfaces pour lire non seulement un document virtuel de la bibliothèque mais pour accéder à toute la bibliothèque, on pourrait choisir une approche très directe consistant à intégrer tous les hyperlivres en un seul grand hyperlivre. Dans ce cas il faudrait créer une ontologie de domaine globale à partir

des ontologies de chaque hyperlivre. Cette approche est cependant très limitative car

- elle force à unifier fortement les concepts, ce qui ne pose pas de problème pour la terminologie bien établie d'un domaine mais qui peut être problématique lorsque les concepts ont des contours flous ou lorsqu'il subsiste des interprétations divergentes ou contradictoires.
- elle ne reflète pas le fait que chaque livre représente le point de vue d'un auteur sur un sujet (on perdrait cette diversité de points de vue)
- elle fait perdre la diversité des styles narratifs (reflétés dans les documents d'interface) adoptés par les différents auteurs.

C'est pourquoi nous proposons une approche basée sur la notion d'ontologie multi-point de vue (Falquet & Mottaz Jiang 2000, 2001).

3.1 Concepts conflits et points de vue

Gaines et Shaw (1989) proposent une méthodologie pour comparer les systèmes conceptuels de plusieurs experts de domaine. Cette méthode est basée sur l'analyse des entités (concepts) du domaine, des termes utilisés pour les désigner, et de leurs attributs. Le but est de mettre en évidence les divergences entre experts afin de faciliter la discussion pour atteindre un consensus. L'analyse peut déboucher sur quatre situations différentes :

- consensus : les experts utilisent le même terme pour désigner le même concept
- correspondance : plusieurs termes différents désignent le même concept
- conflit : le même terme est utilisé pour des concepts différents
- contraste : les experts ont identifié des concepts différents et utilisent des termes différents pour les nommer.

Le conflit représente la situation la plus délicate à traiter dans le cadre de l'intégration. En effet, une situation de correspondance peut simplement signifier qu'il existe des termes synonymes pour un concept. Quant au contraste, il peut indiquer que les experts ont des vues partielles, disjointes et peut-être complémentaires du domaine.

Dans une approche multi-point de vue, la résolution des conflits s'effectue en considérant qu'il peut exister plusieurs concepts associés au même terme, pour autant que ceux-ci appartiennent à des points de vue différents.

Dans le cas de l'intégration d'ontologies appartenant à différents hyperlivres, nous considérerons que chaque hyperlivre correspond à un point de vue différent. On trouvera alors les trois situations suivantes :

1. deux concepts désignés par le même terme n'appartiennent pas au même domaine sémantique (p.ex. le concept table du domaine des meubles et le concept table du domaine des base de donnée).
2. les deux concepts appartiennent bien au même domaine mais ont des définitions différentes.. Les deux définitions représentent alors des points de vue différents sur ce concept.

3. les définitions des deux concepts sont jugées équivalentes. Les points de vue des deux livres coïncident pour ce concept.

3.2 Intégration des ontologies pour les documents virtuels

Etant donné que l'objectif de l'intégration est d'arriver à une ontologie multi-point de vue et non pas à une seule ontologie "monolithique", les techniques d'intégration les plus appropriées dans notre cas sont celles qui établissent des liens entre concepts (alignement d'ontologies).

La technique doit également être capable d'effectuer des calculs de similarité sémantique pour décider si deux concepts appartiennent au même champ sémantique ou non, ensuite elle doit permettre une comparaison fine des définitions de concepts pour décider si la similarité est suffisante pour qu'on puisse établir une relation d'équivalence (même point de vue) ou si l'on doit en rester à la similitude (différents points de vue sur un concept).

Pour cela, nous proposons d'utiliser une extension de la technique de Rodriguez et Egenhofer. (2003). Dans cette technique, la similarité entre deux concepts est la somme pondérée de trois mesures : similarité des termes (ensembles de synonymes) ; similarité des attributs (ensembles de valeurs) et similarité des voisinages sémantiques (ensembles des concepts proches dans le graphe des liens sémantiques). Le calcul de la similarité de ces ensembles tient de plus compte de la différence de profondeur des concepts dans leurs ontologies respectives.

Dans le cas du document virtuel, nous disposons d'une information supplémentaire pour évaluer la similarité entre concepts grâce aux fragments liés à chaque concept. Si deux concepts A et B sont liés par des liens du même type t à des ensembles de fragments $t(A)$ et $t(B)$ respectivement, la similarité « documentaire » entre $t(A)$ et $t(B)$ sera prise en compte dans le calcul de la similarité entre A et B . Pour calculer la similarité entre $t(A)$ et $t(B)$ on utilisera une technique classique de recherche d'information (par exemple, le calcul du cosinus entre les vecteurs *tf-idf* représentant les documents dans l'espace des termes (Salton, 1989) ou la distance de Kolmogorov). Puis on définit la similarité entre $t(A)$ et $t(B)$ à l'aide des similarités entre documents (par exemple en prenant la similarité maximale trouvée entre tous les fragments de $t(A)$ et de $t(B)$).

Les résultats des calculs de similarités pour tous les types de liens t viendront s'ajouter à la mesure de similarité calculée au niveau purement conceptuel.

On notera ici l'importance du typage des liens entre concepts et fragments. En effet, la comparaison n'a de sens que si les fragments comparés jouent le même rôle par rapport aux concepts. Si, par exemple, le fragment a est un exemple du concept A alors que b est un contre-exemple de B , une forte similarité entre a et b n'implique pas une forte similarité entre A et B , au contraire.

Au niveau de l'ontologie de domaine, ces calculs se traduisent par l'introduction de deux types de relations entre concepts : "equiv" et "sim" pour représenter l'équivalence (dans le même point de vue) et la similarité (concepts du même domaine sémantique mais avec des définitions différentes). De plus on introduira dans chaque classe un attribut *hybook* qui permettra d'identifier à quel hyperlivre appartient un fragment, un concept, ou un lien.

4 Génération de documents d'interface pour la bibliothèque

Une caractéristique intéressante du modèle d'hyperlivre virtuel et du modèle d'intégration que nous proposons est la possibilité de réutiliser les spécifications d'interfaces des documents virtuels pour créer des interfaces de lecture globale.

4.1 Réutilisation des interfaces

Une première technique de construction d'une interface globale consiste à réutiliser la spécification d'interface d'un hyperlivre mais en l'appliquant sur l'ensemble des fragments de la bibliothèque et en utilisant les liens d'équivalence et de similarité entre concepts. Si l'on considère qu'un hyperlivre représente un point de vue (sémantique et narratif), on obtiendra une vision de toute la bibliothèque selon le point de vue de cet hyperlivre. En d'autre termes, on étend un hyperlivre à l'aide des autres.

La manière la plus directe d'étendre un hyperlivre L consiste à utiliser les concepts équivalents trouvés dans les autres hyperlivres et les liens issus de ces concepts. Il suffit pour cela d'étendre les expressions de chemins dans les schémas de nœuds de L de la manière suivante :

Tout élément de chemin de la forme

Concept c

est remplacé par

Concept c [hybook = L] - ("equiv") -> Concept c' .

Donc tout chemin qui passe par un concept c du livre L est étendu de manière à passer par tout concept c' équivalent (appartenant au même point de vue). Par exemple, le chemin

```
Fragment f <- "exemple" - Concept c - "exemple" ->
Fragment g
```

devient

```
Fragment f <- "exemple" - Concept c - "equiv" ->
Concept c' - "exemple" -> Fragment g
```

Un fragment f qui contient un exemple pour le concept c sera lié à un fragment g si g est un exemple d'un concept équivalent à c (l'ensemble des exemples d'un livre s'étend grâce aux exemples trouvés dans d'autres livres).

Il faut noter que cette approche est d'autant plus efficace que les ontologies d'hyperlivres sont bien intégrées, c'est-à-dire qu'on trouve les mêmes types de liens et catégories de fragments dans ces ontologies. Ce dernier problème, bien que non trivial, est cependant plus simple que l'inté-

gration des ontologies de domaine car le nombre de concepts en jeu est bien moindre.

La relation de similarité permet, quand à elle, d'étendre un livre en dérivant de nouveaux liens mais avec un changement de point de vue. Le livre de départ s'enrichit alors d'autres points de vue sur le sujet. Dans ce cas, on ne peut pas simplement réutiliser les schémas de nœud en modifiant les expressions de chemin. Pour chaque élément de chemin de la forme

Concept c

remplacé par

Concept c [hybook = L] - ("sim") -> Concept c'

il faut créer un nouveau schéma de nœud pour séparer le point de vue original du livre des autres points de vue. Par exemple, le schéma de nœud `def_concept` de l'exemple 2 (section 3) devient

```
node def_concept_autres[c]
  c.term ,
  { include fragment_et_contexte[f] }
  href fragments_liés[c] ("fragments liés") ,
  from Concept c -"sim"-> Concept c2 -
  "definition"-> Fragment f
```

Il permet de présenter d'autres définitions du concept c correspondant aux autres points de vue présents dans la bibliothèque.

Une autre technique de réutilisation consiste à appliquer la spécification d'interface d'un hyperlivre à un autre. Dans ce cas, on verra le contenu informationnel d'un hyperlivre avec l'interface d'un autre. Dans ce cas, c'est la partie dynamique de la narration d'un livre qui est appliquée à un autre contenu. Cette réutilisation ne nécessite aucune réécriture des schémas de nœuds, mais elle implique que les ontologies d'hyperlivre aient été bien intégrées.

4.2 Création de nouveaux livres "globaux"

Dans ce cas, on suppose qu'un auteur veut créer un nouveau livre à partir de l'information existant dans la bibliothèque. Il s'agit d'un auteur de 2^e niveau qui ne va pas créer de l'information mais inventer de nouvelles narrations et présentations. Cette tâche peut s'accomplir soit en créant de nouveaux schémas de nœuds, soit en réutilisant des schémas provenant de différents hyperlivres. Comme nous l'avons déjà vu, chaque schéma de nœud peut être appliqué à n'importe quel hyperlivre, il en résulte qu'un auteur de 2^e niveau peut créer de nouveaux schémas qui incluent ou font référence à des schémas existants, sans avoir à modifier ces derniers.

5 Implémentation

Nous avons réalisé diverses implémentations d'hyperlivres, en utilisant la techniques des vues hypertextuelles sur les bases de données. L'ontologie de domaine, l'ontologie d'hyperlivre et les fragments sont stockés dans une base de données relationnelle (le diagramme de classes de la figure 3 se traduit facilement en schéma relationnel). Les schémas de nœuds d'interface sont spécifiés dans le

langage Lazy qui est un langage déclaratif pour la spécification de vues hypertextuelles sur les bases relationnelles. Ce langage correspond au langage présenté dans cet article pour la définition des schémas de nœuds d'interface. La principale différence se situe au niveau des expressions de sélection qui, dans Lazy, correspondent au calcul relationnel. On peut cependant aisément traduire les expressions de chemin sous forme d'expressions relationnelles (cette traduction peut être automatisée). Le système Lazy possède un compilateur de schémas de nœuds et un serveur de nœuds (une servlet Java). Ce dernier instancie les nœuds dynamiquement, il interroge la base de donnée, effectue les inclusions et les expansions et produit le contenu XML ou HTML demandé par le lecteur. La figure ci-dessous montre une instance de nœuds d'un hyperlivre. On y reconnaît une définition de concept, suivie de fragments correspondant à ce concept (commentaires et exercices) et une liste de liens vers des concepts liés.

Automate à états fini

Un AF est composé de

- un alphabet de symboles reconnus $A = \{s_1, s_2, \dots, s_n\}$
- un ensemble d'états $Q = \{q_1, q_2, \dots, q_m\}$ (les ronds)
- un état initial q_1 de Q
- un ensemble F d'états finals inclus dans Q
- une fonction de transition d de $Q \times A$ dans Q (les flèches)

Commentaire

Un automate fini est déterministe, intuitivement s'il ne peut se comporter que d'une façon pour un mot donné : étant donné un état de départ et un caractère lu, il y a au plus 1 état d'arrivée

Dans le cas d'un **automate déterministe**, si un mot est accepté par le langage, alors il existe une seule dérivation possible pour ce mot. En revanche, si un automate n'est pas déterministe, il peut y avoir des configurations pour lesquelles plusieurs transitions différentes sont possibles.

Exercice

Exercice : On veut accepter le langage forme de a , de b et de c , dans cet ordre
(p.ex : aaabcccc, abbc, aaaabc, ...)

Concepts liés

- [Acceptation d'une chaîne par un AEF \(algorithme\)](#)
- [Acceptation d'une chaîne par un AEF \(math\)](#)
- [Automate à états fini non-déterministe](#)
- [Chaîne de symboles](#)
- [Langage régulier](#)
- [Machine de Turing](#)
- [State Chart](#)

Fig. 3. Nœud d'un hyperlivre

6 Conclusion et travaux futurs

Nous avons présenté d'une part un modèle d'hyperlivres virtuels basé sur des ontologies et d'autre part une approche de l'intégration d'ontologies adaptée à la lecture d'hyperlivres dans une bibliothèque numérique. Cette approche est basée sur l'idée que chaque hyperlivre correspond à un point de vue sur une domaine. En enrichissant des techniques d'intégration d'ontologie connues on peut construire une ontologie multi-point de vue qui décrit un ensemble d'hyperlivres. Cette ontologie peut ensuite être utilisée par un langage de spécification d'interfaces hypertextuelles pour inférer des liens sémantiques entre fragments d'information et construire des documents d'interface.

L'implémentation que nous avons réalisée nous permet de tester différents designs d'interfaces de lecture, en particulier au niveau des inférences de liens. Nous allons également tester la technique d'intégration proposée et en particulier déterminer l'influence des valeurs des différents

paramètres qui apparaissent dans la technique d'intégration.

Bibliographie

- Ackscyn R., McCracken D., Yoder E., "KMS: a distributed hypermedia system for managing knowledge in organizations". *Communications of the ACM*, vol. 31, July 1988, p. 820-835.
- Bernstein M., "Storyspace". *Proceedings of the thirteenth Conference on Hypertext and Hypermedia*, 2002, College Park, Maryland, 2002, p. 172-181.
- Brusilovsky P., Rizzo R., "Map-Based Horizontal Navigation in Educational Hypertext". *ACM Hypertext 2002 Conference*, College Park, Maryland, 2002.
- Brusilovsky, P., "Methods and techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction". In *Adaptive Hypertext and Hypermedia*, Kluwer, 1998, pp. 1-43.
- Crampes, M., Ranwez, S. "Ontology-Supported and Ontology-Driven Conceptual Navigation on the World-Wide Web". In *Proc. ACM Hypertext 2000*, San Antonio, USA, 2000.
- Crampes M., Vercoustre A. M., Nanard M., Ranwez S. (Eds), "Acte de l'atelier Documents Virtuels Personnalises : De la Définition à l'Utilisation", atelier de la 11ème Conférence Francophone sur l'Interaction Homme-Machine, Montpellier, novembre 1999.
- De Bra P., Calvi L., "Creating Adaptive Hyperdocuments for and on the Web". *Proceedings of the AACE Web-Net Conference*, Toronto, 1997, 149-155.
- Falquet, G., Mottaz Jiang, C.-L. "Conflict Resolution in the Collaborative Design of Terminological Knowledge Bases" In *Proc. EKAW 2000 Conference*, Juan-les-Pins, 2000.
- Falquet, G., Mottaz Jiang, C.-L. "Navigation hypertexte dans une ontologie multi-point de vue" In *Actes de la conférence NîmesTIC 2001*, Nîmes, 2001.
- Falquet G., Ziswiler, J.-C. "A Virtual Hyperbooks Model to Support Collaborative Learning" to appear in *AIED 2003 Supplemental Proceedings*, Sydney, Australia, July 2003.
- Farquhar, A., Fikes, R., Rice, J. "The Ontolingua Server: A Tool for Collaborative Ontology Construction". In: *Knowledge Acquisition Workshop*, 1996.
- Farquhar, A., Fikes, R., Rice, J. "Tools for Assembling Modular Ontologies in Ontolingua". In: *Proceedings of the AAAT'97 Conference*, p. 436-441, 1997.
- Fröhlich P., Nejd W., "A Database-Oriented Approach to the Design of Educational Hyperbooks". *Proceedings of the Workshop "Intelligent Educational Systems on the World Wide Web"*, 8th World Conference of the AIED Society, Kobe, Japan, 1997, p. 18-22.
- Gaines, M., Shaw, B. "Comparing Conceptual Structures: Consensus, Conflict, Correspondence and Contrast". *Knowledge Science Institute, University of Calgary*, 1989.
- Garlatti, S., Crampes, M. *Actes de la conférence Documents virtuels personnalisables (DVP'02)*, Brest, 2002.
- Garret L. N., Smith K. E., Meyrowitz N., "Intermedia: Issues Strategies and Tactics in the Design of a Hypermedia Document System". *CSCW '86 Proceedings*, Austin, Texas, Dec 1986.
- Gruber, T. R. "Ontolingua: A mechanism to Support Portable Ontologies". *KSL 91-66*. Stanford University, Knowledge Systems Laboratory.
- Halasz F., Schwartz M., "The Dexter hypertext reference model". *Communications of the ACM*, vol. 37, no 2, 1994, p. 30-39.
- Iksal, S., "Spécification déclarative et composition sémantique pour des documents virtuels personnalisables". PhD thesis, ENST Bretagne, Brest, 2002.
- Iksal, S., Garlatti, S., "Spécification déclarative pour des documents virtuels personnalisables". In: Garlatti S., Crampes M. (eds.), "Actes du congrès Documents virtuels personnalisables (DVP 2002)". Brest, 2002.
- Landoni M., Crestani F., Melucci M., "The Visual Book and the Hyper-Textbook: Two Electronic Books One Lesson?" *RIAO Conference Proceedings*, 2000, p. 247-265.
- Nanard J., Nanard M., "Should Anchors be Typed Too? An Experiment with MacWeb". In *Proc. ACM Hypertext '93 Conference*, 1993, p. 51-62.
- Pinto, H. S., Gómez-Pérez, A., Martins, J. P. "Some Issues on Ontology Integration". In V. R. Benjamins, B. Chandrasekaran, A. Gómez-Pérez, N. Guarino, M. Uschold (eds.). *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, Stockholm, Sweden, August 2, 1999.
- Preece, A. D. et al., "The KRAFT architecture for Knowledge Fusion and Transformation". In: *Proceedings of the 19th SGES International Conference on Knowledge-based Systems and Applied Artificial Intelligence (ES'99)*, Springer Verlag, 2000.
- Rada R., "Hypertext writing and document reuse: the role of a semantic net", *Electronic Publishing*, vol. 3, no. 3, 1990, p. 125-140.
- Ranwez S., Crampes M., "Ontology-supported and ontology-driven conceptual navigation on the World Wide Web" in *Proceedings of the ACM Hypertext and Hypermedia Conference*, San Antonio, 2000.
- Rodríguez M. A., Egenhofer M. J. "Determining Semantic Similarity among Entity Class from Different Ontologies" in *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 2, March/April 2003.
- Salton, G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.
- Wu H., de Kort E., De Bra P., "Design Issues for General-Purpose Adaptive Hypermedia Systems", *ACM Hypertext 2001 Conference*, Aarhus, 2001, p. 141-150.