

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
République Algérienne Démocratique et Populaire

Université Larbi Ben Møhidi d'Oum El Bouaghi

Faculté des Sciences et de la Technologie

# MÉMOIRE

Présenté en vue de l'obtention du diplôme de  
**Magister en informatique**  
Système : Ecole Doctorale en Informatique de l'Est  
Pôle Constantine  
Option : GÉNIE LOGICIEL

SUJET DU MÉMOIRE :

---

Conception d'applications Hypermédia à base de  
Design patterns

---

Présenté par : HAMZA Lamia

Sous la direction du professeur : BOUFAIDA Mahmoud

Soutenu le 10 Juillet 2010 devant le jury composé de :

**Président :**

ZAROOR Nacereddine, Professeur à l'Université Mentouri de Constantine.

**Rapporteur :**

BOUFAIDA Mahmoud, Professeur à l'Université Mentouri de Constantine.

**Examineurs :**

MAAMRI Ramdane, Maître de conférence à l'Université Mentouri de Constantine.

MOKHATI Farid, Maître de conférence à l'Université Larbi Ben Møhidi d'Oum El Bouaghi.



## ***Remerciements :***

*Ce travail n'aurait pu être mené jusqu'à son terme sans la bienveillance de mon encadreur de thèse, Monsieur Mahmoud BOUFAIDA, professeur de l'Université de MENTOURI de Constantine, qui m'a toujours accordé sa confiance. Je le remercie profondément pour ses conseils, ses encouragements et son soutien constant.*

*Mes remerciements s'adressent aussi aux membres de jury pour avoir accepté de discuter cette thèse.*

*Je remercie aussi mes éminents enseignants qui m'ont permis de pénétrer le monde informatique.*

*C'est un énorme remerciement que j'adresse maintenant à toute ma famille : mon père, ma mère, ma petite jolie sœur Karima (Merci Karoma), mes frères Fares et Salah.*

*Enfin, je remercie mon mari Zouhair : dans mon esprit tu es celle qui occupe la première place. Un énorme merci pour ton amour et ton soutien.*

# Table des matières

## Introduction générale.

1. Contexte de travail	01
2. Problématique et objectifs	02
3. Plan de travail	03

## Chapitre 1 : Caractéristiques des applications hypermédia.

1. Introduction	04
2. Historique	05
3. Quelques définitions	05
3.1 Hypertexte	06
3.2 Hypermédia	06
3.3 Hyperdocument	06
3.4 Relation entre hypertexte, hypermédia et hyperdocument	07
4. Composants d'une application hypermédia	07
4.1 Noeud	07
4.2 Lien	08
4.3 Ancre..	09
5. Notion et formes de navigation	10
5.1 Notion de navigation	10
5.2 Formes de navigation	11
5.2.1 Cheminement	11
5.2.2 Tour guidé	11
6. Structures des documents hypermédia	11
6.1 Hypermédia Linéaire Tourne Page	11
6.2 Hypermédia Arborescent	12
6.3 Hypermédia Combinatoire	12
6.4 Hypermédia en Etoile	12
6.5 Hypermédia en Maille de Filet	12
7. Types d'usage de documents hypermédia	12









# Chapitre 1

## Caractéristiques d'applications hypermédia

### 1. Introduction :

Les applications hypermédia connaissent une large diffusion après l'évolution des technologies Web, notamment dans les milieux culturels, les bibliothèques numériques, le commerce électronique aussi les EIAH (*Environnements Informatiques et Apprentissage Humain*) [Educnet, 2009]. Les possibilités de navigation et la diversité du contenu font de l'hypermédia un moyen privilégié pour s'adresser à un public varié. La conception des applications hypermédia s'apparente aux autres formes de conception. Elle consiste en la réalisation d'un produit conçu par un collectif d'acteurs mettant en œuvre des méthodes et modèles. L'identification de l'information pertinente nécessaire aux différents contextes de lecture, la mise en place des liens, la réalisation des mécanismes de navigation, et l'étude de l'ergonomie et des interactions demeurent les éléments critiques à étudier dans le développement des applications hypermédia.

Mais un système hypermédia c'est aussi un logiciel qui possède des fonctions spécifiques destinées à des utilisateurs identifiés. Ces fonctions seront réalisées avec des outils informatiques connus. L'utilisation de méthodes de conception devient alors nécessaire. Cependant les spécificités des acteurs, des techniques et de l'audience des applications hypermédia nécessitent une adaptation des méthodes existantes. Plusieurs méthodes ont été proposées pour concevoir un hypermédia de manière incrémentale par la définition d'un ensemble de modèles. Face à la prédominance des applications Web basées sur la notion d'hypermédia, certaines méthodes ont évolué, de nouvelles sont apparues avec pour objectif la prise en compte des architectures Web et/ou l'utilisation du langage UML (*Unified Modeling Language*)<sup>1</sup>.

Dans ce chapitre, nous nous intéressons à la présentation des caractéristiques principales des applications hypermédia. Nous définissons quelques termes utilisés dans ce domaine ainsi nous présentons les notions de navigation et d'orientation, et nous essayons dans ce chapitre de faire une synthèse sur les méthodes de construction des applications hypermédia déjà proposées dans la littérature.

---

<sup>1</sup>UML est un langage permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet.



Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

L'idée de l'hypermédia a été énoncée en 1945 par l'américain Vannevar Bush [Bush, 1945]. Il désirait créer une machine multimédia à base de microfilms, dans laquelle il aurait pu stocker toute sorte de documents, aussi bien des livres que des enregistrements. V. Bush signalait dans son article "*As we may think*", que le cerveau humain fonctionne par associations et proposait de mettre en œuvre le même processus pour la gestion des documents de son système appelé "Memex" (*MEMory EXtender*). Le Memex n'a jamais vu le jour faute de technologie appropriée, mais Bush avait défini des notions importantes dans le domaine de l'hypermédia, celles de lien et de noeud.

Douglas Englobart reprenait les réflexions de V. Bush dans ces travaux. En 1968, il définit une première mise en œuvre effective d'un hypertexte rudimentaire dans son système NLS (*oN Ligne System*) [Englebart, 1984]. Dans le même domaine, Engelbart veut amplifier ou augmenter l'intelligence humaine, ce qui suggéra le titre de son deuxième projet "Augment", développé au centre de recherche pour l'augmentation de l'intellect humain qu'il a fondé à Stanford. Le projet "Augment" est un environnement en réseau, de traitement de textes et de traitement d'idées qui permet la collecte des documents, des notes et des rapports de recherche tout en fournissant des moyens de planification, d'analyse, de résolution de problèmes et de communication. Engelbart fournit donc les premiers outils de l'hypertexte, qui, selon ses vœux, ne limitent ni ne contraignent les gens les plus habiles. Il souhaite ainsi encourager la performance et l'excellence.

Au début les années 1970, Ted Nelson conçut un réseau, accessible en ligne, en temps réel, qui permettrait à chacun de lire, écrire, commenter, relier des documents de tous types. L'idée sur laquelle repose ce système, nommé "Xanadu", est de disposer d'un fonds commun rassemblant les documents de l'humanité, « *the magic place of literary memory where nothing is forgotten* » [Nelson, 1970]. Tout le monde pourrait y accéder, y insérer des documents et y établir des liens.

Avec V. Bush, D. Engelbart et T. Nelson, les assises historiques de l'hypertexte sont complétées. Mais ce sont-là trois perspectives différentes. Bush propose l'approche analogique, « *si nous pensons par associations, nous devons construire nos connaissances par associations et nous devons nous donner des outils qui travaillent dans ce style, ce qui est impossible sans ordinateur* » [Bush, 1945]. Nelson va rester le visionnaire qui a popularisé le concept et forgé le terme "hypertexte". Pratiquement, son encyclopédie universelle est un immense réservoir d'informations muni de mécanismes de repérage efficaces. Engelbart est un inventeur d'outils d'interfaces qui propose des environnements de travail en collaboration ou en réseaux, ce qui tend à augmenter, selon sa vision, les capacités intellectuelles des plus doués. Pour cet ingénieur, la confiance est d'abord dans l'outil.

### 3. Quelques définitions :

Dans cette section, nous présentons les définitions de quelques termes les plus utilisés dans le domaine hypermédia et nous essayons de préciser la différence entre eux.



Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

Le terme "hypertexte" a été introduit par T. Nelson comme suit: « *permettez-moi d'introduire le mot hypertexte pour signifier un corpus composé de matériels écrits ou graphiques interconnectés d'une manière tellement complexe qu'il ne serait pas pertinent de le représenter sur papier. Il peut contenir des sommaires, ou des cartes de son contenu et de ses interrelations ; il peut contenir des annotations, des additions et des notes de bas de page insérés par les chercheurs qui l'ont examiné* » [Nelson, 1965].

Selon cette définition, la notion d'hypertexte désigne une organisation permettant la lecture non linéaire des documents à l'initiative du lecteur<sup>2</sup>. Nous remarquons le mot "texte" dans le terme hypertexte, ce mot désigne que l'hypertexte ne contient que des textes. Mais dans la communauté scientifique, la notion d'hypertexte n'est pas limitée au seul aspect textuel des documents, elle concerne toutes les formes de documents, y compris bien entendu les formes multimédia. C'est pour cela que le terme "hyperdocument" serait le mieux adapté en français pour traduire cette notion, mais il n'est que rarement employé [Nanard, 1994].

### 3.2 Hypermédia :

Le terme "hypermédia" a été introduit, surtout par la presse, pour insister sur l'aspect multimédia de certaines applications hypertextes [Nanard, 1994]. L'hypermédia à plusieurs définitions, il a été défini par Quarterons Phillippe comme : « *étant un document informatisé composé des noeuds reliés entre eux par des lien. La nature de ces noeuds peut être aussi bien textuelle, que visuelle, sonore ou encore audio visuelle* » [Phillipe, 1996]. L'hypermédia a été également défini par M. Nanard comme suit : « *un hypermédia est un agent qui aide des humains dans une tâche de lecture active, en apportant l'information pertinente selon le contexte de lecture passé et présent* » [Nanard, 1998].

La différence principale entre l'hypertexte et l'hypermédia réside du côté contenu. Pour l'hypermédia le contenu n'est pas seulement des textes, mais peut être composé de plusieurs médias, tels que des images, des séquences animées, des vidéos, des sons, etc.

### 3.3 Hyperdocument :

Le terme "hyperdocument" a été défini par Jean-Pierre Balpe comme suit : « *tout contenu informatif informatisé dont la caractéristique principale est de ne pas être assujetti à une lecture préalablement définie mais de permettre un ensemble plus ou moins complexe, plus ou moins divers, plus ou moins personnalisé de lectures* » [Balpe, 1990]. Selon cette définition, le domaine des hyperdocuments comprend plusieurs produits, comme par exemple : les bases de données, les hypertextes, les hypermédias.

---

<sup>2</sup> Un hypermédia est avant tout un outil interactif destiné à un utilisateur nommé lecteur [Nanard, 1998].

**Applications hypermédia et hyperdocument :**

En se concentrant sur les hypertextes, J. Balpe [Balpe, 1996] précise leurs différentes catégories : livres électroniques, stations de travail, postes de lecture assistée par ordinateur, multimédias, documents multimédias interactifs et hypermédiats. Une autre catégorie, les réseaux sémantiques, peut s'y ajouter à condition qu'elle soit comprise dans un hypertexte ou représentée sous forme hypertextuelle : « *un hypertexte peut éventuellement contenir un réseau sémantique décrivant les relations entre les nœuds, spécifiant les relations entre les types d'informations disponibles. Ce réseau sémantique peut être utile pour effectuer des recherches intelligentes, mais l'hypertexte n'est pas identifiable à ce réseau sémantique et un réseau sémantique n'est pas non plus un hypertexte* » [Balpe, 1996].

À partir de ce que nous avons présenté, nous pouvons conclure que l'hyperdocument est une catégorie plus large qui peut contenir des hypertextes et des hypermédiats. L'hypermédia repose donc sur la technologie de l'hypertexte et son organisation à base d'hyperliens. Toutefois, plusieurs auteurs<sup>3</sup> utilisent indifféremment les trois concepts, "hyperdocument", "hypertexte" et "hypermédia" pour montrer que le contenu ne représente pas l'intérêt majeur des systèmes hypermédiats, c'est plutôt l'hyperespace et sa structure qui intéressent les utilisateurs.

Dans notre mémoire, nous nous sommes intéressés par les applications hypermédia, nous conservons le terme "hypermédia" dans les sections suivantes de ce chapitre et dans les chapitres suivants de ce mémoire.

**4. Composants d'une application hypermédia :**

Comme nous avons vu dans la section 3, une application hypermédia est un ensemble de nœuds reliés entre eux par des liens. À partir de cette définition, nous pouvons induire les composants de base d'une application hypermédia : les noeuds, les liens qui relient ces nœuds et les ancres qui permettent d'activer les liens. Dans ce qui suit, nous précisons chacun de ces concepts.

**4.1 Noeud :**

Le noeud est l'unité minimale d'information dans un hypermédia. Si nous nous référons à diverses théories cognitives [Stash, 2007], nous parlons aussi de "bloc", "chunk", "îlot" ou de "frame" et "script". Dans un noeud, l'information est modularisée, dans un texte, elle est linéarisée. La granularité d'un nœud désigne la quantité d'information dans ce noeud. La multiplication des noeuds ou la fragmentation de l'information peut engendrer une désorientation et une confusion. Chaque noeud comprend idéalement une seule "idée", "concept", ou "sujet" qui peut s'accrocher à d'autres (par des liens) qui lui sont naturellement connexes ou à d'autres qui dépendent du choix de l'utilisateur. Les noeuds connexes peuvent être des exemples, des élaborations ou des idées nouvelles. Toute séquence ou tout agencement de noeuds est donc possible.

<sup>3</sup> L'auteur : est le concepteur de document hypermédia.



Une information peut être une page, un écran, une carte, une partie d'écran appelée fenêtre, si l'information est textuelle. Si elle n'est pas uniquement textuelle, le support d'un noeud peut être un graphique, une animation, une image, une séquence de vidéo ou d'audio ou un autre élément externe comme une maquette, etc. L'information contenue dans un noeud peut être modifiée dans la plupart du temps. Les noeuds d'information peuvent être de divers types: définitions, attributs, références, notes, illustrations, exemples, etc. Ces types de noeuds sont souvent identifiés visuellement par un titre, une icône, etc. [Rhéaume, 2009].

Un ensemble de noeuds forme un réseau. Ce réseau correspond à la structure de la matière ou au réseau sémantique de l'utilisateur. L'ensemble des noeuds forme une base de données qui est emmagasinée dans la mémoire d'un ordinateur grâce au disque dur, CD-Rom, vidéodisque, etc. Dans un logiciel comme "HyperCard", introduit par la société "Apple" [Apple, 2006], les noeuds textuels sont inscrits dans des champs qui sont des objets résidant dans une carte.

Les noeuds peuvent être classés en trois sous-types [Reyes, 2007] :

- a. **Noeuds édités** : ce sont des unités élémentaires associées à des fragments d'information d'un ou de plusieurs types : texte, graphique, image, son, etc. De manière plus exhaustive, deux types de noeuds édités peuvent être identifiés :
  - *Noeuds doc-texte* : ils contiennent des fragments de texte. Le contenu d'un noeud texte peut être un document proprement dit ou faire partie d'un document défini à un niveau supérieur comme un noeud calculé.
  - *Noeuds descriptifs* : ce sont des entités contenant des attributs dont les instances peuvent être atomiques (entier, décimal, etc) ou bien complexes (image graphique, son, vidéo, animation interactive, programme exécutable). Dans ce dernier cas, l'entité peut être incrustée dans un noeud ou former un noeud indépendant.
- b. **Noeuds calculés** : ils sont construits par traitement automatique, ou assistés, sur le contenu des noeuds édités. Ils peuvent être perçus comme une association de noeuds édités.
- c. **Noeuds composites** : ce sont des montages de réseaux hypertextes qui permettent de constituer un super-réseau hypertexte modulable à volonté.

## 4.2 Lien:

Dans une application hypermédia, les noeuds sont associés entre eux par des liens. Les liens constituent le principal moyen pour organiser un document d'une manière non séquentielle [Rhéaume, 2009]. Ils permettent à l'utilisateur de se déplacer d'un endroit à un autre dans un document, ou d'un document à un autre, ou d'un endroit d'un document à un



Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.

Click Here to upgrade to  
Unlimited Pages and Expanded Features

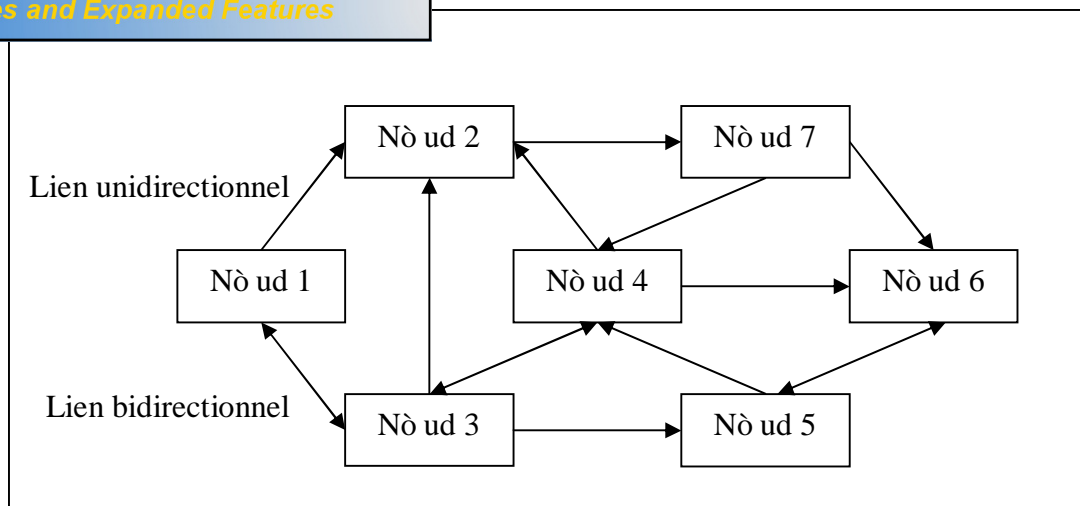
peut être uni ou bi-directionnel. Dans le cas d'un lien unidirectionnel, la source de ce lien est appelé une "référence" et leur but est appelé un "référent".

Les liens peuvent être classés en trois sous-types [Reyes, 2007] :

- a. **Liens référentiels édités** : ils rendent possibles les références directes entre différents nòuds, notamment celles entre des nòuds de niveaux différents. Ces liens sont les références intégrées dans le document par l'auteur afin d'orienter les lecteurs vers une partie du document en cours de consultation, ou vers un autre document susceptible de leur fournir de plus amples informations sur un point particulier.
- b. **Liens calculés** : ces liens s'opposent aux liens édités, établis manuellement. Ils sont issus d'un processus automatique qui peut prendre place avant chaque nouvelle version du produit hypertexte (statiques) ou issus d'un calcul en temps réel (dynamiques).
- c. **Liens typés** : ils permettent l'édition d'objets génériques à partir des nòuds édités et/ou calculés. Ces liens permettent une modélisation des opérations d'agrégation, de décomposition et d'héritage selon l'approche orientée objet. Deux exemples de liens typés sont les liens "est-un" (*is-a*) et les liens "a" (*has*).
- d. **Liens inverses** : c'est l'inverse de tout lien, valable aussi bien pour les nòuds édités que pour les nòuds calculés.

#### 4.3 Ancre :

Dans le cas d'un texte, l'ancre repère un endroit d'un texte, c'est-à-dire une unité sémantique de niveau inférieur à celui du nòud. Les ancres sont définies comme une zone sensible à l'intérieur d'un nòud et à laquelle des propriétés et des actions peuvent être associées. Une ancre dans un texte est perçue couramment comme une unité de type inférieur à celui des nòuds "doc-texte". Dans la plupart des cas, une ancre est visible par les lecteurs, elle est tout de même fonctionnelle dans la mesure où elle établit un point de repérage dans un nòud. Quant aux ancres dans un objet image ou son, elles contiennent le point de départ ou d'arrivée d'un lien [Reyes, 2007].



**Figure 1 :** Structure de documents hypermédia.

À partir de ce que nous avons présenté, nous pouvons induire la figure 1 qui montre la structure de document hypermédia (les deux termes "document hypermédia" et "application hypermédia" sont similaires [Nanard, 1998]). Les rectangles désignent les nœuds dans le document. Les flèches avec un seul sens désignent les liens unidirectionnels et les flèches avec deux sens désignent les liens bidirectionnels.

## 5. Notion et formes de navigation :

Nous essayons dans cette section de préciser la notion de navigation puis nous présentons quelques formes principaux de celle-ci.

### 5.1 Notion de navigation :

La métaphore d'hypermédia la plus vulgarisée est celle de la navigation. Le terme "navigation" fait référence aux différentes méthodes d'accès et lecture des documents hypermédia [Nanard, 1994]. La navigation dans un document hypermédia consiste de suivre des liens et d'utiliser des ancres pour consulter les noeuds. La navigation appliquée à la connaissance à été introduit, pour la première fois, par Francis Bacon qui va introduire, avec la métaphore de "l'océan de la connaissance", et par la, de la navigation, une véritable réforme de la pensée encyclopédique [Chatlain, 1996].

La navigation permet, à partir d'un mot, d'accéder aux autres occurrences de ce mot dans le texte par l'intermédiaire d'un pointeur appelé un lien hypertextuel, et qui est l'analogue exacte d'un index. Avec l'informatique l'automatisation des procédures d'accès facilite grandement cette opération qui peut même être encore généralisée : il est loisible de construire n'importe quel lien hypertextuel non seulement d'un mot vers ses différentes occurrences, mais aussi d'un mot vers sa définition, d'un mot vers son équivalent dans une autre langue, d'un mot vers ses synonymes, d'un mot vers sa transcription phonétique, les possibilités sont infinies [Educnet, 2009].



vu dans cette partie, nous pouvons conclure que la navigation est l'action d'accéder à l'information désirée (disponible), habituellement par un clic de souris sur un lien indiqué par un bouton ou une marque colorée et soulignée, sur le réseau WWW. La navigation est une substitution de la "lecture" pour désigner l'exploration de l'information électronique.

## 5.2 Formes de navigation :

Nous distinguons plusieurs formes de navigation selon le type d'hypermédia et selon l'objectif de l'utilisateur [Rhéaume, 2009]. Dans cette partie nous présentons les deux types les plus utilisés dans les applications hypermédia.

### 5.2.1 Cheminement :

Le cheminement est une séquence ou une sélection de noeuds d'information pertinents, à un objectif de navigation. Tout comme l'auteur d'un livre suggère de lire son oeuvre de la première à la dernière page, l'auteur d'un hypermédia peut suggérer sous forme de menu ou de carte des itinéraires convenant à telle ou telle circonstance. Le cheminement peut aussi désigner le parcours de navigation effectivement suivi par un usager à travers un hypermédia, de manière à pouvoir retourner à des noeuds vus antérieurement. Cette trace du cheminement est très utile, pour l'auteur, au moment de la construction d'un hypermédia; pour l'enseignant, au moment d'évaluer le cheminement d'un apprenant ou pour tout usager, simplement comme mode personnel de lecture et de navigation.

### 5.2.2 Tour guidé :

Pour l'utilisateur qui n'a pas d'objectifs précis de navigation en tête, un tour guidé peut être offert en guise de sentier tutorial, par exemple. Le tour guidé est aussi appelé le cheminement par défaut. Les sentiers sont des adaptations de l'information aux besoins ou aux caractéristiques individuelles des usagers. Par exemple, des sentiers plus graphiques peuvent être offerts aux usagers qui apprennent mieux par des relations visuelles, ou des sentiers au vocabulaire plus simple peuvent convenir à ceux qui apprennent une nouvelle langue, etc.

## 6. Structures de documents hypermédia:

Une grande diversité de structures et de modes de navigation est offerte par les hypermédiés [Sahraoui, 2007].

### 6.1 Hypermédia Linéaire Tourne Page :

Cette structure d'hypermédia se veut conforme à l'habitude de lecture traditionnelle, elle n'apporte aucun plus par rapport au livre. Chaque noeu dans cette structure se enchaîne au suivant et éventuellement au précédent par un lien. Ce modèle imite le feuilletage séquentiel du livre. L'interactivité recherchée dans les documents hypermédia est restreinte, le lecteur ne fait que tourner les pages une par une. Comme dans un livre il peut revenir en arrière et arrêter sa lecture quand il le désire.



Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

L'utilisateur dans un hypermédia structuré en arborescence se déplace selon un itinéraire préconstruit par l'auteur, c'est-à-dire d'un niveau à un autre. Comme dans la table des matières d'un livre (organisée en chapitres, sections, sous sections, paragraphes, phrases, mots, etc), l'utilisateur chemine de façon ascendante ou descendante (d'un niveau à l'autre de la hiérarchie). Donc l'information dans cette structure d'hypermédia apparaît sous une organisation hiérarchique dans laquelle les différents niveaux sont présentés de façon descendante. Autrement dit, de la racine principale à l'élément terminal en passant par des éléments intermédiaires.

### 6.3 Hypermédia Combinatoire :

L'architecture d'un hypermédia combinatoire est délinéarisée, elle se compose d'un nombre fini de nœuds. L'ensemble des parcours possibles est appelé "graphe fini" (déterminés par calcul mathématique). L'ensemble des liens est présenté à l'utilisateur de manière explicite pour l'aider dans son orientation. Le parcours à réaliser est à choix multiples. L'interactivité permet à l'utilisateur de choisir une direction librement, mais dans un ensemble déjà construit par l'auteur.

### 6.4 Hypermédia en Etoile :

Dans cette structure d'hypermédia, le lecteur peut accéder les différents nœuds à partir d'un nœud central. Une possibilité de retour de chacun de ces nœuds vers le point de départ est prévue. À partir du nœud central, l'utilisateur active des liens pour obtenir l'information complémentaire recherchée, il peut par la suite revenir au point de départ. Ce type d'hypermédia est souvent utilisé dans une hypermédia définitionnel, car sa structuration en étoile (nœud central d'informations principales donnant accès à des nœuds périphériques à informations secondaire) lui convient parfaitement.

### 6.5 Hypermédia en Maille de Filet :

Dans cette structure d'hypermédia, le lecteur peut se rendre à n'importe quel nœud à partir de chaque nœud, en empruntant le parcours qu'il désire. C'est l'utilisateur de cette architecture qui choisit les relations à établir entre les nœuds disponibles ; il active des liens dans l'ordre qui lui convient. Cette structuration de l'hypermédia est dynamique, elle illustre sa virtualité. Pour réaliser son propre hypermédia, l'utilisateur ne parcourt que les nœuds de son choix et ignore les autres. Théoriquement, chaque nœud permet l'accès à tous les autres. C'est le cas des dictionnaires électroniques.

## 7. Types d'usage de documents hypermédia :

Il y a en gros trois types d'usage des documents hypermédia [Nanard, 1994]:

- L'extraction d'information brute dans des bases d'information, apparentée à la recherche documentaire.



tion existante pour mieux la présenter, faciliter son accès, augmenter son impact chez le lecteur, en un mot, la valoriser.

- La production d'information ou des structures de connaissances nouvelles.

Dans ce qui suit, nous essayons de présenter les notions principales de chacun de ces types d'usage.

### 7.1 Extraction d'information :

Le lecteur dans un hypermédia est avant tout concerné par l'aspect recherche d'information. Il est plus intéressé par la rapidité d'accès à des documents pertinents et par l'abondance des informations obtenues que par les autres aspects, tels leur valorisation immédiate, la présentation, ou même la qualité des relations qu'un auteur aura ancré dans ces documents. Donc, l'hypermédia est un remarquable outil d'extraction d'information. Par son style d'interaction basé sur les documents, l'hypermédia est plus naturel pour le novice que les systèmes à base de requêtes.

L'information pour l'utilisateur est un minerai qu'il exploitera lui-même. En conséquence dans ce type d'usage, la structure de connaissance est souvent gérée de façon automatique, ce qui est une quasi nécessité compte tenu des volumes importants d'information mis en jeu. De plus, le type d'organisation de cette connaissance doit se refléter dans la surface de l'hypermédia par la présence de nombreux documents de style table des matières facilitant une navigation en profondeur. Donc l'utilisateur d'un hypermédia destiné à la recherche d'information est un fouineur plus qu'un lecteur.

L'hypermédia destiné à la recherche d'information est un outil pédagogique extrêmement flexible [Nanard, 1994] . Cet outil peut être utilisé de deux façons :

- **L'exploration libre d'information (ou vagabondage) :** dans la quelle l'utilisateur feuillette l'hypermédia comme la lecture d'une encyclopédie sans but initial. Ceci est très caractéristique des premières prises de contact des utilisateurs avec ces types de systèmes, par exemple WWW.
- **La recherche intentionnelle d'information :** c'est le type d'emploi finalisé des hypermédiats de cette catégorie. Elle a un rôle formateur extrêmement important, ceci pour deux raisons. D'une part, l'homme aura de plus en plus à acquérir d'information par lui-même dans les décennies qui arrivent. Il est donc très important de le former à ce type d'activité. D'autre part, en l'obligeant à se concentrer sur un but précis et à résister aux tentations de digression qui sont nombreuses lors d'une navigation. Ce type d'exercice a un rôle formateur extrêmement important, même chez les plus jeunes.

### 7.2 Valorisation de l'information :

La valorisation de l'information désigne une transformation destinée à rendre l'information mieux et plus rapidement assimilable par l'homme. Un énorme savoir-faire s'est développé grâce d'une part aux arts graphiques, d'autre part à tout un effort



es cognitives, et enfin une maîtrise des techniques de structuration de l'information [Nanard, 1994]. Les ouvrages pédagogiques actuels en sont un remarquable exemple. En dissociant l'aspect structurel de contenu, en permettant grâce au multimédia la consultation de documents au pouvoir évocateur fort, en permettant grâce aux réseaux l'accès à de l'information répartie.

Le rôle d'une équipe pédagogique qui développe un hypermédia afin de valoriser de l'information est typiquement d'extraire des éléments informationnels, de les interconnecter pour permettre au lecteur de satisfaire sa curiosité instantanée. Les hypermédiâs sont le vecteur d'avenir de la diffusion d'information valorisée. Les grands éditeurs ont déjà perçu l'intérêt de ce marché, et les nombreux CD-Roms disponibles n'en sont que le signe avant coureur.

### 7.3 Génération de l'information ou de structures de connaissances :

Les systèmes hypermédia destinés à la génération de l'information sont souvent appelés gestionnaires d'idées. La métaphore du jardinage introduite par Cathy Marshall, puis développé par Bernstein est très riche et subtile [Nanard, 1994]. Elle est extrêmement importante pour saisir l'esprit de ce type d'usage. Le jardinier d'information fait pousser de nouvelles informations et de nouvelles structures à partir d'informations existantes. Il est le vrai producteur d'information, il commence à travailler sur un ensemble d'informations nom hypertextualisé. L'observation puis l'explicitation de relations entre les informations conduit à l'émergence de structures qui s'ancrent sur les documents. Donc, l'activité d'usage de ce type d'usage de document hypermédia est de faire émerger des nouvelles informations et des nouvelles connaissances.

### 8. Ergonomie des applications hypermédia :

Un système hypermédia, en tant qu'application informatique, est déterminé par les règles générales de conception et d'ergonomie qui ont été développées au cours de l'évolution des ordinateurs [Reyes, 2007]. Dans cette section, nous présentons l'aspect ergonomique dans les applications hypermédia.

Depuis la présentation du système NLS (*oN Line System*) de D. Engelbart en 1968 [Englebart, 1984], l'évolution de l'interface graphique d'utilisateur a effectivement suivi une tendance vers la concrétisation de plusieurs de ses composants, tendance qui ne s'oppose pas à la recherche de nouvelles formes.

Généralement, la communauté scientifique s'accorde pour dire que l'interface graphique d'utilisateur GUI (*Graphical User Interface*) est composée de six éléments [Apple, 2006]:

- **Texte et médias :** il s'agit de ressources sous format texte ou média qui sont représentées à l'écran. Elles peuvent exister en tant que mécanisme de sortie ou d'entrée :
  - Sortie : éléments pour formater le texte : tableaux, cadres, calques.



Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

permettent d'exécuter des opérations habituelles telles que saisie, saisi, couper, copier, et coller comme les formulaires.

- **Icônes**  : ce sont des images utilisées pour identifier des boutons ou d'autres objets. Quatre types sont distingués :
  - Icônes d'une application : ce sont les icônes qui lancent un logiciel.
  - Icônes de documents : ce sont les icônes associées aux documents créés avec un logiciel.
  - Icônes de hardware: ce sont les icônes associées aux composants hardware de l'ordinateur.
  - Icônes à l'intérieur d'une application : ce sont les icônes qui par convention sont associées aux opérations que réalise un logiciel.
- **Pointeurs**  : ce sont les diverses représentations qu'un pointeur peut acquérir. Quatre types de pointeurs ont été identifiés : pointeurs de liens et d'état, pointeurs de sélection, pointeurs de déplacement d'éléments, pointeurs de défilement et de redimensionnement.
- **Menus**  : ils permettent d'accéder aux actions que réalisent les objets ou d'ajouter des attributs à un objet déterminé. Dans le cas du Web, cette fonction est souvent utilisée comme moyen d'accès aux différentes parties d'un site. Les menus sont parfois regroupés en sous-menus lorsque y existent des attributs interdépendants.
- **Fenêtres**  : elles fournissent un cadre pour la visualisation et l'interaction avec les données et les applications. Même si leurs comportements sont parfois similaires, quatre types de fenêtres peuvent être distingués :
  - Fenêtres de documents : elles contiennent la liste des documents appartenant à un répertoire.
  - Fenêtres d'applications : ce sont les fenêtres qui affichent le contenu. Dans le cas du Web, nous trouvons parfois des sous-fenêtres tel que les frames.
  - Fenêtres utilitaires : elles contiennent des contrôles ou des boîtes à outils.
  - Fenêtres d'alerte et de dialogue : elles requièrent une réponse de la part de l'utilisateur.
- **Contrôles**  : ce sont des objets graphiques qui déclenchent une action, ou des résultats visibles. Il existe divers types de contrôles:
  - Contrôles de sélection : ce sont les menus déroulants et les placards (petites régions qui montrent l'état actuel d'une fenêtre, par exemple 100%).
  - Contrôles d'ajustement : ce sont les flèches (vers le haut ou vers le bas), la barre de défilement, les boutons de taille plein écran, la taille personnalisée, la réduction et l'agrandissement.
  - Contrôles d'indicateurs : ce sont les barres de progrès, de relevance et de niveau.
  - Contrôles d'entrée du texte : ce sont les barres de recherche.



ation : ce sont les différents types d'affichage comme par exemple : affichage par onglets, affichage simultané et affichage arborescent.

Du point de vue du développeur, J. Raskin [Raskin, 2000] insiste sur le fait qu'un langage de programmation est aussi une espèce de GUI. Pour lui, la complexité des systèmes informatiques doit être contrebalancée par la simplicité de leur langage de développement.

## 9. Méthodes de construction de documents hypermédia :

La conception d'une application hypermédia s'apparente aux autres formes de conception. Elle consiste en la réalisation d'un produit conçu par un collectif d'auteurs mettant en œuvre des méthodes et des modèles [Gilles, 2005]. L'identification de l'information pertinente nécessaire aux différents contextes de lecture, la mise en place des liens, la réalisation des mécanismes de navigation, et l'étude de l'ergonomie et des interactions demeurent les éléments critiques à étudier dans le développement des applications hypermédia.

### 9.1 Caractéristiques de documents hypermédia :

Dans cette sous section, nous essayons de présenter les caractéristiques principales des applications hypermédia qui doivent être prises en compte par les auteurs pendant le processus de construction de leurs applications.

Les caractéristiques principales de document hypermédia sont [Educnet, 2009]:

- **Non-linéarité** : la non-linéarité de la lecture rendue possible par la navigation ( voir section 5) implique que le concepteur prévoie, organise et même contrôle la liberté d'exploration qu'il offre au lecteur. Un excès de liberté conduit à un phénomène connu sous le nom de "désorientation", comme nous le verrons dans la section 9 de ce chapitre, très rapidement, le lecteur ne sait plus où il est dans l'espace d'informations explorés. À l'inverse, une structure de navigation trop rigide (par exemple, principalement hiérarchique, comme nous avons vu dans la section 6 de ce chapitre) conduit à perdre l'intérêt majeur de la structure hypertextuelle : le partage d'informations entre situations de lecture différentes. L'organisation d'une hypermédia reflète la structure mentale de son concepteur ; cependant, chaque utilisateur peut adopter un parcours spécifique qui lui permettra d'adapter sa lecture à son niveau de compréhension.
- **Perspectivisme** : l'hypermédia est un document qui peut parfaitement expliciter la multiplicité des niveaux d'analyse et des points de vue possibles sur un même objet. Autrement dit, l'hypermédia concrétise et rend visible le perspectivisme nécessaire à l'appréhension de certains concepts. Autrement dit, l'hypermédia permet non seulement de multiplier indéfiniment les points de vue autour d'un concept, mais également de faire prendre conscience au lecteur du rapport entre un point de vue particulier et la définition spécifique que ce point de vue exprime sur l'objet.



**PDF Complete**

Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

Sur le World Wide Web, de nombreux facteurs humains et sociologiques jouent un rôle prépondérant, par exemple : l'exploration très rapide implique que la présentation visuelle des pages rende immédiatement perceptibles les points forts de leur contenu.

- **Conception centrée sur la communication :** à la différence des applications informatiques, dont la conception est le plus souvent fortement dirigée par les données, la conception des documents hypermédia doit être dirigée par les contraintes liées à la communication avec les humains et, en tout cas, centrée sur le lecteur.
- **Aspects multimédias :** les aspects graphiques et multimédias sont prépondérants. Ils demandent des équipes de conception pluridisciplinaires capables de prendre en compte les contraintes ergonomiques que celles d'origine informatique.
- **Spécificité du Web, des CD-Rom, de la documentation technique :** un document hypermédia doit être conçu spécifiquement pour l'usage de lecture qui en est fait et en tenant compte des contraintes techniques propres au moyen de diffusion utilisé. Un hypermédia relatif à une documentation technique aura une structure de navigation très fortement liée à la tâche ou à l'objet qu'il décrit ; au contraire, la structure d'un CD-Rom grand public découlera plutôt d'un travail créatif dans lequel la structure rhétorique est prépondérante ; enfin, le Web, vu comme un espace d'information, supporte assez bien des structures de navigation beaucoup plus simples et quasi standardisées. Plus encore, le Web est aussi un espace de liberté et de communication de type "tous-tous" dans lequel les structures émergent de l'interaction entre usagers tout autant, et sinon plus, qu'elles ne résultent d'une construction volontaire.
- **Synchronisation :** les objets multimédia variés qui constituent un document hypermédia sont certes structurés au niveau logique, c'est-à-dire le document est décomposé en parties, chapitres, sections, sous-sections, etc=et au niveau spatial, en permettant le contrôle du placement des différents objets à l'écran ou dans une page. Mais au niveau temporel, très peu d'outils permettent d'organiser les objets multimédia. Bien qu'une page HTML (*Hyper Text Markup Language*)<sup>4</sup> contienne différents médias animés, ces derniers ne sont pas synchronisés entre eux et n'exploitent que de manière simpliste les capacités vidéo. Dans certains types de document comme les supports pédagogiques, il est intéressant de structurer les différents médias afin d'obtenir une présentation sophistiquée et extrêmement planifiée dans le temps, avec néanmoins une bonne interaction avec l'utilisateur. Ces aspects temporels sont regroupés sous le terme de "synchronisation hypermédia".

<sup>4</sup> HTML: est un langage qui formalise l'écriture d'un document avec des balises de formatage indiquant la façon dont doit être présentée le document et les liens qu'il établit avec d'autres documents. Il permet, entre autres, la lecture de documents sur Internet à partir de machines différentes grâce aux protocoles.

Applications hypermédia citées au dessus, rendent la conception de ce type d'applications une tâche très difficile, parce que l'auteur se trouve dans une situation où il doit prendre en compte tous ces caractéristiques pendant leur processus de développement.

### 9.2 Dimensions de documents hypermédia :

L'hypermédia est une "vitrine" sur un espace à plusieurs dimensions qui doit être analysée, décrite et mise en scène. Trois dimensions sont généralement identifiées [Gilles, 2005]:

- **Dimension informationnelle** : il s'agit ici d'analyser et de décrire l'information multimédia qui sera accessible par l'hypermédia, définir sa forme, sa structure et les liens existant entre chaque granularité d'information. L'étude de cette dimension attribue à l'information le statut de donnée, elle s'apparente aussi à une description et une représentation d'un domaine de connaissance particulier.
- **Dimension navigationnelle** : cette dimension expose le potentiel des parcours pertinents qui peuvent être réalisés dans la dimension informationnelle. Ces parcours sont établis grâce aux liens identifiés entre les entités informationnelles identifiées dans la première dimension.
- **Dimension interactionnelle** : cette dimension décrit la mise en scène des parcours destinés aux utilisateurs de l'hypermédia sous la forme d'un enchaînement d'interactions sur les informations multimédias présentées.

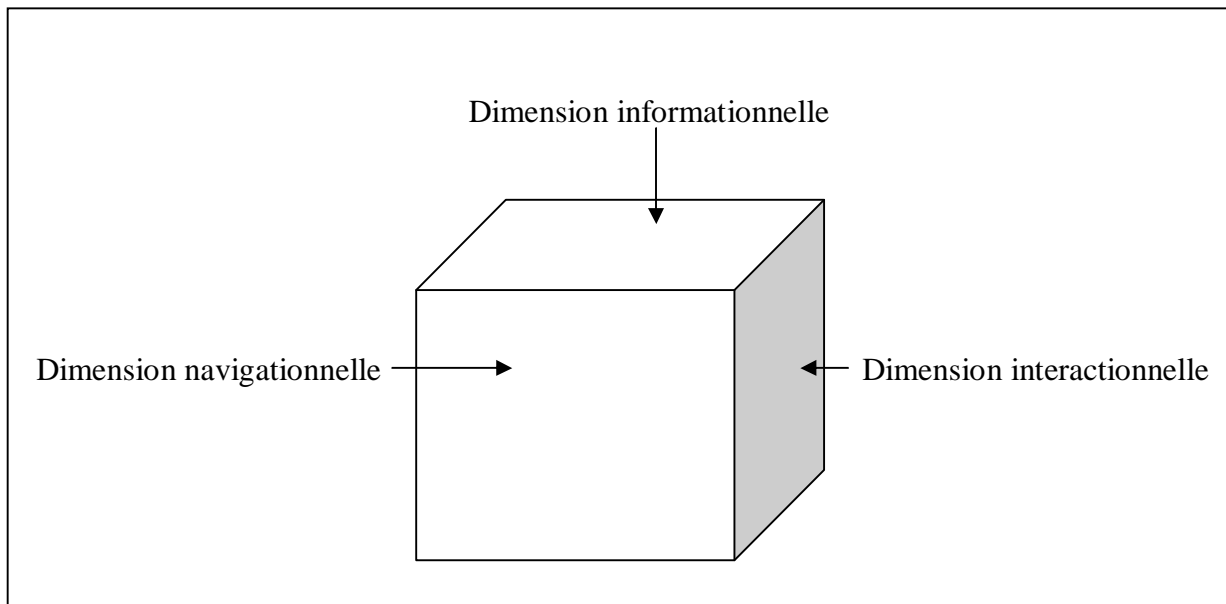


Figure 2 : Dimensions de documents hypermédia.

La figure 2 montre que les trois dimensions d'une application hypermédia sont complémentaires, la dimension informationnelle permet d'identifier les entités d'un



liens entre ces entités, la dimension navigationnelle permet d'identifier les parcours de navigation possibles entre les entités de domaine identifier dans la dimension informationnelle, et la dimension interactionnelle décrit la mise en scène de ces parcours de navigation pour identifier les interactions possibles entre l'utilisateur et l'application hypermédia.

### 9.3 Acteurs et domaines d'applications des systèmes hypermédia :

La conception de système hypermédia engendre plusieurs formes d'activités réalisées par un collectif d'acteurs que les méthodes essaient d'instrumenter. Parmi ces activités, nous identifions la conception de logiciel, la conception et la modélisation d'espace, la conception de vitrine (interface graphique), la conception d'un objet et plus globalement la conception d'un produit. Même si le domaine d'application couvert par l'hypermédia va conditionner la participation de certains types d'acteurs dans la conception, un ensemble commun ou "noyau" d'acteurs peut être identifié [Behaz, 2008]:

- **Spécialiste du domaine (enseignant, documentalist, etc) :** le spécialiste de domaine possède la connaissance du domaine à hypermédiatiser et des utilisateurs potentiels.
- **Concepteur de l'espace :** c'est souvent un informaticien mais ce peut être toute personne ayant une connaissance dans la modélisation de données et qui aura la charge de construire les modèles (dimensions) décrivant l'espace.
- **Maître d'ouvrage (ou le producteur ou encore le mandataire) :** c'est celui qui finance, à qui il faut montrer rapidement que l'hypermédia a une chance de voir le jour.
- **Graphiste :** le graphiste va proposer l'habillage de la vitrine, il doit être impliqué rapidement dans le processus de développement.
- **Programmeur :** c'est certainement un informaticien qui connaît la plateforme de développement et qui aura la charge de traduire les modèles sous une forme "exécutable".

De nombreux domaines où l'outil hypermédia est utilisé avec des objectifs et des besoins différents sont identifiés dans le tableau 1. Cette étendue des domaines d'applications, dans laquelle l'hypermédia prend place, montre bien l'importance que prend cette forme de conception de logiciel et la nécessité d'être encadré et guidé par des méthodes.

		Objectifs
Culturel/Ergonomique	Présentation d'un musée, artiste, lieu, film, etc.	Informer, susciter l'intérêt, la curiosité
Pédagogique	Apprentissage d'un domaine de connaissance, d'une langue	Apprendre, suivre l'évolution, évaluer
Professionnel	Présentation d'une activité, d'un produit, d'un catalogue	Communiquer, vendre
Documentaire	Présentation d'une collection de documents	Trouver une information, comparer, évaluer la pertinence
Gestion de projet	Présentation de l'état d'un projet, configuration d'un projet	Coordonner l'activité d'un groupe

**Tableau 1 :** Domaines d'applications des systèmes hypermédia (tiré de [Gilles, 2005]).

#### 9.4 Exemples de quelques méthodes de construction de documents hypermédia:

Comme nous avons cité dans la section 2 de ce chapitre, c'est V. Bush [Bush, 1945], qui a proposé l'idée d'un outil permettant à un homme d'assister dans sa quête d'information par un parcours non linéaire de texte. T. Nelson proposa ensuite, les termes d'hypertexte et d'hypermédia [Nelson, 1970]. Depuis, de nombreuses applications utilisant les principes des hypermédiats ont vu le jour. L'apparition des CD-Roms puis ensuite du Web a amplifié le phénomène [Millard, 2006]. Face à cette prolifération d'applications, des travaux sur la construction et la réalisation des documents hypermédia ont vu le jour. Dans cette sous section, nous essayons de présenter une synthèse sur quelques unes.

Les méthodes de construction des documents hypermédia les plus connues sont :

- **La méthode HDM (Hypermedia Design Method):** cette méthode permet de construire le document hypermédia comme un ensemble des entités. Chaque entité est composée d'une hiérarchie des composants, les entités peuvent être connectées aux autres entités ou composants par des liens. Ces liens peuvent être des liens structurels ou des liens d'application : les liens structurels donnent la structure hiérarchique des entités de document hypermédia et les liens d'application permettent de lier les entités ou les composants aux autres entités ou composants pour former les relations de domaines d'applications hypermédia [Garzotto, 1995].
- **La méthode RMM (Relationship Management Method):** cette méthode repose sur le modèle E/A ( Entité/Association). Elle a été étendue ultérieurement par ses auteurs pour s'adapter aux applications complexes exigeant la combinaison



**PDF Complete**

Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

proposition webArchitect [Takahashi, 1997]. Cette méthode définit une version étendue des notations de RMM pour décrire l'architecture des systèmes d'information basés sur le Web [Isakowitz, 1995].

- **La méthode OOHDM** (*Object-Oriented Hypermedia Design Method*) : cette méthode a été la première à utiliser les concepts orientés objet. Depuis, plusieurs autres méthodes orientées objet se sont succédées en mettant l'accent sur des aspects différents [Schwabe, 1996].
- **La méthode HFPM** (*Hypermedia Flexible Process Modeling*): la méthode HFPM est une méthode orientées objet. Elle propose un processus d'analyse qui permet de couvrir la totalité du cycle de développement [Olsina, 1998].
- **La méthode SOHDM** (*Scenario-based Object-oriented Hypermedia Design Methodology*): cette méthode se base sur les scénarios pour établir des diagrammes d'activités déterminant les structures d'accès aux nœuds [Lee, 1998].
- **L'approche OO/ pattern** (*Object-oriented Pattern*): cette approche se distingue par l'introduction de patrons de conception de navigation et de présentation [Thomson, 1998].

D'autres propositions de méthodes ont particulièrement visé le développement des sites Web, par exemple :

- **La méthode WSDM** (*Web Site Design Method*): cette méthode propose une conception d'application Web en reprenant les différents niveaux de conception d'un hypermédia mais avec pour originalité de placer l'utilisateur au centre de l'approche de conception [DeTroyer, 1997].
- **La méthode WebML** (*Web Modeling Language*): cette méthode représente un aboutissement de travaux effectués autour des méthodes HDM et RMM. Elle se caractérise par l'utilisation du standard XML pour décrire ses modèles. Elle propose un processus sous la forme d'une itération de constructions de plusieurs modèles [Ceri, 2000]. Les principaux modèles se présentent comme suit :
  - *Le modèle structurel* : il s'agit du modèle E/A classique comportant un héritage simple et des relations binaires sans attributs.
  - *Le modèle de composition* : ce modèle a pour objectif la détermination des pages Web à construire et ce qu'elles reflètent des nœuds d'information.
  - *Le modèle de navigation* : c'est le modèle qui permet d'établir les liens hypertextes entre les unités et les pages du modèle de composition.
  - *Le modèle de présentation* : il permet de définir l'apparence des pages Web.



**PDF Complete**  
Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

se présente comme une proposition de conception hypermédia plus complète, car elle définit une extension UML (un méta-modèle) pour la construction de modèles de conception, de navigation et de présentation [Koch, 2000].

- **La méthode OO-H Method** (*Object-oriented Hypermedia Method*): cette méthode repose aussi sur l'utilisation d'UML pour la description conceptuelle des données. Elle propose aussi deux autres niveaux de conception, navigation et présentation [Gomez, 2000].
- **La méthode Teresa** (*Transformation Environment for inteRactivE Systems representAtions*): cette méthode désigne à la fois un environnement de développement supportant la conception et la génération d'une interface utilisateur multimodale, mais aussi le langage de description d'interface utilisateur qui l'accompagne. La méthode utilisée pour le développement d'une interface multimodale consiste en plusieurs raffinements successifs d'un modèle de tâche décrit à un haut niveau d'abstraction. Cette méthode est divisée en quatre étapes : modélisation haut niveau de la tâche d'une application multi contexte, développement du modèle de tâche système pour les différentes plateformes considérées, transformation d'un modèle de tâche système vers une interface utilisateur abstraite et génération de l'interface utilisateur [Paterno, 2003].
- **Le langage UsiXML** (*USer Interface eXtensible Markup Language*): est un langage XML de description d'une interface utilisateur pour différents contextes d'utilisation telles que les interfaces graphiques, tactiles, sonores ou multimodales. Ce langage s'inscrit dans une méthode dite de développement multidirectionnel d'interface utilisateur. Ce type de développement nécessite un cadre de référence de base, c'est-à-dire un processus de conception composé de plusieurs étapes adapté au développement d'applications interactives multi contextes [Limbourg, 2004]. UsiXML se base sur les travaux du projet "Cameleon" [Calvary, 2003] dans lequel le processus de conception est constitué de quatre étapes suivantes pour deux contextes d'utilisation différents : tâche et concepts, user interface abstraite, user interface concrète et user interface finale.

### 9.5 Avantages et inconvénients des méthodes de construction de documents hypermédia :

Dans la partie précédente nous avons présenté les différentes méthodes de construction de documents hypermédia. Pour les avantages de ces méthodes, nous avons remarqué que le Web est devenu le support consacré des applications hypermédia. La prise en compte des spécificités du Web et de ses technologies dans les méthodes de conception des applications hypermédia est devenue incontournable. Les nouvelles méthodes utilisent les nouvelles technologies pour réaliser l'hypermédia tout en s'appuyant sur les concepts énoncés par les anciennes méthodes. Elles proposent toutes une modélisation du domaine, des parcours et de l'interface avec une meilleure prise en compte des différents types d'utilisateurs. Une nouvelle forme d'ingénierie est née : l'ingénierie du Web (*Web*

globalement la conception de SIw (Système d'Informations pour le Web) [Acerbis, 2008].

Pour les inconvénients, une première observation est que certaines de ces méthodes ne couvrent pas le cycle de vie en entier, ceci est particulièrement vrai pour les méthodes basées sur des langages de description d'interface, par exemple : UsiXML [Limbourg, 2004] ou Teresa [Paterno, 2003] qui ne traitent pas des phases d'utilisation et d'évaluation de l'interface utilisateur finale et de maintenance. En outre, chaque processus de conception d'application est différent des autres sur plusieurs points : l'activité menée une même étape peut être différente, par exemple : dans l'étape de spécification de l'application, appelé "Stratégie" du processus de conception de [Fleming, 1998], un modèle de la structure de l'application est produit, alors que dans la même étape pour Teresa [Paterno, 2003], un modèle de tâche est produit. Certaines étapes qui sont les mêmes, sont divisées en sous étapes différentes, par exemple : WebML [Ceri, 2000] divise l'étape de conception en quatre étapes, alors que dans OOHDM [Schwabe, 1998], cette même étape est divisée en deux sous étapes.

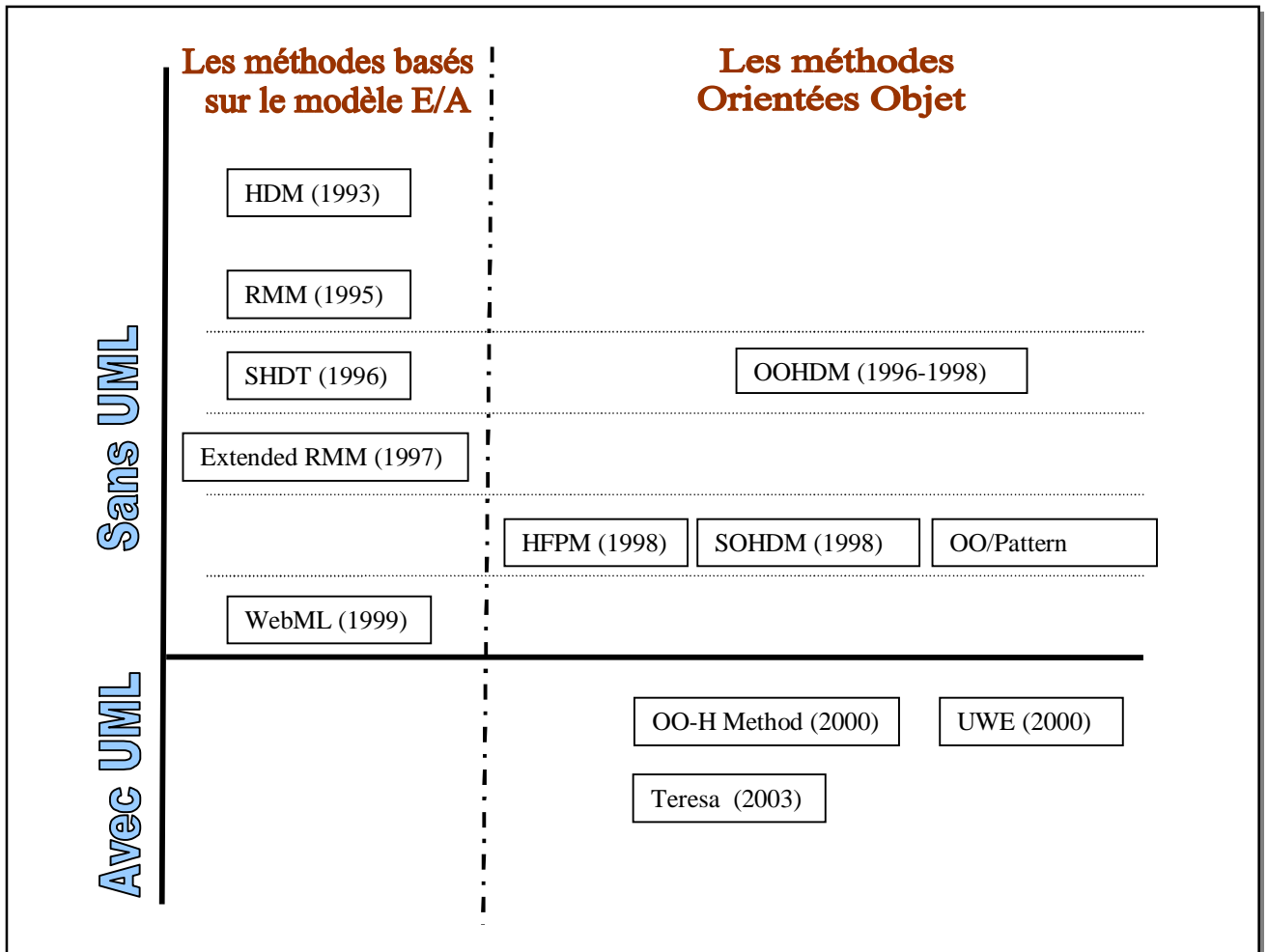


Figure 3 : Classification des méthodes de conception d'applications hypermédia (tirée de [Gilles, 2005]).

Les modes de conceptions des applications hypermédia présentées dans cette sous section. Elle présente deux classifications principales : une classification selon le modèle qu'elle utilise, et une classification selon leur utilisation du langage UML. Le point commun à l'ensemble de ces méthodes est de proposer un ensemble de modèles pour représenter toutes ou une partie des différentes dimensions d'un hypermédia : informationnelle, navigationnelle, interactionnelle (comme nous avons vu dans la sous section 9.2 de ce chapitre). Associé à ces modèles, nous trouverons parfois une démarche (processus) permettant de progresser d'un modèle à l'autre.

## 10. Problèmes issus d'applications hypermédia :

Quand la structure du document est éclatée, l'information contenue dans un hypermédia est dispersée. L'utilisateur ne peut donc pas avoir une vue complète ni de l'ensemble des composantes du document, ni de l'abondance de ces dernières. La problématique de l'utilisation de l'hypermédia représente une des grandes difficultés de la recherche [Sahraoui, 2007]. En tant que technologie d'apprentissage, les hypermédiés sont en même temps utiles et problématiques. La virtualité, la flexibilité structurelle, l'interactivité sont leurs atouts par rapport aux textes traditionnels, mais en même temps ils peuvent être aussi, la source de plusieurs problèmes. Les plus importants sont ceux de la surcharge cognitive et de la désorientation. Ce sont deux problèmes qui s'appellent. Dans ce qui suit, nous essayons de présenter ces deux problèmes.

### 10.1 Surcharge cognitive :

La surcharge cognitive est un effet produit chez le lecteur qui n'a qu'un écran pour travailler et qui doit s'efforcer de trouver à quoi telle information doit être associée pour être mémorisée et significative. Cette surcharge provient d'une part de la mémoire à court terme qui a tendance à tout oublier en passant d'un écran à l'autre, et d'autre part du manque d'acculturation de l'utilisateur qui n'a jamais développé cette habitude de lecture et d'apprentissage [Rhéaume, 2009].

### 10.2 Désorientation :

La façon dont l'hypermédia a été conçu et structuré au départ permet au lecteur d'appréhender et de maîtriser sa recherche d'informations plus ou moins facilement. Dans un hypermédia, la désorientation est l'effet cognitif produit chez l'utilisateur qui ne fait plus le lien entre son projet de navigation initial et les zones d'informations qu'il est en train de lire [Rhéaume, 2009]. Le problème peut provenir encore de la mémoire à court terme qui est trop courte pour se rappeler des informations déjà reçues ou pour les relier aux objectifs poursuivis. C'est l'éternelle question: Ce qui se résume en un certain nombre de questions du type :

- «Où suis-je ?»,
- «D'où est-ce que je viens ?»
- «Mais qu'est-ce que je recherche ?».

La désorientation est une des grandes difficultés de l'utilisation des applications hypermédia. Elle est principalement due à l'organisation des connaissances, prévue par les



des non linéaires. Celle-ci ne correspond pas toujours à celle qui est privilégiée par les utilisateurs. Pour remédier à cette désorientation plusieurs auteurs proposent de construire des environnements centrés sur les utilisateurs autrement dit, en prenant en compte leur profil cognitif [Rouet, 1999].

### 10.3 Outils de navigation :

Face à des problèmes de la surcharge cognitive et la désorientation, les concepteurs des applications hypermédia proposent des outils d'aides à la navigation pour remédier ces deux problèmes [Sahraoui, 2007]. Dans ce qui suit, nous essayons de présenter quelques uns :

- **Historique :** l'historique permet de retrouver tous les nœuds activés au cours de sessions de navigation. Il permet aussi de connaître le temps passé par l'utilisateur lors de la consultation d'un nœud.
- **Retours en arrière :** les retours en arrière autorisent la réactivation des documents déjà vus. Généralement, deux types de retours sont distingués : les retours directs en arrière vers le nœud choisi ou bien les retours en arrière des liens un par un (premier entré dernier sorti).
- **Traces :** les traces permettent de retrouver le cheminement suivi par un utilisateur (liste des documents, noms des documents, dates de leurs activations, etc). Elles permettent aussi, à un enseignant de suivre les séquences d'apprentissage et d'évaluer ses apprenants.
- **Graphiques :** les graphiques permettent de représenter un réseau hypermédia afin d'avoir une idée de sa taille, ainsi que du nombre de nœuds et de liens. La carte obtenue met en relief les textes en cours d'activation, ainsi que les nœuds connexes désactivés.
- **Signets :** les signets sont utilisés dans les cheminements. Le cheminement peut être volontairement marqué par des signets qui permettent, à l'utilisateur de retourner à des endroits spécifiques. Ces signets permettent entre autres de se retrouver dans un cheminement qui s'étale sur plusieurs sessions à l'ordinateur.

Ces outils permettant de guider l'utilisateur d'une application hypermédia dans son parcours de navigation pour minimiser l'effet des problèmes de la surcharge cognitive et de la désorientation.

### 11. Conclusion :

Dans ce chapitre, nous nous sommes intéressés à la présentation des caractéristiques principales d'applications hypermédia. Nous avons vu que hypermédia est une approche de la gestion de l'information dans laquelle les données sont stockées dans un réseau de nœuds connectés. Ces nœuds peuvent contenir des textes, des graphiques, des sons ou des images aussi bien que du code informatique ou tout autre forme de données. Nous avons vu également que l'hypermédia est une généralisation de l'hypertexte dans lequel



**PDF Complete**  
Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

sous un formalisme textuel. Et enfin nous avons présente une synthèse sur les méthodes principales de construction d'applications hypermédia.

D'après cette synthèse, nous pouvons constater que la proposition d'une nouvelle approche pour la conception des applications hypermédia devient un besoin nécessaire. Cette approche doit prendre en considération les inconvénients des méthodes proposées dans la littérature. Nous avons remarqué que le principal inconvénient est que ces méthodes ne couvrent pas le cycle de vie développement de document hypermédia en entier, donc nous avons besoin d'une approche qui permet de construire un document hypermédia dans tout leurs cycle de vie. Par ailleurs, la technologie des patrons de conception a montré son efficacité dans le développement de logiciels dédiés à divers domaines. Cette technique du génie logiciel permet une définition ou une réutilisation des patrons tout en prenant en considération les principes et les moyens d'une bonne conception. Ces patrons de conception offrent les caractéristiques suivantes : la réutilisabilité, l'uniformité, la facilité d'utilisation et la fiabilité. C'est l'objectif du chapitre suivant de présenter les caractéristiques de ces patrons de conception.

## Chapitre 2

# Réutilisation et patrons de conception

### 1. Introduction :

Nous avons vu dans le chapitre 1 que plusieurs méthodes de construction des applications hypermédia sont apparues pour améliorer aussi bien la qualité des produits que la performance des processus utilisés [Gilles, 2005]. Mais, la complexité croissante de ces applications et leur évolution de plus en plus rapide, ont motivé un intérêt accru pour une approche de développement plus efficace : la réutilisation. Cette approche permet de construire une application nouvelle à partir d'éléments (objets, composants, etc) existants, testés et réutilisables. La réutilisation est une stratégie permettant de réduire le coût et d'améliorer la qualité des applications hypermédia [Nanard, 1998].

Les patrons de conception sont une forme particulière de composants réutilisables, ils constituent, en ce sens, une des voies les plus pertinentes en matière de réutilisation. Ils répondent, en effet, à ce besoin tout en capitalisant d'importantes des connaissances acquises et approuvées par plusieurs développeurs des applications hypermédia. De telles connaissances sont réutilisables dès la phase d'identification des besoins jusqu'à la phase d'implémentation. D'ailleurs, plusieurs techniques et méthodes ont été définies pour l'ingénierie de ces patrons et pour leur réutilisation dans le processus de développement [Gzara, 2000]. Nous nous intéressons dans ce chapitre à la présentation d'un état de l'art sur la réutilisation comme une approche générale et les patrons de conception comme une forme de réutilisation.

### 2. Réutilisation :

La réutilisation a été énoncée pour la première fois en 1968 par M. McIlroy [McIlroy, 1976], dans une conférence de l'OTAN sur l'ingénierie du logiciel, suite à un échec patent de la part de développeurs pour livrer des logiciels de qualité à temps et à prix compétitifs. Elle vise donc trois objectifs : diminuer les coûts de développement et de maintenance, réduire le temps de mise sur le marché, et améliorer la qualité du logiciel. La réutilisation se définit comme une nouvelle approche de développement de systèmes, qui propose notamment de construire un nouveau système à partir de composants logiciels existants. Elle s'oppose ainsi aux approches traditionnelles de développement, dans lesquelles la construction d'un nouveau système part de rien et nécessite de réinventer de grandes parties d'applications à chaque fois. Depuis, cette idée s'est imposée et la réutilisation constituée un enjeu majeur de recherche en ingénierie des systèmes d'information. La réutilisation introduite

tout d'abord pour essentiellement améliorer la productivité des équipes de programmation, en particulier pour le développement des interfaces. Les tendances actuelles exploitent la réutilisation dans les phases en amont du processus d'ingénierie des systèmes d'information : phases d'identification des besoins, d'analyse et de conception, passant de la réutilisation de code à la réutilisation de connaissances et des démarches de raisonnement de différents niveaux [Quafa, 2006].

### 3. Dimensions de la réutilisation :

La réutilisation dans l'ingénierie des systèmes d'information concerne en général deux dimensions : la dimension "produit" et la dimension "processus" [Quafa, 2006]. La dimension "produit" concerne les différents types de composants réutilisables. Dans la dimension "processus" sont considérées, d'une part, les différentes activités spécifiques à la production de ces composants, et d'autre part, les activités spécifiques à la mise en œuvre d'une approche de développement par réutilisation de ces composants. Dans ce qui suit, nous essayons de présenter la dimension "produit" avec les différents types de ses composants réutilisables puis nous présenterons la dimension "processus" avec leurs différentes tâches.

#### 3.1 Dimension produit (composants réutilisables) :

Une grande variété de composants réutilisables a déjà été proposée dans la littérature pour intégrer la réutilisation dans tout le processus de développement des systèmes d'information [Conte, 2001]. D'une manière générale, un composant est vu comme une solution testée et acceptée pour résoudre un problème fréquemment rencontré lors du développement des systèmes. Toutefois, plusieurs critères permettent de distinguer les très nombreux types de composants proposés aujourd'hui dans la littérature. Dans ce qui suit, nous essayons de présenter les différents types de ces composants.

##### 3.1.1 Composants de différents types de connaissances :

Les composants réutilisables peuvent être classés selon le type des connaissances qu'ils utilisent. Ils peuvent capitaliser des connaissances de type produit ou de type processus. Les composants de type produit correspondent au but à atteindre, ils sont destinés à être réutilisés et intégrés directement dans le système en cours de développement. Les patrons du GoF<sup>4</sup> [Gamma, 1995], les architectures logicielles et les bibliothèques de fonctions sont des exemples de composants de type produit. Les composants de type processus correspondent plutôt au chemin à parcourir pour atteindre le résultat. Ils servent à définir des fragments de démarches facilitant le développement. Les patrons de S. Ambler [Ambler, 1998] sont des exemples de composants de type processus.

---

<sup>4</sup> E. Gamma, R. Helm, R. Johnson and J. Vlissides, regroupés sous le patronyme du *Gang of Four* (GoF).

### 3.1.2 Composants de différentes couvertures :

La couverture d'un composant désigne le degré de leur généralité, c'est le problème traité par ce composant qui détermine sa couverture et donc son type. Certains peuvent être généraux, d'autres plus spécifiques à un domaine ou à une entreprise. Si le problème est très fréquent dans de nombreux domaines d'applications, celui-ci est dit "composant général". S'il est spécifique à un domaine particulier, ou à une entreprise particulière, celui-ci est dit "composant de domaine" (ou d'entreprise). Les frameworks [Beck, 1994] spécifiques à la comptabilité sont des exemples de composants de domaine.

### 3.1.3 Composants de différentes portées :

Les composants réutilisables peuvent être classés selon sa portée. La portée d'un composant détermine la phase du cycle de développement concernée par celui-ci, par exemple les composants d'analyse, de conception ou encore d'implémentation. Les frameworks d'architecture [Beck, 1994] sont par exemple des composants de conception. De manière générale, la plupart des composants actuels sont dit "orientés activité" et sont dédiés exclusivement à une phase particulière du cycle de développement. Il existe cependant des composants qui concernent plus d'une phase, et sont définis avec une certaine forme de traçabilité à travers les différentes phases concernées par ces composants.

### 3.1.4 Composant de différentes natures de solutions :

La solution d'un composant peut être de nature conceptuelle ou logicielle. Les composants logiciels correspondent à la forme la plus ancienne et la plus répandue de composants réutilisables. Ils permettent de réutiliser des fragments de programmes destinés à être intégrés directement dans le code (au niveau de la phase d'implémentation). Les composants conceptuels, par exemple : les patrons de S. Ambler [Ambler, 1998], et les patrons du GoF [Gamma, 1995], proposent des petites structures ou des architectures permettant de spécifier les systèmes d'information.

### 3.1.5 Composants de différents niveaux de granularité :

La granularité mesure le plus souvent le nombre d'unités modulaires constituant le composant. Elle spécifie le degré de complexité de sa structure interne ou de l'architecture qu'elle propose. Elle peut être faible (moins de 10 entités), moyenne (moins de 100 entités) ou forte (plus de 100 entités). La plupart des composants proposés sont de faible granularité, ils sont généralement définis par quelques entités modulaires atomiques telles que les classes. Les patrons de conception du GoF [Gamma, 1995], ou ceux d'analyse de P. Coad [Coad, 1992] offrent, par exemple, des petites structures de deux à six classes. Il existe cependant des composants de structures plus complexes. Le framework MVC<sup>5</sup> (*Model, View, Controller*) [Schmuker, 1987] de smalltalk-80 en est un exemple.

---

<sup>5</sup> MVC est constitué de trois types d'objets. Le modèle est l'objet d'une application, la vue est sa représentation visuelle, et le contrôleur définit la nature des réactions de l'interface utilisateur aux introductions de données.

### 3.1.6 Composants de différentes ouvertures :

L'ouverture d'un composant caractérise son niveau de transparence. Selon que son utilisation nécessite ou non la modification de sa structure interne, il convient de distinguer plusieurs types de composants qui vont de la boîte noire (où la structure interne est invisible et non modifiable), à la boîte blanche (où la structure interne est visible et modifiable), en passant par la boîte en verre (où la structure interne est visible mais non modifiable) ou la boîte grise (où la structure interne est visible et modifiable par paramètres) [Quafa, 2006].

### 3.2 Dimension processus:

Deux processus complémentaires sont considérés dans la dimension "processus": l'ingénierie par réutilisation de composants (*design by reuse*) et l'ingénierie de composants réutilisables (*design for reuse*).

#### 3.2.1 Ingénierie par réutilisation de composants :

L'ingénierie des systèmes d'information par réutilisation des composants couvre les tâches de recherche, de sélection, d'adaptation et d'intégration de ces composants. La recherche de composants consiste à rechercher un composant parmi une collection, permettant de répondre à un problème particulier. Elle tient donc compte du besoin du développeur.

Dans la plupart des cas, il n'y a pas un seul composant candidat mais plusieurs. Il est donc nécessaire de comparer l'ensemble des candidats en vue de sélectionner le plus approprié. Une fois sélectionné, le composant doit être adapté au contexte du système en cours de développement. Les techniques d'adaptation sont nombreuses : depuis la modification du composant, jusqu'au mécanisme d'instanciation, en passant par la paramétrisation et la spécialisation [Cauvet, 1999]. Enfin, la réutilisation d'un composant est effectivement close par l'intégration de celui-ci au système considéré. Il est à noter ici que le processus d'ingénierie de systèmes d'information peut être très différent, selon qu'il s'agit ou non de considérer l'ensemble des tâches ainsi identifiées. Par exemple, la tâche de recherche est nécessaire dans le contexte de certains types de composants [Cauvet, 2001]. En revanche, elle a peu d'intérêt, par exemple, dans le contexte d'un framework.

Une approche idéale de réutilisation devrait permettre de générer de manière quasi-automatique un système d'information, de sorte que le rôle du développeur se limite à fournir les propriétés et les besoins spécifiques à son système. Une telle approche doit être systématique et complète, pour couvrir l'ensemble des tâches tout en maîtrisant leur complexité. En ce sens, plusieurs travaux [Conte, 2001], [Arnaud, 2004], [Khayati, 2005], etc. centrent leurs efforts sur la recherche de méthodes systématiques qui sont mieux adaptées au développement de systèmes par réutilisation des composants.

### 3.2.2 Ingénierie de composants réutilisables (ingénierie pour réutilisation) :

L'ingénierie de composants réutilisables recouvre les tâches d'identification, de spécification, de qualification, d'organisation et d'implémentation de composants. L'ensemble de ces tâches est essentiel, puisqu'il a pour objectif de produire les ressources nécessaires à la mise en œuvre d'une approche d'ingénierie par réutilisation. Toutefois, il n'existe pas d'approches spécifiquement dédiées à l'ensemble de ces tâches. Dans ce qui suit, nous présentons l'ensemble de ces tâches.

#### a. Tâche d'identification de composants :

Diverses techniques d'identification de composants ont été définies par F. Semmak [Semma, 1998], ces techniques appartenant à deux grandes approches :

- **Approche par rétro-ingénierie :** l'approche par rétro-ingénierie vise à abstraire des éléments réutilisables à partir de l'analyse de produits existants. Cette approche ascendante permet d'évaluer les similarités et les différences entre produits, afin d'identifier de manière systématique des éléments candidats à la réutilisation. Dans ce cas, deux techniques de base sont principalement utilisées : l'analyse de similarités [Spanoudakis, 1996] et la recherche par analogie [Holyoak, 1989]. Bien que ces deux techniques aient sensiblement le même objectif, le degré de généralité des composants résultats de leur application est plus grand dans les techniques de recherche par analogie. En effet, la recherche par analogie, contrairement aux techniques d'analyse de similarités, évalue les ressemblances sémantiques entre éléments candidats (tels que la nature des liens entre objets, leur organisation au sein de systèmes, etc) ; plutôt que leurs similitudes syntaxiques qui rendent nécessaire une terminologie commune et donc le plus souvent mono domaine.
- **Approche par analyse de domaine :** cette approche consiste aussi à identifier les éléments communs à des systèmes similaires, afin d'élaborer un modèle de domaine. Elle peut couvrir plusieurs sous tâches, allant d'une analyse de contexte pour définir les limites du domaine et les sources de connaissances en rapport avec celui-ci, jusqu'à une conception détaillée de ces connaissances. Dans ce type d'approche, deux techniques qualifiées de statique et de dynamique [Cauvet, 2001] sont proposées :
  - Dans la première technique [Wartik, 1992], il s'agit de considérer une famille de systèmes ayant des fonctionnalités similaires (telles que la gestion des comptes bancaires, la gestion de stock, les systèmes de réservation, etc) dans l'objectif de produire un modèle de domaine réutilisable : un modèle qui exprime les ressemblances (partie commune) et les différences (parties spécifiques) entre les systèmes considérés.
  - Dans la deuxième technique [Semma, 1998], un domaine est défini par un ensemble de problèmes, pour lesquels différentes solutions sont

envisageables. Les modèles de domaine obtenus sont donc orientés problème et non plus solutions [Grosz, 1996], contrairement à la première démarche. L'optimisation des réservations ou la satisfaction des attentes des clients peuvent être des exemples des problèmes spécifiques au domaine de réservation de places d'avions.

### **b. Tâche de spécification et principes d'abstraction et de variabilité :**

La réutilisation des composants est d'autant plus efficace s'ils sont associés à une spécification pertinente. Une telle spécification doit favoriser leur réutilisation tout en fournissant l'ensemble des informations utiles et nécessaires à leur bonne utilisation (telles que les indications et les contraintes d'utilisation, les forces de la solution proposée, etc). Elle doit permettre de décider, par exemple, de la pertinence d'utiliser un composant ou un autre au moment de la recherche de ceux-ci. L'abstraction et la généralité sont deux techniques phares en matière de spécification de composants. Tout modèle de spécification doit idéalement appliquer ces deux principes. L'abstraction consiste à distinguer deux types de connaissances dans la spécification d'un composant : les connaissances effectivement réutilisables (un modèle s'il s'agit d'un composant conceptuel par exemple, ou un fragment de programme s'il s'agit d'un composant logiciel) et celles requises pour une utilisation correcte de celui-ci. Ces dernières mettent en avant, par exemple, le problème du composant ou encore son contexte d'utilisation [Cauvet, 2001].

Plusieurs approches de spécification peuvent être distinguées : approche orientée problème, approche par descripteur de composant (ou par classification) et approche par méta-connaissance [Quafa, 2006]:

- La première approche est exclusivement centrée sur la définition explicite du problème du composant.
- La deuxième approche propose d'associer à chaque élément d'une collection de composants une liste de mots clés, tout en le classant.
- La dernière approche s'applique exclusivement sur les composants définis sous la forme de classes. Elle propose de spécifier les composants sur deux niveaux : le niveau application (présentant les parties réutilisables des classes) et le niveau meta (associant des meta\_classes aux classes du composant, afin de guider le développeur dans le processus de réutilisation).

La généralité introduit une forme de variabilité dans la spécification des composants. En effet, le plus souvent, la solution proposée par un composant n'offre qu'une réalisation possible de celui-ci. Pour augmenter sa réutilisation, un composant doit offrir plusieurs réalisations possibles : sa solution doit ainsi être flexible afin de pouvoir s'adapter aux spécificités du système en cours de développement. Ce principe consiste donc à distinguer, dans l'ensemble des connaissances spécifiant le composant (outre la partie réalisation), entre

une partie fixe et une partie variable. La partie variable permet d'exprimer différents choix de développement, par exemple, ou encore de définir des extensions possibles de la réalisation proposée [Cauvet, 2001].

### c. Tâche qualification de composants :

La qualification des composants s'apparente à une forme de validation de ceux-ci. Elle consiste à vérifier la qualité de composants identifiés du point de vue de leur réutilisation. Peu de techniques de qualification sont actuellement proposées. De plus le résultat de cette tâche est en général très subjectif. Une telle qualification relève plus de la "qualité" du processus d'ingénierie des composants réutilisables que des composants eux-mêmes [Quafa, 2006].

### d. Tâche d'organisation de composants :

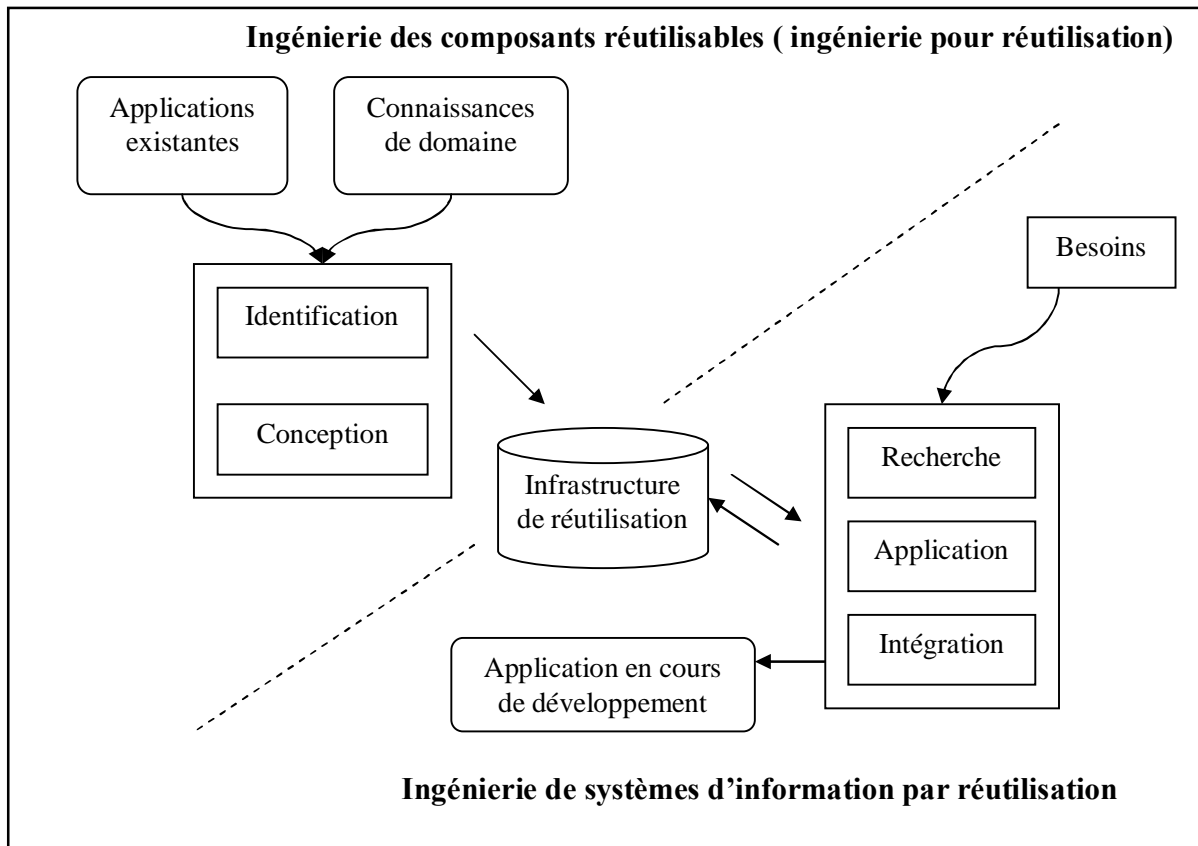
Cette tâche traite des différents types de liens entre les composants, afin de pouvoir utiliser conjointement plusieurs composants pour obtenir des architectures plus importantes. Elle a donc une influence directe sur la manière d'exploiter les composants dans le processus d'ingénierie de systèmes d'information. Elle facilite la sélection de composants et améliore leur réutilisation. Deux approches d'organisation de composants sont actuellement proposées statique ou dynamique [Cauvet, 2001]:

- **Approche statique :** cette approche prescrit entièrement le processus de recherche et/ou de composition des composants, sans tenir compte des spécificités du système en cours de développement. Elle est largement utilisée, par exemple, dans les frameworks qui prédéfinissent la structure du produit final.
- **Approche dynamique :** cette approche propose, en revanche, de coupler un ensemble de composants avec un processus qui automatise et/ou guide la construction d'un produit par la recherche, la composition, et l'adaptation des composants. Elle ne prédéfinit ni la recherche, ni la composition de ceux-ci, et vise une construction par réutilisation qui tient compte des besoins du système en cours de développement.

### e. Tâche d'implémentation :

L'implémentation des composants consiste à outiller la réutilisation (gestion automatique de leurs collections, recherche, sélection, composition, adaptation, etc) de ceux-ci [Quafa, 2006].

Ces différentes techniques contribuent à la mise en place de bases de composants pour capitaliser des savoirs et des savoir-faire réutilisables dans le cadre d'une ingénierie par réutilisation (voir figure 4).



**Figure 4 :** Ingénierie pour et par réutilisation de composants (tirée de [Cauvet, 2001]).

La figure 4 résume les tâches principales de la dimension "processus" de réutilisation que nous avons présenté dans cette sous section. Elle montre les deux processus complémentaires de cette dimension : l'ingénierie de composants réutilisables (*design for reuse*) et l'ingénierie de systèmes d'information par réutilisation de composants (*design by reuse*).

#### 4. Patrons de conception:

Comme nous avons introduit, les patrons de conception sont une forme particulière de composants réutilisables, ils constituent en ce sens une des voies les plus pertinentes en matière de la réutilisation. Ils répondent, en effet, à ce besoin tout en capitalisant d'importantes connaissances acquises et approuvées par plusieurs développeurs. Nous nous intéressons dans cette section à la présentation des patrons de conception comme une forme particulière des composants réutilisables.

##### 4.1 Historique :

La notion de patron a été généralisée par C. Alexander, dans le domaine de l'architecture des bâtiments [Alexander, 1977], [Alexander, 1979]. C. Alexander constatait que les grands

bâtitseurs, au fil des siècles, n'avaient pas suivi un modèle préétabli avec des règles rigoureuses, mais que les architectures avaient été adaptées les unes après les autres et à leurs environnements. Il a remarqué avec ses collègues que certaines solutions de conception, considérées comme efficaces et bien acceptées, étaient récurrentes et s'appliquaient dans différentes situations de construction. Cette idée est concrétisée à travers de 253 patrons propres aux problèmes récurrents en architecture décrits d'une manière uniforme pour des raisons de commodité, de clarté et de diffusion. Des approches similaires à celle de C. Alexander sont apparues dans d'autres disciplines. L'un des patrons de C. Alexander le plus fréquemment cité est "A place to wait" (un endroit pour attendre), qui propose des solutions conceptuelles au problème de l'attente [Alexander, 1979].

Dans le domaine de l'informatique, les premiers patrons ont été présentés par K. Beck et W. Cunningham [Beck, 1997] lors de la conférence "OOPSLA" de 1997. Il s'agissait d'une première adaptation du langage de patrons de C. Alexander à la conception et à la programmation objet. Un ensemble de patrons plus spécifiquement dédiés à la conception par objets a été élaboré, un peu plus tard, par E. Gamma [Gamma, 1991] en 1991 lors des travaux de sa thèse. Ces travaux ont été suivis en 1995 d'un premier ouvrage collectif [Gamma, 1995]. À la même période, dans le cadre de l'ingénierie de systèmes d'information, P. Coad [Coad, 1992] proposait de faciliter l'analyse d'un système en identifiant les besoins selon 7 patrons pré-établis. Il définit un patron "Objet" comme une abstraction de structures de classes similaires, une abstraction qui peut être réutilisée, encore et encore, pour le développement d'applications. Depuis, de nombreux ouvrages et articles ont été publiés sur les patrons [Gamma, 1995], [Coplien, 1996], [Buschmann, 1996], [Coad, 1997], [Booch, 1997], [Fowler, 1997], [Ambler, 1998], [Larman, 2002], [Palkovic, 2009], etc.

#### 4.2 Définition :

C. Alexander [Alexander, 1977] définit un patron comme suit : « *un patron d'architecture, tout comme un patron de couture, capitalise un savoir-faire permettant de résoudre un problème récurrent du domaine (l'architecture ou la couture) ... chaque patron décrit à la fois un problème qui se produit très fréquemment dans votre environnement et l'architecture de la solution à ce problème ; de telle façon que vous puissiez utiliser cette solution des millions de fois sans jamais l'adapter deux fois de la même manière* ». D'ailleurs, sa définition reste très proche de celle proposée par le dictionnaire : « *un patron est une forme finie et originale d'un modèle proposé ou accepté pour imitation ; quelque chose considéré comme un exemple destiné à être copie* ».

De manière générale, un patron constitue une base de savoir et de savoir-faire pour résoudre un problème récurrent dans un domaine particulier. Il permet d'identifier le problème à résoudre par capitalisation et organisation de connaissances d'expériences. Un patron propose une solution possible, correcte, générale et consensuelle pour y répondre et il offre les moyens d'adapter cette solution au contexte spécifique [Mathieu, 2007].

### 4.3 Utilité des patrons:

Les patrons ont d'indéniables qualités en termes d'usages, mais aussi, en termes d'intérêts dans l'ingénierie des systèmes d'information. Ils apportent des solutions efficaces à divers problèmes d'analyse et de conception, permettant ainsi de créer des conceptions approuvées et réutilisables. L'écriture des programmes est également facilitée par les patrons qui proposent des consignes types pour une bonne implémentation [Fowler, 1997].

D'ailleurs, la validation de spécifications de systèmes est souvent guidée par des patrons, qui conduisent à de meilleures conceptions. Les patrons offrent également un vocabulaire commun à l'ensemble de l'équipe de développement [Gamma, 1995], [Larman, 2002]. Ils améliorent la documentation, et donc la compréhension des applications en cours de développement. Celles-ci deviennent, par conséquent, plus facile à faire évoluer et à maintenir.

En termes d'intérêts, les patrons constituent un véritable moyen de transfert de connaissances. Ils permettent l'échange du savoir et savoir-faire des développeurs expérimentés, ayant rencontré et résolu plusieurs fois les mêmes problèmes. Ils offrent un gain de temps et une efficacité, tout en réduisant largement les tâtonnements fastidieux nécessaires à l'élaboration d'un système de qualité. De plus, tout en augmentant les facultés d'abstraction des développeurs, les patrons apportent également une meilleure compréhension des systèmes. Ils permettent, en effet, d'aborder des systèmes complexes de manière plus simple. Ils aident à mieux structurer les applications, guidant la perception du monde réel et permettant d'en obtenir une description à un niveau d'abstraction élevé. Il devient ainsi facile, par exemple, de comprendre le fonctionnement d'un framework, sans qu'il soit nécessaire de se pencher sur les détails de son implémentation [Schmidt, 1995].

Les patrons capturent donc les propriétés essentielles d'une architecture, tout en masquant les détails non pertinents. Ils facilitent, de la sorte, la communication autour d'abstractions, de concepts et de techniques que décrivent ou portent ces patrons, dans le but d'assurer de meilleures compréhension, évolution et maintenance des applications. D'autant plus que la documentation de celles-ci se trouve enrichie et simplifiée, facilitant le transfert des spécifications, des modèles et des concepts, notamment avec le personnel non technique qui n'est pas apte à comprendre tous les détails d'implémentation, mais est, en revanche, plus dispose à saisir les abstractions véhiculées [Larman, 2002].

Dans le cas des patrons de conception, l'utilisation de celui-ci offre de nombreux avantages [Mathieu, 2007]:

- Tout d'abord cela permet de répondre à un problème de conception grâce à une solution éprouvée et validée par des experts. Ainsi nous gagnons en rapidité de conception (nous diminuons également les coûts) et en qualité.
- De plus, les patrons de conception sont réutilisables et permettent de mettre en avant les bonnes pratiques de conception.

- Les patrons de conception étant largement documentés et connus d'un grand nombre de développeurs, ils permettent donc de faciliter la communication.

## 5. Différents types de patrons :

Dans cette section, nous introduisons les différents types de patrons.

### 5.1 Patrons produit et patrons processus :

Selon le type de connaissances capitalisées par les patrons, deux types de patrons sont distingués : les patrons produit et les patrons processus.

- **Patrons produit :** les patrons produit décrivent un savoir, ils offrent une prescription, adoptée ou non, pour construire tout ou partie des spécifications ou de l'implémentation du système à développer. Les patrons du GoF [Gamma, 1995] par exemple, sont des patrons généraux de type produit, dédiés à la phase de conception. Les patrons de P. Coad [Coad, 1997] sont cependant dédiés à la phase d'analyse. Les patrons de M. Adams [Adams, 1995] concernent plutôt la phase de recensement des besoins.
- **Patrons processus :** les patrons processus décrivent un savoir-faire sous la forme d'actions et/ou de tâches destinées à être suivies pour le développement des systèmes. Les actions composantes peuvent elles-mêmes être des patrons. Un patron processus décrit en général un processus de développement dans le but de répondre à des besoins spécifiques, il est ainsi exprimé sous la forme d'adaptations à une situation concrète. Les patrons de S. Ambler [Ambler, 1998] sont des patrons de type processus généraux qui couvrent l'ensemble du cycle de développement des systèmes.

### 5.2 Patrons de différents niveaux d'ingénierie :

Les patrons sont le plus souvent classifiés en fonction de la phase d'ingénierie à laquelle ils s'adressent, par exemple, les patrons d'analyse, de conception, etc. Les patrons d'architecture sont d'autres types de patrons qui s'adressent généralement à la phase de conception. Dans ce qui suit, nous présentons chacun de ces types.

#### 5.2.1 Patrons dédiés à la phase d'identification de besoins :

Ce type de patrons permet une description efficace des besoins des utilisateurs, sous la forme de comportements génériques attendus du système à développer. Il s'agit en général de comportements fonctionnels. Dans certains cas, les comportements non fonctionnels, tels que la performance ou la fiabilité, peuvent cependant être considérés. Les patrons dédiés aux systèmes de télécommunication tolérants aux pannes [Adams, 1995] traitent, par exemple, de

quelques besoins non fonctionnels tels que la fiabilité [Rising, 2000]. Usuellement, dans la littérature, les patrons d'identification de besoins font partie des patrons d'analyse.

### 5.2.2 Patrons d'analyse :

Les patrons d'analyse traitent des problèmes qui apparaissent lors de l'analyse des besoins des systèmes. Ils aident le développeur dans la construction de modèles devant représenter au mieux les besoins de son système d'information. Un patron de gestion de ressources permet, par exemple, de répondre à un problème de ressources humaines qui comprend l'affectation de personnes à des activités (ou encore, à un problème de gestion de ressources matérielles, pour réaliser l'affectation de machines dans un contexte de production) [Avgeriou, 2003].

### 5.2.3 Patrons de conception :

Les patrons de conception sont certainement les patrons les plus connus et sans doute les plus utilisés parmi les patrons d'ingénierie. Ils identifient, nomment et abstraient les connaissances convenues de plusieurs expériences, qui répondent à des problèmes de conception communs (qu'il s'agisse de conception détaillée, globale ou liée à des architectures particulières) [Gamma, 1995].

### 5.2.4 Patrons d'architecture :

Les patrons d'architecture sont des patrons de conception qui déterminent la structure de base d'une application. Ils offrent un ensemble de sous-systèmes prédéfinis, spécifient leurs responsabilités, et les collaborations et communication entre ceux-ci. Ils incluent, de plus, les règles et les conseils pour établir leurs associations. Tout patron d'architecture décrit en effet les règles d'organisation et de fonctionnement, les stratégies de haut niveau, les propriétés et les mécanismes globaux d'une application, permettant ainsi de créer des architectures extensibles et inter-opérables [Buschmann, 1996].

### 5.2.5 Patrons d'implémentation :

Les patrons d'implémentation sont des patrons de bas niveau qui traitent des problèmes d'implémentation. Ils peuvent décrire, par exemple, comment implémenter certains mécanismes absents dans des langages (simulation de l'héritage multiple en Java, par exemple). Ils sont généralement spécifiques à une technologie, voire, à un langage de programmation donné [Quafa, 2006].

## 6. Formalismes de description de patrons :

De manière générale, un patron décrit un "problème" fréquemment rencontré dans un "contexte" particulier ainsi qu'une "solution" générale et consensuelle pour résoudre ce problème [Avgeriou, 2003]. Pour représenter ce triplet {problème, solution, contexte} généralement commun à la majorité des patrons, de nombreux formalismes ont été proposés. Nous entendons par formalisme la structure adoptée par le concepteur de patrons pour

représenter des patrons. Deux classes de formalismes peuvent être distinguées: les formalismes "narratifs" tels que ceux de C. Alexander [Alexander, 1979], et les formalismes "structurés" tels que ceux de M. Fowler [Fowler, 1997] et de P. Coad [Coad, 1997] pour les patrons d'analyse, d'E. Gamma [Gamma, 1995] pour les patrons de conception, ou de S. Ambler [Ambler, 1998] pour les patrons processus. Par opposition aux formalismes narratifs par nature peu structurés et informels, le second type de formalisme offre une meilleure représentation des patrons. Ces formalismes structurés, composés chacun d'un ensemble de rubriques qui leur est propre, sont globalement équivalents dans la mesure où ils expriment tous le triplet {problème, solution, contexte}. Ils diffèrent cependant par le nombre de rubriques proposées et le degré de détail de celles-ci.

Nous nous limitons à présenter ici trois formalismes. Le premier a été introduit par C. Alexander [Alexander, 1979] pour décrire les patrons nécessaires à la construction des bâtiments dans le domaine architectural. Les deux suivants sont ceux que nous considérons comme plus représentatifs des formalismes utilisés pour représenter les patrons dans le domaine de l'analyse, la conception et l'implémentation orientée objet.

### 6.1. Formalisme de C. Alexander :

Selon C. Alexander [Alexander, 1979], un patron comprend cinq rubriques principales:

- **Nom** : un nom ou une phrase courte, familière et descriptive pour identifier le patron, par exemple "Alcôve", "Entrée Principale", "Fenêtres Intérieures", etc.
- **Un ou des exemples** : un/des dessins, diagrammes et/ou descriptions illustrant l'application du patron.
- **Contexte** : le contexte désigne les situations dans lesquelles le patron s'applique.
- **Problème** : cette rubrique offre une description des forces essentielles du patron et des contraintes sous lesquelles il s'applique, ainsi que les interactions entre ces deux éléments.
- **Solution** : cette rubrique indique la manière de résoudre le problème, composée de relations statiques et de règles dynamiques décrivant comment construire les artefacts en accord avec le patron. Souvent, les variantes sont proposées ainsi que les manières d'ajuster la solution en fonction de circonstances. Parfois, la solution nécessite l'utilisation d'un autre patron.

Globalement, ce formalisme de représentation est narratif et peu structuré. Il invite tout lecteur qui cherche à un élément particulier à une lecture complète. En comparaison, les patrons d'analyse, de conception et d'implémentation orientée objet sont souvent très complexes et en incluant le code nécessaire à la réalisation de la solution proposée dans un langage cible. Leur description atteint souvent une dizaine de pages. La représentation de tels

patrons nécessite donc d'être très claire, c'est pourquoi les formalismes de représentation des patrons utilisés de nos jours dans ce domaine sont structurés, bien que leur contenu soit globalement équivalent à celui proposé par C. Alexander.

## 6.2 Formalisme de P. Coad :

P. Coad [Coad, 1997] offre 148 stratégies et 31 patrons destinés à guider un concepteur dans la construction de modèle objet effectifs et complets. En particulier, les quatre dernières stratégies se concentrent sur l'identification de nouveaux patrons : comment découvrir de nouveaux patrons, comment les nommer, comment les raffiner et comment les décrire. Selon P. Coad la description d'un patron nécessite l'inclusion des rubriques suivantes :

- **Nom** : le nom du patron est constitué des noms des différents acteurs composant le patron.
- **Catégorie** : la catégorie du patron précise le type de patron selon son contexte. Elle est choisie parmi cinq types définis par l'auteur : le patron fondamental, les patrons de transaction, les patrons d'agrégation, les patron de plan et les patrons d'interaction. Elle peut éventuellement être une nouvelle catégorie.
- **Gabarit** : c'est une représentation graphique des classes du patron, qui utilise une notation issue de la méthode OMT (*Object Modeling Technique*)<sup>6</sup>.
- **Interaction** : cette rubrique représente l'interaction entre les objets intervenant dans le patron.
- **Exemples d'utilisation** : se sont les instanciations possibles des différentes classes composant le patron.
- **Combinaisons** : se sont les combinaisons possibles avec d'autres patrons.
- **Remarques** : se sont les remarques éventuelles complémentaires.

## 6.3 Formalisme d' E. Gamma :

Le formalisme d'E. Gamma [Gamma, 1995], présenté ci-dessous, est en revanche dédié à la représentation de patrons de type produit. Il comporte 14 rubriques :

- **Nom** : le nom de patron contient succinctement son principe, il permet d'identifier le patron.

---

<sup>6</sup> OMT est une méthode pour la représentation graphique des classes dans un modèle [Gamma, 1995].

- **Classification** : la classification permet d'organiser les patrons selon deux critères. Le rôle qui traduit ce que fait le patron (création, structuration, comportement), et le domaine qui précise si le patron s'applique en général à une ou plusieurs classes ou objets.
- **Intention** : l'intention précise le problème de conception soulève par le patron, c'est une courte déclaration qui répond aux questions suivantes : que fait effectivement le patron de conception ? Quel est sa raison d'être et son but ? Quel cas ou quel problème particulier de conception concerne-t-il ?
- **Alias** : donne d'autres noms connus pour le patron, s'il existe.
- **Motivation** : c'est un scénario d'application du patron posant son problème dans un contexte particulier. La motivation permet de mieux comprendre l'intérêt et la solution du patron.
- **Indications d'utilisation** : les indications d'utilisation reconnaissent des situations dans lesquelles le patron peut être utilisé.
- **Structure** : la structure représente les classes participant dans la solution du patron et montre leurs interactions dans des diagrammes OMT.
- **Constituants** : les constituants définissent les classes et/ou les objets participants ainsi que leurs responsabilités.
- **Collaboration** : la collaboration discute la manière dont les constituants collaborent pour assumer leurs responsabilités.
- **Conséquences** : les conséquences décrivent comment le patron réalise ses objectifs et présente les résultats, compromis et impacts de l'application de la solution du patron.
- **Implémentation** : l'implémentation explique la technique d'implémentation du patron. Elle présente les pièges existants et propose des solutions en fonction du langage utilisé, si elles existent.
- **Exemple de code** : se sont des extraits de programmes, qui illustrent la façon qu'il convient d'employer pour développer le patron en langage C++ ou Smalltalk.
- **Utilisations remarquables** : cette rubrique présente des imitations du patron considéré dans des applications connues (dans des frameworks, par exemple).
- **Patrons apparentés** : cette rubrique référence d'autres patrons utilisant ou utilisés par celui-ci.

Nous remarquons que toutes les rubriques proposées par le formalisme d'E. Gamma sont présentées dans l'un ou l'autre des formalismes de C. Alexander et de P. Coad. La partie "Classification" est rattachée à la rubrique "Catégorie" de P. Coad car le but de ces deux catégories est le même (classer le patron dans une catégorie), mais les catégories proposées par chacun des formalismes sont différentes : P. Coad propose de classer les patrons selon leur contexte (agrégation, interaction, etc) alors que E. Gamma distingue les patrons selon leur nature (créationnel, structurel ou comportemental) et leur portée (une classe, un objet ou une combinaison de classes et d'objets).

### 7. Catalogue de patrons de conception :

Un catalogue de patrons désigne un groupement de patrons répondant à une problématique commune. D'autres termes sont aussi utilisés pour désigner de telles collections de patrons. Nous parlons aussi de "famille" et de "système". Il s'agit dans tous les cas d'une collection de patrons s'adressant à un même domaine dont l'application est assurée par des règles qui permettent de les combiner. Cette collection doit être suffisamment complète et organisée pour parvenir à répondre à la problématique visée. Tout comme le concept de patron, la notion de système ou catalogue a été introduite par C. Alexander qui définit un système de patrons comme : «*une collection de patrons formant un vocabulaire qui permet de comprendre et communiquer les idées* » [Alexander, 1977]. Si un patron est une solution récurrente à un problème dans un contexte donné, alors un système de patrons est une collection de solutions qui coopèrent pour résoudre un problème complexe selon une solution ordonnée pour aboutir à un but prédéfini.

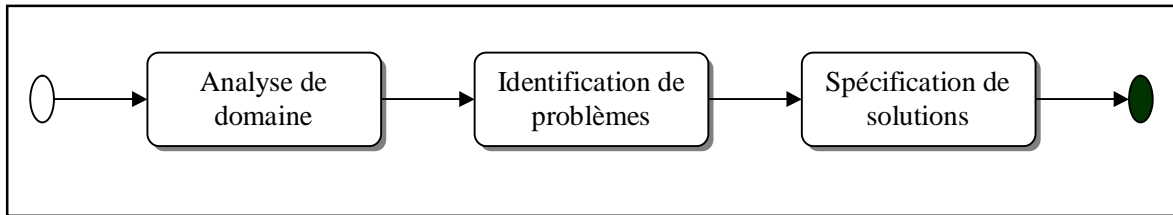
Les patrons tiennent rarement une place à eux tout seuls. Les systèmes de patrons sont un ensemble de patrons, classifiés par catégorie, étroitement reliés entre eux par des contextes identiques. L'intérêt majeur est de faire apparaître facilement la nécessité d'autres patrons et d'exprimer les différents liens qu'il y a entre eux. Un système de patrons bien élaboré doit permettre aux concepteurs d'avoir la liberté d'exprimer leurs problèmes et concevoir eux-même les solutions qui répondent à leurs besoins particuliers dans le contexte où les patrons peuvent être appliqués [Avgeriou, 2003].

### 8. Démarche d'ingénierie de patrons :

Dans cette section, nous présenterons une démarche d'ingénierie des patrons proposée par L. Gzara [Gzara, 2000]. L'ingénierie de patrons peut être abordée de la même manière que l'ingénierie de composants en général. Trois phases sont considérées dans le processus d'ingénierie de patron :

- **Analyse de domaine** : cette phase consiste à identifier la source de connaissances susceptibles de contenir des problèmes réutilisables et de spécifier un référentiel de domaine.

- **Identification de problèmes** : cette phase consiste à rechercher des problèmes réutilisables à partir du référentiel de domaine.
- **Spécification de solution** : une fois les patrons identifiés (en terme de problèmes qu'ils soulèvent), il s'agit de les spécifier (concevoir), c'est-à-dire de proposer une solution au problème soulevé par chaque patron ainsi identifié. La figure 5 illustre les étapes principales du processus d'ingénierie de patrons.

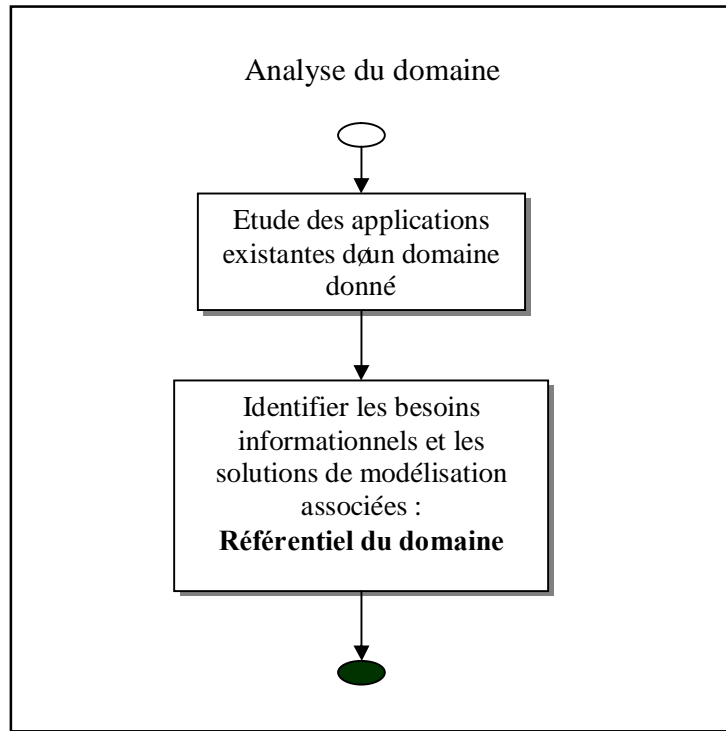


**Figure 5** : Processus d'ingénierie de patrons (tirée de [Gzara, 2000]).

### 8.1 Analyse de domaine :

L'analyse de domaine constitue la première phase dans l'ingénierie de patrons. Elle est définie dans [Hess, 1990] comme l'activité d'identification et de représentation d'information pertinentes pour un domaine donné (une classe de systèmes similaires partageant les mêmes fonctionnalités). Elle fut utilisée depuis plus d'une vingtaine d'années dans des contextes variés : pour appréhender et comprendre un domaine, pour expliquer le fonctionnement d'un domaine et plus récemment pour produire des composants réutilisables.

L'analyse de domaine a pour but d'acquérir les connaissances sur un champ d'application et de les structurer dans un modèle de domaine suffisamment générique pour constituer un cadre de référence pour les développements de tous les systèmes d'un domaine donné. Ce cadre de référence exprime en terme de solutions les besoins souvent pris en compte dans les différents systèmes existants. Ainsi le référentiel obtenu constitue une source de connaissances susceptibles de contenir des besoins (des problèmes) réutilisables.



**Figure 6** : Démarche d'analyse de domaine (tirée de [Gzara, 2000]).

La figure 6 résume la démarche de la phase d'analyse du domaine. Cette démarche se commence par l'étude des applications existantes d'un domaine donné. À la fin de cette phase un référentiel de domaine est obtenu.

## 8.2 Identification de problèmes :

L'objectif de l'approche à base de patrons est de mettre en évidence les divers besoins informationnels souvent rencontrés dans les projets de développements des applications existantes d'un domaine donné (identifier dans la phase d'analyse de domaine) et en suite de proposer une manière de modéliser ces besoins informationnels afin d'obtenir à la fin de la phase de spécification un modèle pour les applications de ce domaine. L'analyse de domaine a produit un référentiel de domaine. Tel qu'il est, il n'a que peu d'intérêt pour un concepteur des applications de domaine qui souhaite travailler par réutilisation. Il est nécessaire d'organiser la connaissance de domaine identifiée dans le référentiel, sous forme des patrons.

L'objectif de chaque patron est de fournir une solution qui correspond à un problème particulier à résoudre. Chaque patron doit, par ailleurs, constituer une unité de raisonnement modulaire : il est le plus indépendant possible des autres patrons, résout un problème standard et peut être réutiliser plusieurs fois.

Ainsi la phase d'identification de problèmes, c'est-à-dire d'identification de patrons (nous pouvons utiliser le nom "identification de problèmes" ou le nom "identification de patrons", ces deux noms sont similaires, parce que si nous avons identifié les problèmes d'un domaine donné nous pouvons identifier les patrons nécessaires pour résoudre ces problèmes) revient à mettre en évidence les différents fragments du modèle à construire par réutilisation de façon à associer un patron à chacun de ces fragments de telle sorte que chaque fragment résolve un problème récurrent dans le développement des applications de domaine.

À l'issue de l'analyse de domaine, le référentiel de domaine élaboré met en évidence l'existence de besoins informationnels "permanents" qui sont souvent gérés dans les applications et de besoins informationnels "variants" qui varient d'une application à une autre selon la spécificité de la plate-forme. Explicitement, dans le référentiel de domaines divers "blocs" de besoins sont distingués. Chaque bloc est une constante du modèle, c'est-à-dire qu'il existe dans toutes les applications de domaine, quelque soit la spécificité de la plate-forme.

La phase d'identification des problèmes (et donc des patrons) revient ainsi à distinguer les divers blocs dans le référentiel obtenu et ensuite à identifier, pour chaque bloc, les divers points de variabilité propres à ce bloc et qui impliquent différentes constructions du fragment de modèle associés au bloc. Une fois les blocs et les points de variabilité sont identifiés, un patron dont l'objectif est de construire le fragment de modèle qui représente le bloc considéré est associé. Ce patron fixe les différents points de variabilité propres au bloc considéré (selon la spécificité de la plate-forme) et dirige ensuite vers divers autres patrons permettant chacun de construire le fragment de modèle représentant le bloc considéré selon un point de variabilité fixé.

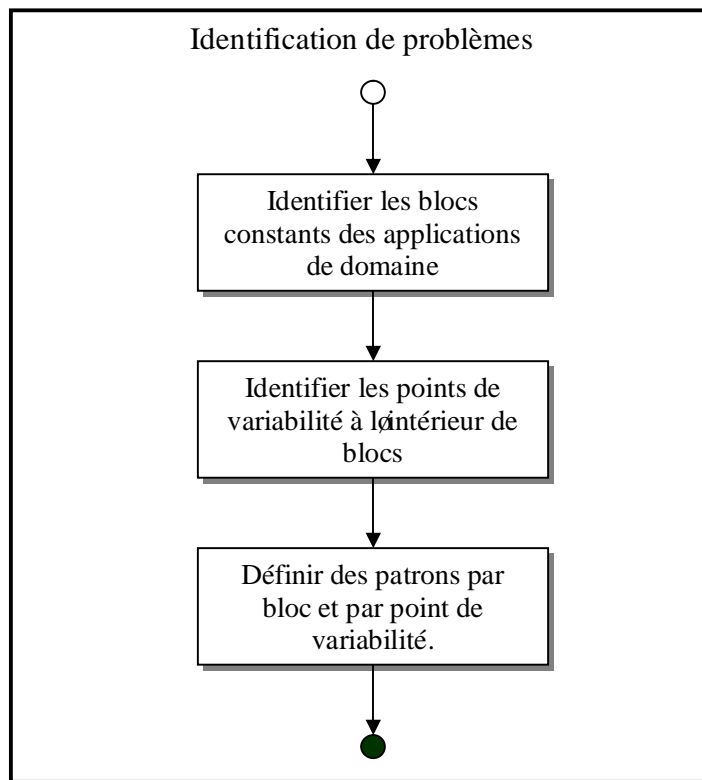


Figure 7 : Démarche d'identification de problèmes (tirée de [Gzara, 2000]).

Cette figure résume les étapes principales de la phase d'identification de problèmes. La démarche de cette phase se commence par l'identification de blocs constants à partir de référentiel de domaine et se termine par l'identification des patrons nécessaires.

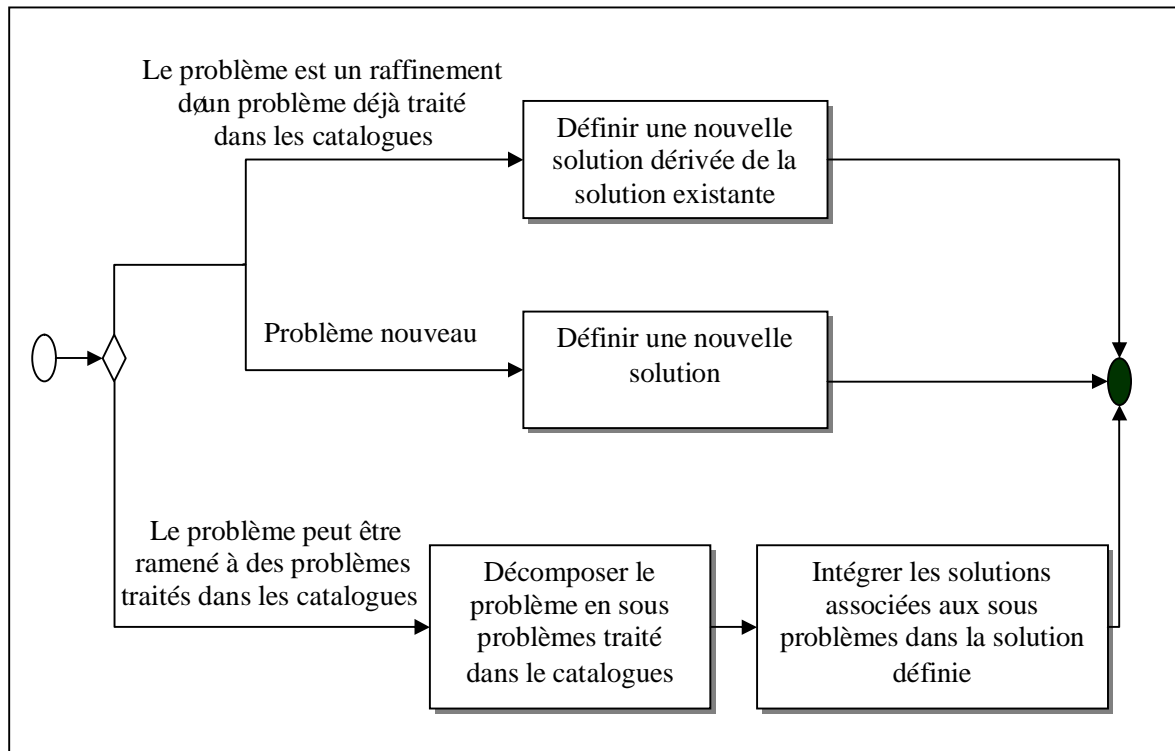
### 8.3 Spécification de solutions :

Une fois les patrons identifiés (en terme de problème à résoudre), la troisième phase de processus d'ingénierie de patron consiste à spécifier les solutions offertes par ces patrons. Lorsque le problème est un problème de modélisation, cette phase est basée sur l'exploitation de solution existantes, disponibles dans des catalogues de patrons de conception proposés dans la littérature, par exemple patron d'E. Gamma [Gamma, 1995]. Dans ce cas, le problème de modélisation considéré est comparé aux problèmes traités dans les catalogues existants.

Plusieurs scénarios peuvent être existés lors de la spécification de solutions :

- *Le problème est un raffinement (un cas particulier) d'un problème déjà traité dans les catalogues de patrons existants : une nouvelle solution est alors définie en se basant sur la solution existante résolvant le problème général.*

- *Le problème est complexe et peut être ramené à des problèmes plus simples traités dans les catalogues de patrons existants* : le problème est alors divisé en des sous problèmes traités dans d'autres catalogues. Les solutions proposées sont adaptées et intégrées dans la solution de patron considéré.
- *Le problème est nouveau* : une nouvelle solution est alors proposée.



**Figure 8** : Démarche de spécification des solutions de patrons (tirée de [Gzara, 2000]).

Cette figure montre des différents scénarios qui peuvent être existés lors de la démarche de la phase de spécifications de solutions et montre comment les solutions peuvent être spécifiées pour chaque scénario.

#### 8.4 Résumé de la démarche :

Dans cette sous section, nous essayons de résumer la démarche de l'ingénierie des patrons. Cette démarche est constituée de trois phases principales qui sont :

- **L'analyse de domaine** : elle consiste à produire un référentiel de domaine recensant d'une part les besoins informationnels souvent rencontrés dans les projets de développement des applications existantes d'un domaine donné (qualifiés de

"connaissance du domaine") et d'autre part les problèmes de modélisation associés à ces besoins (qualifiés de "connaissance de développement").

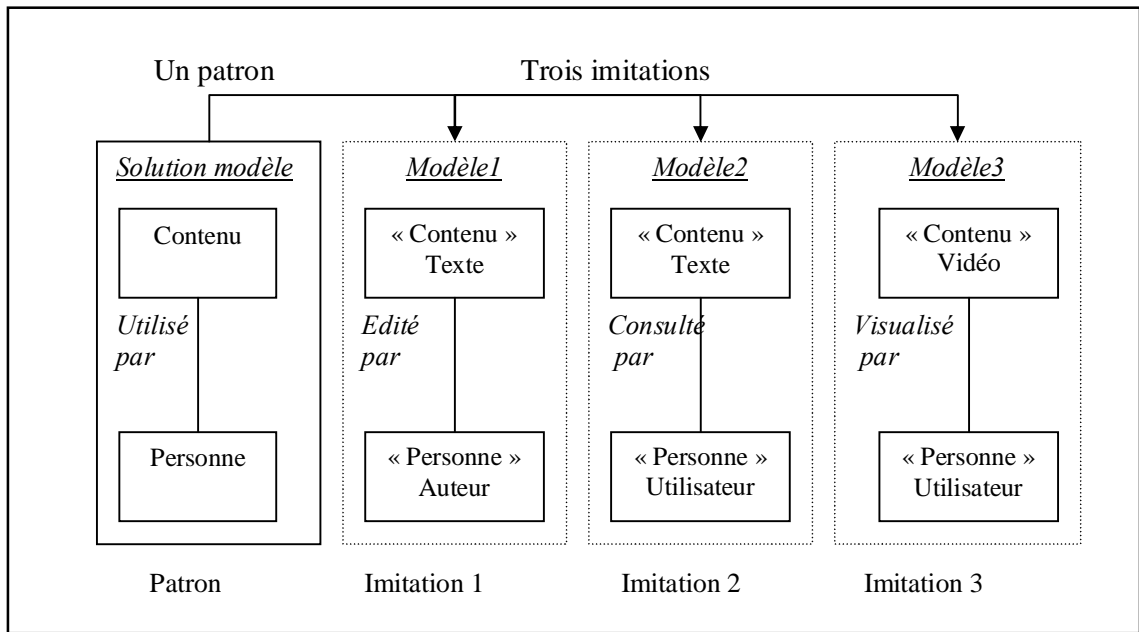
- **L'identification de problèmes ( et de patrons) :** le référentiel obtenu lors de l'analyse de domaine sert de base pour l'identification des divers fragments à construire par réutilisation (divers problèmes) de façon à associer un patron à chacun de ces fragments. À l'issue de cette phase, un ensemble de patrons organisant les problèmes (besoins informationnels), souvent rencontrés dans le domaine donné, est identifié.
- **La spécification des solutions :** une fois les patrons identifiés, cette phase consiste à spécifier les solutions offertes par ces patrons aux problèmes identifiés. Elle est partiellement basée sur l'exploitation des catalogues de patrons proposés dans la littérature.

### 9. Imitation de patrons et intégration d'imitations :

L'ingénierie des systèmes d'information par réutilisation consiste essentiellement en la sélection et la réutilisation de patrons. Bien que ce sujet soit encore du domaine de la recherche, quelques mécanismes pour supporter ce processus sont proposés dans la littérature. D. Rieu propose dans [Rieu, 1999] deux opérations sur les patrons permettant de les réutiliser. Ces opérations sont : l'imitation du patron et l'intégration d'imitations.

#### 9.1 Imitation de patrons :

L'imitation d'un patron concerne l'imitation d'une solution à un problème analogue. Elle porte sur les diagrammes de classes proposés par le patron et peut être guidée par la solution. Un patron n'est pas directement instancié au sens de l'instanciation objet. Le concepteur doit donc le dupliquer puis l'adapter à un contexte donné. L'adaptation consiste essentiellement à renommer des classes et des propriétés. Elle peut se poursuivre par l'ajout ou la suppression de propriétés. En imitant un diagramme, les classes (respectivement les associations et les propriétés) obtenues par duplication des éléments de ce diagramme peuvent être figées avec les noms des classes (respectivement les associations et les propriétés) de patron avant la duplication [Rieu, 1999].

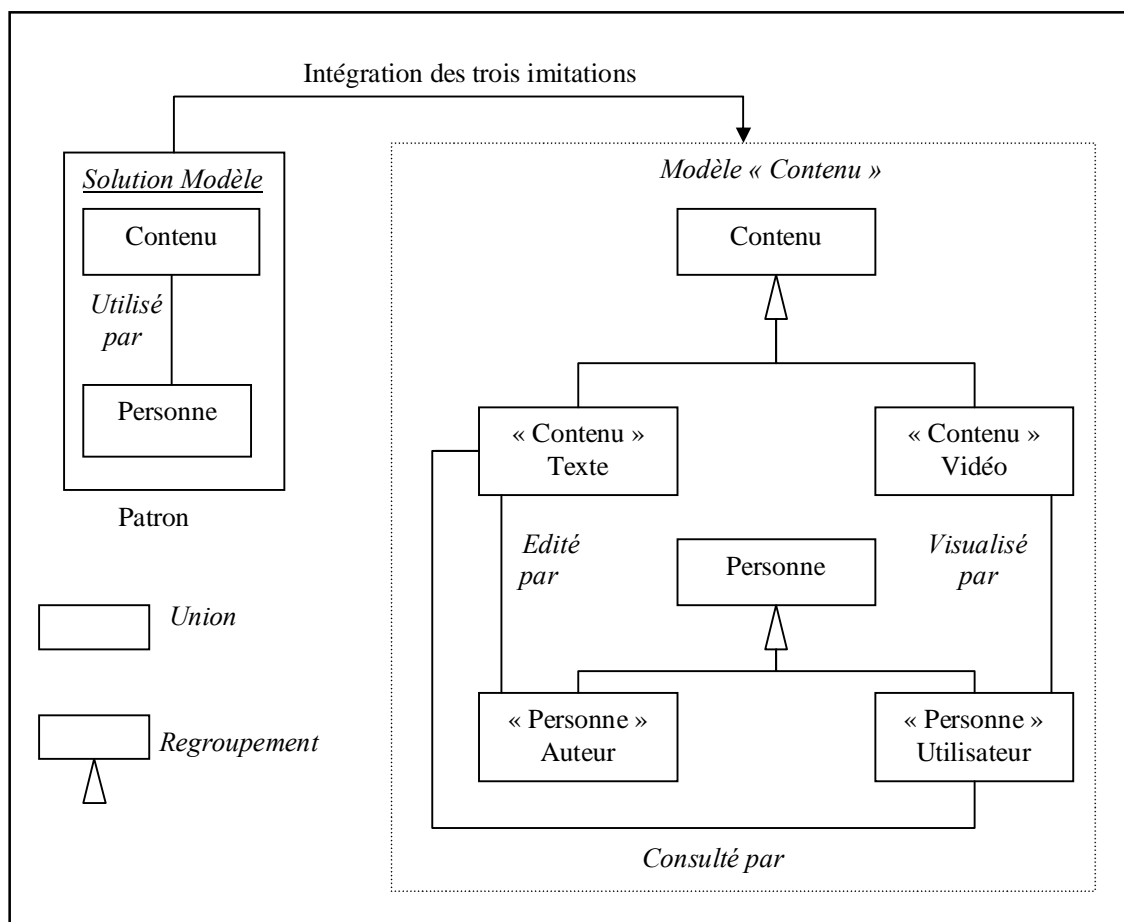


**Figure 9 :** Exemples d'imitations d'un patron.

Pour préciser l'opération d'imitation d'un patron, nous avons pris un exemple de patron qui associé à chaque contenu les personnes qui le manipulent. La figure 9 illustre trois imitations possibles de ce patron, le contenu peut être un "Texte" ou une "Vidéo" et le personne peut être un "Utilisateur" ou un "Auteur". Les associations possibles entre les classes de ces imitations sont : "Un texte est édité par un auteur", "Un texte est consulté par un utilisateur" et "Une vidéo est visualisée par un utilisateur". Ces trois imitations du même patron mettent en évidence que l'opération d'imitation n'est ni une instantiation, ni une spécialisation de modèle mais essentiellement une différenciation dirigée par le contexte.

## 9.2 Intégration d'imitations :

Le résultat de l'imitation de plusieurs patrons ou de plusieurs imitations du même patron correspond à plusieurs sous modèles qu'il s'agit d'intégrer afin de disposer d'un modèle cohérent du système à concevoir. L'intégration de classes consiste essentiellement à unir et à regrouper les classes. "L'union" consiste à remplacer deux classes par une seule, qui regroupe alors l'union des propriétés des deux classes initiales. Le "regroupement" de classes consiste à créer une super-classe à l'ensemble obtenu par duplication d'une même classe du diagramme proposé par le patron, portant le nom de cette classe imitée [Rieu, 1999].



**Figure 10 :** Intégration des imitations dans un patron.

La figure 10 illustre l'intégration des trois imitations présentées dans la figure 9. Pour intégrer ces trois imitations, nous avons créé une super-classe "Contenu" et une super-classe "Personne", puis nous avons fusionné les deux classes "Texte" (des modèles 1 et 2) présentées dans la figure 9 dans une seule classe "Texte" qui regroupe les propriétés des deux classes précédentes. Ainsi nous avons fusionné les deux classes "Utilisateur" (de modèle 2 et 3) présentées dans la figure 9 dans une seule classe "Utilisateur" qui regroupe les propriétés des deux classes précédentes.

### 10. Quelques applications de patrons de conception :

L'approche à base des patrons a montré son efficacité dans le développement d'applications et de logiciels à divers domaines. Dans cette section, nous essayons de présenter quelques applications de ces patrons :

- Patrons de conception pour les applications hypermédia :** Dans [German, 2000], 51 patrons de conception spécifiques aux applications hypermédia sont mentionnés, chaque patron est une solution à un problème rencontré pendant la conception de telle application. L'auteur qui cherche à trouver une solution à une situation donnée, doit en premier temps, analyser cette liste entièrement pour sélectionner les patrons nécessaires à la résolution de cette situation. Les patrons dans cette liste sont classifiés et organisés en un système (catalogue de patrons) pour être plus facile à utiliser.
- Patrons de conception pour la duplication de composants :** les nouveaux environnements d'exécution présentent de nouveaux défis aux solutions de duplication : les facteurs comme la communication, l'échelle mondiale ou la mobilité nécessitent des mécanismes de gestion flexible et adaptable. Dans un projet de recherche "SARDES", V. Marangozova [Marangozova, 2002] propose une approche prometteuse de gestion adaptable de l'aspect de duplication basée sur les principes du paradigme composant et la définition de patrons de duplication. La solution est définie dans le contexte particulier des composants CORBA (*Common Object Request Broker Architecture*)<sup>7</sup> et d'une implémentation existante du modèle.
- Patrons de conception pour l'ingénierie des systèmes d'information par réutilisation :** H. Quafa propose dans sa thèse de doctorat [Quafa, 2006] une approche de patrons de conception à base d'aspect (tirés du domaine de la POA (*Programmation Orientée Aspect*) [Pawlak, 2004]) pour l'ingénierie des systèmes d'information. Cette approche offre une nouvelles solutions pour les patrons de conception par objets. H. Quafa a choisi d'implémenter, à titre de validation, les patrons de conception d'E. Gamma [Gamma, 1995] à l'aide des mécanismes issus des langages de programmation relevant de la technologie "Aspect". Il a choisi ce catalogue de patrons parce qu'il est l'un des plus connus et utilisés de nos jours. Dans leur approche du problème, il considère ce catalogue de patrons essentiellement comme un catalogue de problèmes récurrents en conception par objets plutôt que comme un catalogue de solutions. En effet, les solutions des patrons ne sont décrites dans [Gamma, 1995] qu'en termes de concepts "Objets", et leur propos est de tirer une partie des nouvelles possibilités offertes par les mécanismes et les concepts "Aspects".
- Patrons de conception pédagogiques :** C. Olavo [Olavo, 2007] propose une démarche de conception pour les EIAH destinées à l'e-formation, qui s'appuie sur le concept de patrons pédagogiques de conception. Cette démarche parte d'un constat quant aux difficultés du processus de conception lui-même, et aux risques de coupure qu'il peut y avoir entre pédagogues et développeurs de logiciels. Pour améliorer la capacité des équipes pédagogiques à pouvoir spécifier, et même implémenter, des scénarios pédagogiques, dans une approche basée sur les patrons, cette démarche proposée permet de garder côte à côte et en coïncidence les descriptions informelles, et

<sup>7</sup> CORBA est une spécification définie par l'OMG (*Object Management Group*) qui regroupe de nombreuses entreprises du monde informatique.

les descriptions plus formelles et normatives, et d'offrir la possibilité d'effectuer des allers-retours à toutes les phases du cycle de vie du dispositif pédagogique.

- **Patrons pour la conception des scénarios d'apprentissage :** J. Pernin et D. Rieu proposent dans leur projet de recherche [Pernin, 2009] d'étudier les apports de l'ingénierie des systèmes d'information (via les notions de patrons d'ingénierie et d'ingénierie des besoins) pour l'automatisation du processus de conception et d'exécution des scénarios d'apprentissage en EIAH. Cette proposition est à la croisée de deux domaines importants de la recherche en informatique : l'ingénierie des EIAH d'une part, et l'ingénierie des systèmes d'information d'autre part.

À partir de ce que nous avons présenté dans cette section, nous pouvons dire que l'approche basée sur les patrons a montré son efficacité dans plusieurs domaines. Elle peut être appliquée dans n'importe quel domaine, et elle n'est pas spécifique à un domaine particulier.

## 11. Conclusion :

L'approche à base des patrons représente aujourd'hui un apport capital pour le développement de logiciels à moindre coût. Elle permet de réutiliser lors du développement d'une nouvelle application des techniques éprouvées depuis le niveau de l'expression des besoins jusqu'à celui de l'implémentation dans un système cible.

Nous avons vu dans la section 10, que les patrons de conception sont exploités dans le domaine des applications hypermédia. À partir de notre étude, nous avons observé que malgré le grand nombre des patrons de conception proposées dans ce domaine [German, 2000], il n'existe pas un catalogue complet de patron qui permet à un auteur de concevoir une application hypermédia et qui prend en considération toutes leurs caractéristiques. Il est important donc d'avoir un catalogue de patrons dédiés à ces applications, ce catalogue doit contenir les patrons de base qui permettent de résoudre les problèmes de conception dans le domaine des applications hypermédia. C'est l'objectif de notre travail et dans le chapitre suivant, nous essayons de proposer ce catalogue de patrons de conception.

## Chapitre 3

# Conception d'un catalogue de patrons pour les applications hypermédia

### 1. Introduction :

Dans le premier chapitre de ce mémoire, nous avons exposé les caractéristiques des systèmes hypermédia. Et dans le deuxième chapitre, nous avons présenté la notion de la réutilisation, et par la suite nous avons dressé un panorama des patrons de conception comme une approche qui se base sur cette notion. Dans ce chapitre nous allons nous intéresser à la présentation de l'architecture globale de notre proposition et à la description des patrons du catalogue que nous avons proposé pour la conception des applications hypermédia. Nous présenterons, en premier temps, une démarche pour l'ingénierie des patrons de conception basée sur des techniques favorisant leur réutilisabilité, tout en respectant les principes essentiels d'abstraction et de variabilité. Ensuite nous présenterons les divers patrons constituant le catalogue proposé pour le développement des applications hypermédia.

### 2. Démarche d'ingénierie de patrons de conception pour les applications hypermédia:

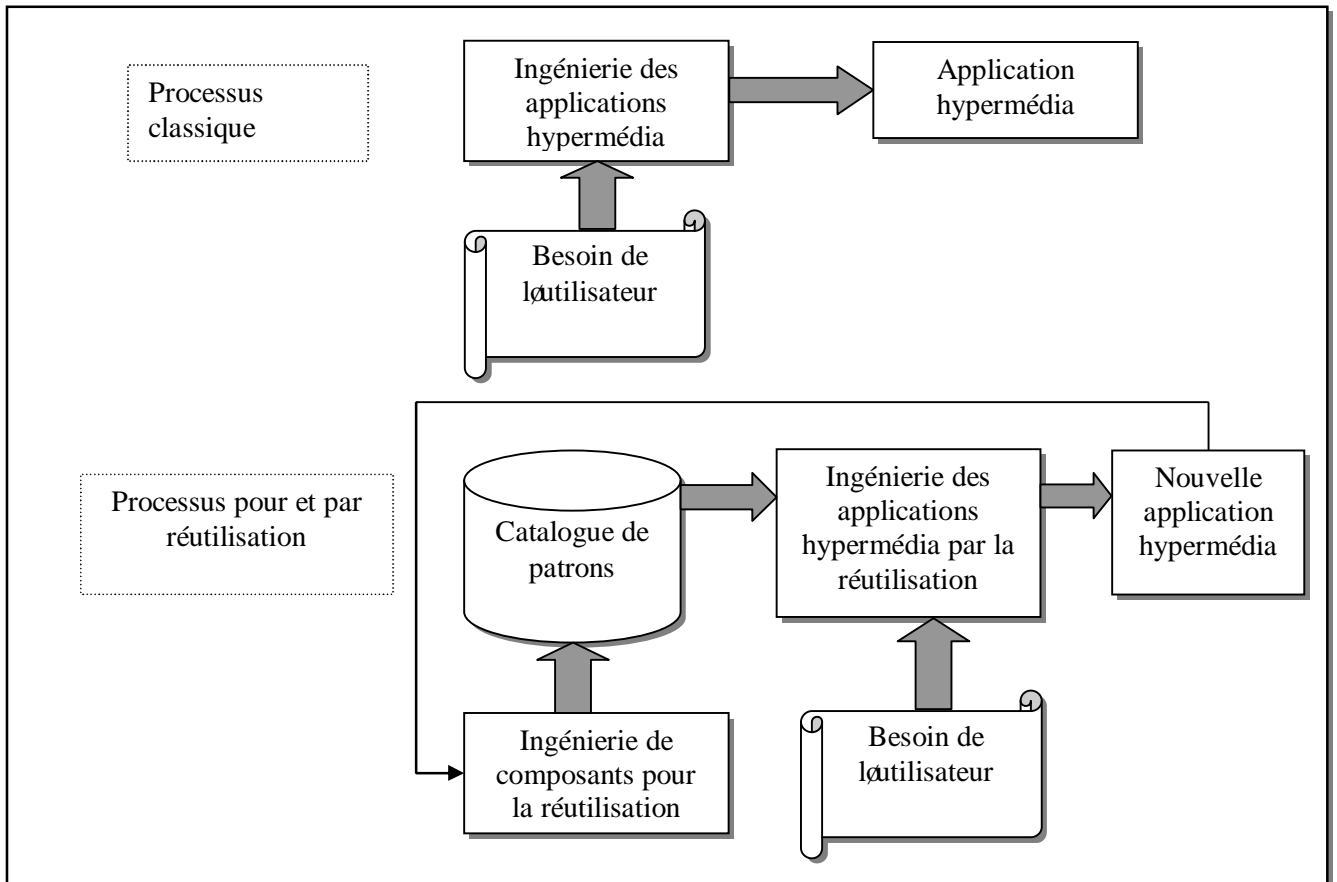
Dans cette section, nous présenterons la démarche que nous avons proposée pour l'ingénierie des patrons de conception dédiés au développement des applications hypermédia. L'ingénierie des composants réutilisables reste un domaine de recherche récent, tout comme le domaine d'ingénierie à base des patrons. Ce qui justifie, l'absence d'approches d'ingénierie des patrons dédiées à la conception des applications hypermédia. Dans notre mémoire nous avons adopté une démarche pour l'ingénierie de ces composants. Des adaptations ont été introduites sur la démarche initiale proposée par L. Gzara [Gzara, 2000]. Cette démarche a été proposée pour l'ingénierie des patrons pour les systèmes d'information produit. Nous avons adapté cette démarche pour être spécifique à l'ingénierie des patrons pour le développement d'applications hypermédia.

#### 2.1 Processus de la démarche :

L'adoption d'une approche d'ingénierie des applications hypermédia, basée sur la réutilisation des composants, nécessite d'abord de produire les composants à réutiliser et ensuite de les appliquer d'une façon systématique. Dans notre approche, l'introduction des composants réutilisables, qui sont les patrons de conception, dans le processus de développement des applications hypermédia fait donc émerger deux nouveaux processus complémentaires:

**Ingénierie dédiée à l'ingénierie des patrons :** ce processus revient à identifier d'abord les problèmes récurrents lors de développement des applications hypermédia et ensuite à spécifier les différents patrons proposant des solutions aux problèmes identifiés. Ce processus permet d'ajouter des nouveaux patrons.

- Processus par la réutilisation dédié au développement des applications hypermédia :** ce processus revient à offrir des moyens pour identifier les problèmes à résoudre, sélectionner les patrons qui résolvent ces problèmes et enfin adapter les solutions proposées au contexte dans le quel se trouve le concepteur.



**Figure 11:** Comparaison entre le processus classique et les processus pour et par la réutilisation.

Dans la figure 11, nous avons fait une comparaison entre le processus classique de développement des applications hypermédia et les deux nouveaux processus que nous avons utilisé dans notre approche. Dans le premier type de développement, la conception d'une nouvelle application hypermédia parte uniquement sur les besoins des utilisateurs. Dans les deux nouveaux processus utilisés, la conception d'une nouvelle application hypermédia parte sur les besoins des utilisateurs et sur le catalogue de patrons proposé dans la littérature. C'est-à-dire que dans notre approche, l'auteur d'une application



Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

à trouver des solutions à ces situations données dans le catalogue de patrons. Nous avons dit que ces deux processus sont complémentaires parce que le processus "pour réutilisation" alimente le processus "par réutilisation" afin de tenir compte régulièrement de nouveaux problèmes ou de nouvelles solutions à des problèmes existants.

Dans notre approche, nous avons distingué deux types de patrons, nous appelons le premier type "Patrons solutions" et nous appelons le deuxième type "Patrons manquants" :

- **Patrons solutions** : ce sont les patrons existants dans le catalogue de patrons qui permettent de résoudre une partie des situations rencontrées par l'auteur de développement d'applications hypermédia.
- **Patrons manquants** : ce sont les patrons qui ne se trouvent pas dans le catalogue, mais ils sont nécessaires à la résolution de la partie restante des situations rencontrées par l'auteur de développement d'applications hypermédia. Ces patrons doivent donc être ajoutés au catalogue de patrons.

## 2.2 Présentation de la démarche :

Dans cette sous section, nous nous sommes intéressés de présenter la méthode proposée dans notre travail. Dans le début de cette section, nous avons mentionné que notre démarche est basée sur une démarche proposée par L. Gzara [Gzara, 2000]. Cette dernière a été présentée dans la section 8 du chapitre 2. Dans cette sous section, nous décrirons les adaptations et les spécifications appliquées sur cette démarche. Trois phases principales sont considérées dans la démarche d'ingénierie de patrons pour le développement des applications hypermédia: analyse de domaine hypermédia, identification des patrons hypermédia et spécification de solutions.

### 2.2.1 Analyse de domaine hypermédia:

Pour les applications hypermédia, l'analyse de domaine consiste à spécifier un référentiel de domaine spécifique à ce type d'applications. Nous appelons ce référentiel un "Référentiel hypermédia". Les solutions exprimées dans ce référentiel ne sont autres que des modèles d'applications hypermédia devant être élaborés par réutilisation de patrons.

L'analyse de domaine hypermédia revient :

- Dans un premier temps à explorer les besoins informationnels fréquemment exprimés par les experts et les utilisateurs des applications hypermédia durant la phase d'analyse. Il s'agit des connaissances relatives aux différents concepts manipulés dans les applications hypermédia, en d'autres termes "les connaissances du domaine".
- Dans un second temps à explorer les besoins de modélisation souvent exprimés par les concepteurs des applications hypermédia pour spécifier les connaissances de domaines identifiées auparavant. Il s'agit donc d'explorer "les connaissances de développement".



Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

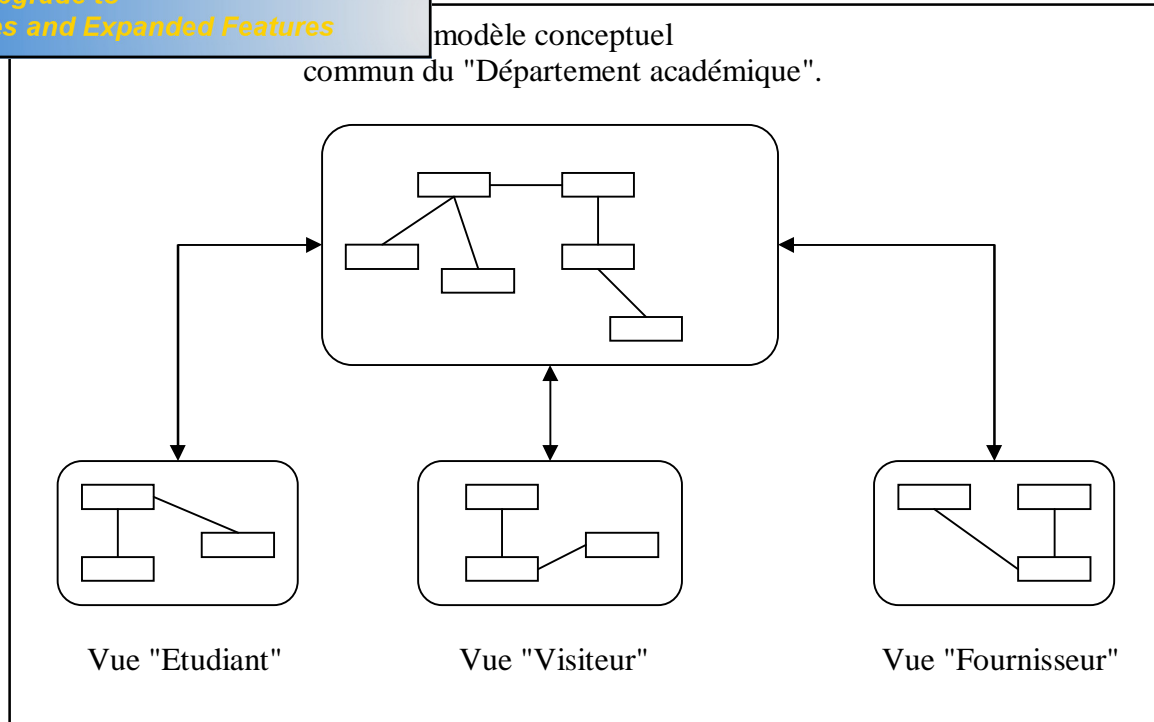
référentiel de domaine hypermédia qui :

- Propose un cadre terminologique et sémantique des connaissances de domaine. Ce cadre doit permettre d'identifier et de classifier l'ensemble des objets ou concepts gérés dans les applications hypermédia.
- Identifie les problèmes de modélisation associés.
- Propose un modèle de référence hypermédia qui résout les problèmes de modélisation en structurant les concepts (et notamment les relations entre ces concepts) et en explicitant leur comportement (à l'aide des connaissances de développement).

Maintenant, nous appliquons cette analyse de domaine hypermédia sur une application hypermédia existante pour identifier le référentiel hypermédia. Nous avons utilisé une application hypermédia pour la gestion d'un département académique présentée dans [Schwabe, 1995] pour identifier notre référentiel. Nous avons choisi cette application à cause de leur simplicité à comprendre. Ce département académique se compose d'un département général d'informations, des enseignants, des cours, des travaux de recherche et des publications.

Les types des utilisateurs possibles de l'application "Département académique" sont : les visiteurs, les étudiants et les fournisseurs. Il est clair que chaque type d'utilisateur accède cette application de façon différente que les autres, par exemple : les visiteurs accèdent le département général pour obtenir les renseignements qu'ils demandent, et les fournisseurs accèdent les travaux de recherche pour fournir les supports financiers à des travaux choisis. Pour chaque type d'utilisateur, nous associons donc une vue navigationnelle. Cette dernière désigne un ensemble des objets hypermédia spécifiques à un type d'utilisateur. Dans cette application, les vues navigationnelles possibles sont : la vue "Etudiant", la vue "Visiteur" et la vue "Fournisseur".

Le modèle de cette application est basé sur la construction d'un modèle conceptuel commun pour représenter les objets de cet "Département académique" et les relations possibles entre ces objets. Les vues navigationnelles qui prennent en compte le profil de chaque utilisateur sont dérivées à partir de ce modèle commun. Le modèle que nous pouvons inférer pour cette application est représenté par la figure suivante :

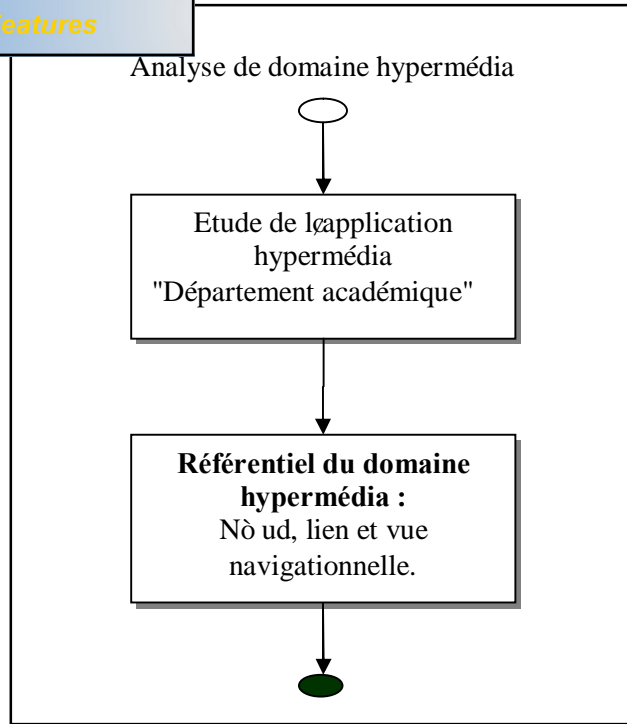


**Figure 12 :** Modèle conceptuel commun et leurs vues navigationnelles de l'application "Département académique" (tirée de [Schwabe, 1995]).

À partir de la figure 12, nous pouvons retirer les connaissances hypermédia pour construire notre référentiel comme suit :

- Les objets de modèle conceptuel sont les nœuds de l'application hypermédia (les objets sont représentés dans cette figure comme des petits rectangles).
- Les relations entre ces objets sont les liens de cette application hypermédia (la relation entre deux objets est représenté par une ligne entre deux petits rectangles).
- Les différents types d'utilisateurs sont les vues navigationnelles de cette application hypermédia (les différents types d'utilisateurs sont représentés comme des grands rectangles qui regroupent les objets de modèle).

Notre référentiel hypermédia se compose donc de trois connaissances de domaine principales qui sont : le nœud, le lien et la vue navigationnelle.



**Figure 13 :** Démarche d'analyse de domaine hypermédia.

La figure 13 résume les différentes étapes de la phase d'analyse de domaine hypermédia. Cette phase se commence par l'étude de l'application hypermédia "Département académique". À la fin de la phase, référentiel hypermédia est obtenu.

### 2.2.2 Identification des patrons hypermédia:

Cette phase débute par le référentiel hypermédia identifié dans la phase précédente. Elle permet d'organiser les connaissances hypermédia identifiées dans ce référentiel sous forme des patrons, nous appelons ces patrons "Patrons hypermédia". L'objectif de chaque patron hypermédia est de fournir une solution qui correspond à un problème particulier à résoudre.

Premièrement, nous rappelons les étapes principales de cette phase puis nous les appliquons sur le référentiel hypermédia.

La première étape d'identification des problèmes (et donc des patrons) consiste à distinguer les divers blocs dans le référentiel obtenu et ensuite à identifier, pour chaque bloc, les divers points de variabilité propres à ce bloc et qui impliquent différentes constructions du fragment de modèle associés au bloc. Une fois les blocs et les points de variabilité identifiés, nous associons à chaque bloc un patron dont l'objectif est de construire le fragment de modèle qui représente ce bloc. Ce patron fixe les différents points de variabilité propres au bloc considéré et dirige ensuite vers divers autres patrons permettant chacun de construire le fragment de modèle représentant le bloc considéré selon un point de variabilité fixé.

se compose de trois concepts principaux qui sont : le nœud, le lien et la vue navigationnelle.

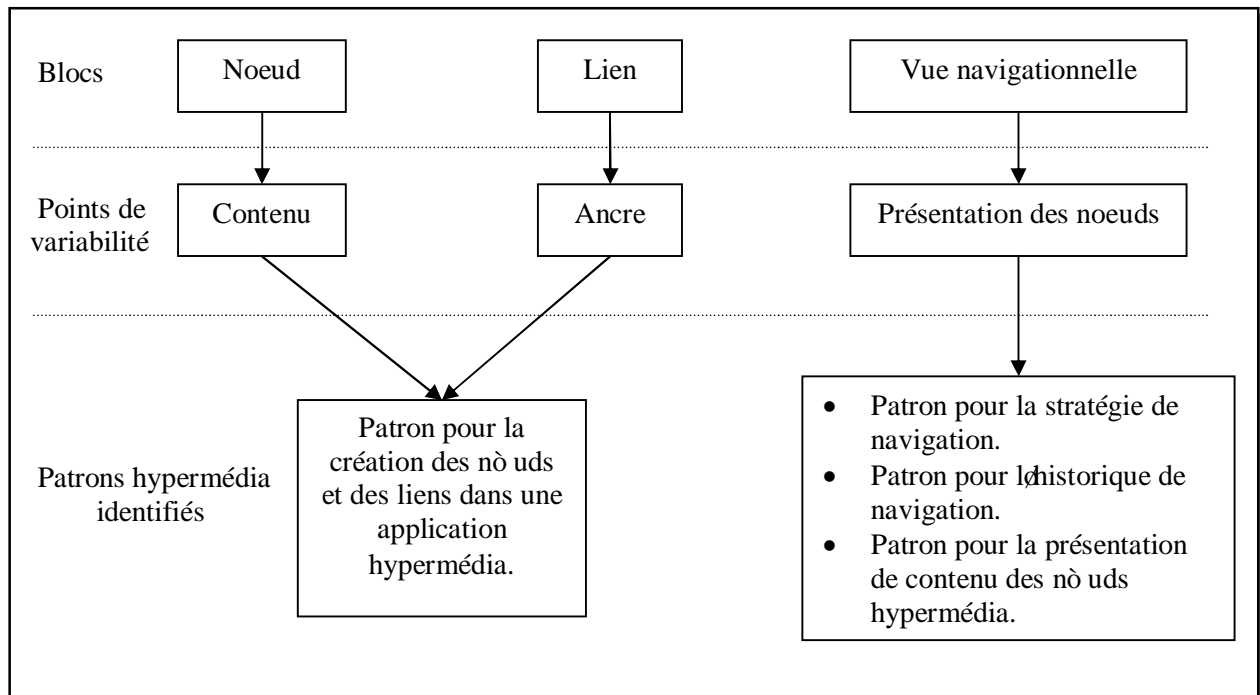
Dans le chapitre 1, nous avons vu qu'une application hypermédia est un ensemble des nœuds reliés entre eux par des liens. Un nœud est l'unité minimale de l'information et le lien constitue le moyen principal d'organisation de l'ensemble de ces nœuds pour garantir la notion de navigation. Les utilisateurs d'une application hypermédia permettent d'identifier les différentes vues navigationnelles de cette application. Les blocs constants dans notre référentiel sont donc : le bloc "Nœud", le bloc "Lien" et le bloc "Vue navigationnelle". Nous présenterons les points de variabilité pour chacun de ces blocs comme suit :

- **Pour le bloc "Nœud" :** de manière générale, les points de variabilité sont les besoins informationnels qui varient d'application à une autre selon la spécificité de la plateforme. Dans les applications hypermédia, le contenu d'un nœud est leur point de variabilité, parce que ce contenu varie d'une application hypermédia à une autre. Le contenu d'un nœud permet d'identifier leur taille. Cette dernière est en relation avec la granularité utilisée par l'application hypermédia pour diviser le contenu entre ces différents nœuds.
- **Pour le bloc "Lien" :** le point de variabilité principal que nous pouvons retirer à partir de bloc lien est leur direction. Un lien peut être uni ou bidirectionnel, la direction d'un lien est identifiée par l'ancrage, c'est-à-dire un lien est bidirectionnel s'il existe une ancre dans le nœud destination permettant de retourner au nœud source. Sinon, le lien est unidirectionnel.
- **Pour le bloc "Vue navigationnelle" :** une vue navigationnelle est un ensemble des nœuds intéressés par un type d'utilisateur. Le point de variabilité que nous pouvons retirer à partir de ce bloc est la manière de représentation de contenu d'un nœud à un utilisateur identifié. Chaque utilisateur a un profil qui doit être mis en considération lors de la construction des vues navigationnelles. L'ensemble des nœuds intéressés par un utilisateur et la manière de représentation de ces nœuds sont identifiés à partir de leur profil.

Les patrons hypermédia par bloc et par point de variabilité sont identifiés de la manière suivante :

- **Pour les deux blocs "Nœud" et "Lien" :** nous avons besoin d'un patron qui permet de construire deux ces éléments. Ce patron hypermédia doit prendre en considération les points de variabilité de deux ces blocs, c'est-à-dire le contenu pour le nœud et l'ancrage pour le lien.
- **Pour le bloc "Vue navigationnelle" :** nous avons besoin d'un patron qui permet d'identifier la stratégie de navigation pour chaque utilisateur. Nous avons aussi besoin d'un autre patron qui permet de garantir la présentation la plus adéquate pour chaque type d'utilisateur. L'un des besoins que l'application hypermédia doit offrir à un utilisateur est l'historique de

montrer leur parcours dans l'espace hypermédia. Nous avons donc besoin d'un patron permettant de construire cet historique de navigation.



**Figure 14:** Démarche d'identification des patrons hypermédia.

La figure 14 résume les étapes de la démarche d'identification des patrons hypermédia. Une fois ces patrons sont identifiés en termes de problèmes à résoudre, la troisième phase de notre démarche consiste à spécifier les solutions offertes par ces patrons.

### 2.2.3 Spécification de solutions :

La phase de spécification des solutions consiste à spécifier les solutions offertes par les patrons hypermédia (identifiés dans la phase précédente de notre démarche) aux problèmes identifiés. Cette phase est basée sur l'exploitation des catalogues de patrons proposés dans la littérature. Cette exploitation peut être par une adaptation ou une réutilisation directe de ces patrons pour les problèmes déjà traités dans les catalogues de conception existants. Elle est aussi basée sur la construction des nouveaux patrons pour les nouveaux problèmes.

Les patrons de conception proposés dans la littérature pour la conception des applications hypermédia sont beaucoup et variés. Ces patrons sont regroupés et organisés dans un travail de recherche [German, 2000] sous une forme d'une liste des patrons hypermédia. Ces derniers sont classés en catégories selon leurs objectifs pour faciliter leur utilisation par les auteurs de développement des applications hypermédia.



**PDF Complete**

Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

pour construire notre catalogue de patrons hypermédia

sont les suivantes .

- Analyser la liste des patrons hypermédia proposés dans la littérature.
- Prendre le premier patron identifié dans la phase d'identification de patrons hypermédia.
- Comparer l'intention de ce patron avec les intentions des patrons de la liste.
- S'il existe un patron dans la liste qui a une intention identique à l'intention de notre patron, le patron trouvé sera réutilisé directement. Sinon, s'il existe un patron dans la liste qui a une intention proche à l'intention de notre patron, le patron trouvé sera adapté selon le problème à résoudre. Sinon, un nouveau patron doit être créé.

Ce processus doit être répété jusqu'à le dernier patron. Les patrons identifiés pour construire notre catalogue sont présentés en détail dans la section suivante.

### 3. Structure du catalogue :

Nous présenterons dans cette section une vue d'ensemble des patrons proposés. Nous concentrerons notre effort sur la description du problème, du contexte et de la solution. Les patrons proposés sont organisés selon trois processus principaux qui sont :

- La construction des patrons manquants dans la liste proposée dans la littérature.
- L'adaptation des patrons proposés dans la littérature.
- La réutilisation des patrons proposés dans la littérature.

Premièrement, nous précisons le formalisme utilisé pour la description des patrons de notre catalogue.

#### 3.1 Formalisme utilisé pour la description des patrons de catalogue :

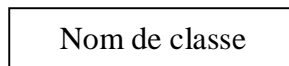
Les patrons dans le catalogue doivent être représentés dans un format standard et unique pour être plus efficace. Dans le chapitre 2, nous avons présenté les différents formalismes pour la description des patrons de conception. Nous avons choisi le formalisme de E. Gamma [Gamma, 1995] pour la description des patrons de notre catalogue parce que ce formalisme est très complet et fournit des directives pour mettre en place les patrons dans les applications.

Le formalisme de E. Gamma utilise la méthode OMT pour la représentation de la structure d'un patron, mais nous utilisons le diagramme UML pour la représentation des classes composantes de chaque patron parce que le langage UML devient le standard de modélisation maîtrisé par plusieurs développeurs.

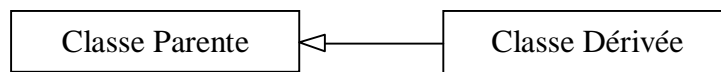
Dans notre catalogue nous ne souhaitons pas traiter des exemples d'implémentation, parce que les technologies changent très rapidement et si nous essayons de proposer des technologies spécifiques, les patrons deviendront bientôt dépassés. C'est pour cette raison que nous avons utilisé seulement les 10 premières rubriques de ce formalisme car les autres rubriques spécifient l'implémentation du patron. La même pratique est utilisée dans des patrons proposés par G. Rossi [Rossi, 1997].

Dans les parties suivantes, nous présenterons le patron que nous avons créé, les patrons que nous avons adaptés et le patron que nous avons réutilisé. Avant la présentation du catalogue, nous présentons quelques notations UML utilisées dans la description des patrons composants de ce catalogue.

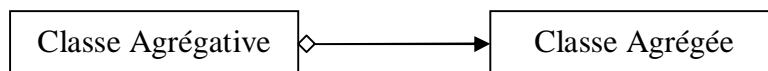
- Une classe est représentée par un rectangle avec le nom de la classe comme suit :



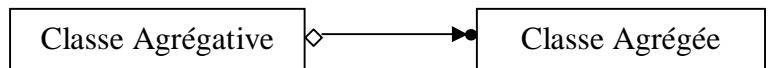
- L'héritage de classes est représenté par une ligne traversant un triangle comme suit :



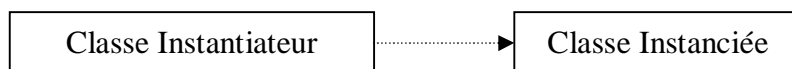
- L'agrégation (ou la composition) est représentée par une flèche dotée d'un losange à sa base comme suit :



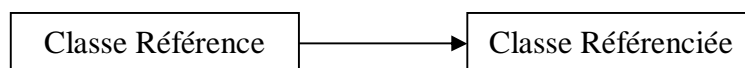
- La composition multiple est représentée comme l'agrégation avec un petit cercle noir à la fin de la flèche comme suit:



- La dépendance (ou l'instanciation) est représentée par une flèche en pointillés comme suit:



- La délégation (ou la relation de référence) est représentée par une flèche pleine comme suit:



Dans le chapitre 4 de ce mémoire, nous essayons de préciser comment les diagrammes de classe UML, utilisés dans la représentation de la structure des patrons de catalogue, peuvent être transformés en une description XML.



**PDF Complete**  
 Your complimentary use period has ended.  
 Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

**"Création d'applications hypermédia" :**

Nous avons mentionné dans la section précédente que nous avons analysé la liste des patrons proposés dans la littérature [German, 2000] pour identifier les patrons nécessaires à la construction de notre catalogue. Dans la phase d'identification des patrons hypermédia, nous avons trouvé que nous avons besoin du patron permettant de créer les nœuds et les liens dans une application hypermédia. Dans la liste des patrons proposés [German, 2000], nous avons trouvé plusieurs patrons spécifiques à la création des nœuds et des liens, mais ces patrons ne montrent pas comment ces nœuds et ces liens sont organisés dans une application hypermédia, leurs intentions basés seulement sur les types et les règles de construction de ces nœuds et ces liens.

Le patron que nous avons proposé, permet de concevoir une application hypermédia. Il montre comment ses composants (nœud, contenu, lien, ancre) sont organisés, il précise aussi les relations entre ces composants. L'auteur de développement d'une application hypermédia peut utiliser ce patron de manière très facile, parce que leurs constituants sont très simples à comprendre. Nous présentons maintenant la spécification de ce patron en utilisant le formalisme de E. Gamma [Gamma, 1995] comme nous l'avons précisé précédemment.

<p><b>Nom :</b> Création d'applications hypermédia.</p>
<p><b>Intention :</b></p> <p>Comment assurer la conception, la structuration des différents composants d'une application hypermédia.</p>
<p><b>Motivation :</b></p> <p>Le champ des systèmes hypermédia est très complexe et évolutif. Pour cela, il existe plusieurs définitions d'une application hypermédia. Elle peut être vue comme un matériau composite des documents et des connaissances sur ces documents. Elle peut être aussi vue comme un ensemble des nœuds reliés entre eux par des liens. Nous avons donc besoin d'un patron de conception qui permet de faire la création d'une application hypermédia de façon simple et avec une structure bien définie.</p>
<p><b>Indication d'utilisation :</b></p> <p>Ce patron de conception considère l'application hypermédia comme un ensemble des nœuds reliés entre eux par des liens. Il est utilisé lorsque un auteur veut créer une application hypermédia sur la base de cette considération.</p>

Structure :

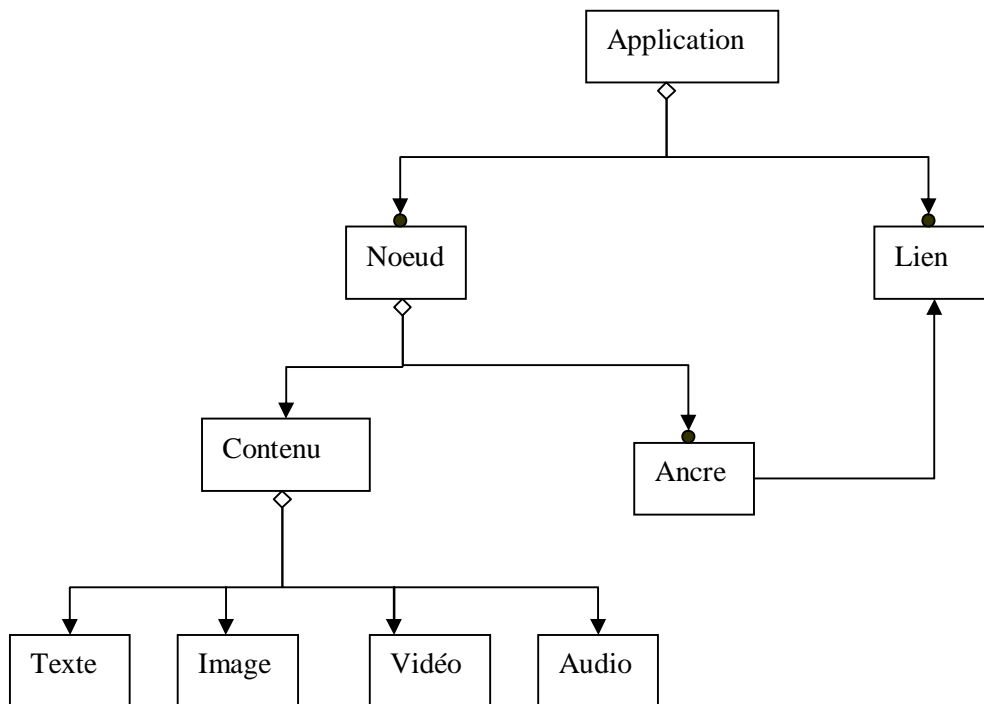


Figure 15 : Structure du patron de conception "Création d'applications hypermédia".

**Constituants :**

- Application :** la classe Application permet de définir les opérations de création des Nœuds et des Liens. L'opération de création des nœuds renvoie des objets de type Nœud, et l'opération de création des liens renvoie des objets de type Lien.
- Nœud :** la classe Nœud définit les opérations nécessaires à la gestion d'un nœud. Parmi ces opérations, nous pouvons trouver l'ouverture d'un nœud, la fermeture d'un nœud, etc. La classe Nœud contient aussi la représentation d'un nœud qui comporte plusieurs informations, par exemple : le nom de nœud pour l'identifier, ses données d'interactions, les types de médias sur lesquels l'utilisateur peut interagir.
- Lien :** la classe Lien définit les informations nécessaires permettant d'identifier un lien. Parmi ces informations, nous pouvons trouver : la source, la destination et la description sémantique. Un lien a une source (ou origine) et une destination (ou cible). La source d'un lien est généralement un élément (mot, phrase, image, etc) dans un nœud. La

être un élément du même nœud, comme elle peut être également un autre nœud. Un lien est bidirectionnel si sa destination est aussi la source réciproque du même lien. Sinon, un lien est unidirectionnel, et rien ne distingue un élément destination d'un quelconque autre élément.

- **Contenu** : la classe Contenu regroupe les différents types de médias que le nœud peut les contenir, c'est-à-dire les objets d'information qui peut se trouver dans le nœud. Ces objets d'information peuvent être : des textes, des images, des vidéos et des sons.
- **Ancre** : la classe Ancre contient toutes les ancres que le nœud peut les contenir. Une ancre peut être un mot, une phrase, une image, un bouton, etc. Un événement sur l'ancre permet d'activer le lien correspondant, par exemple : le passage d'une page Web à une autre dans le WWW se fait à partir d'un simple cliquer sur une ancre existe dans la page source.
- **Les classes Texte, Image, Audio et Vidéo** : ces classes représentent les différents types des contenus supportés par un nœud, par exemple : la classe Image contient l'identification de cette image, la source de cette image, les opérations que l'utilisateur peut effectuer sur cette image, comme : l'enregistrement, l'agrandissement, etc.

### Collaborations :

- Une Application est composée de plusieurs Nœuds et de plusieurs Liens. Un Nœud est lui-même composé d'un Contenu et des Ancres. Le contenu d'un nœud est organisé par l'intermédiaire de la classe Contenu.
- Pour que ce contenu sera visualisé et manipulé facilement par l'utilisateur, il doit avoir une taille moyenne, c'est-à-dire sa taille ne doit pas être trop grande ou trop petite.
- Il est préférable qu'une information se trouve entièrement dans un nœud, c'est-à-dire une information ne doit pas être répartie entre plusieurs nœuds. Cela aide l'utilisateur dans sa tâche de recherche. Au lieu qu'il consulte plusieurs nœuds pour obtenir leur but, il consulte seulement le nœud qui contient ce but.
- La sélection d'une ancre permet d'activer le lien correspondant. Les ancres d'un lien dans les nœuds sources sont des clients qui activent les liens, mais ils n'ont pas la responsabilité de la sélection des nœuds destinations. Ils sont seulement un moyen d'accéder et d'activer le lien.



**PDF Complete**  
 Your complimentary use period has ended.  
 Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

**Conséquences :**

L'utilisation de patron Création d'applications hypermédia offre plusieurs avantages :

- L'ajout d'un nouveau nœud ou d'un nouveau lien est une tâche très simple, parce qu'elle se fait par la création d'une instance de la classe Nœud ou une instance de la classe Lien.
- La modification de contenu d'un nœud est aussi une tâche simple, parce que les types des contenus sont séparés.

Pour montrer la validité de ce patron, nous essayons de l'implémenter. Les aspects de leur implémentation sont présentés dans le chapitre suivant de ce mémoire.

**3.2.2 Adaptation des patrons "Stratégie de navigation" et "Décorateur" :**

Dans la phase d'identification des patrons hypermédia, nous avons identifié que nous avons besoin d'un patron pour la stratégie de navigation et un patron pour l'historique de navigation. Ces deux patrons existent dans les catalogues de patrons proposés dans la littérature, mais les solutions proposées par ces patrons doivent être adaptées avant leurs utilisations. Dans ce qui suit, nous présentons ces deux patrons avec ses solutions adaptées.

**a. Adaptation du patron "Stratégie de navigation" :**

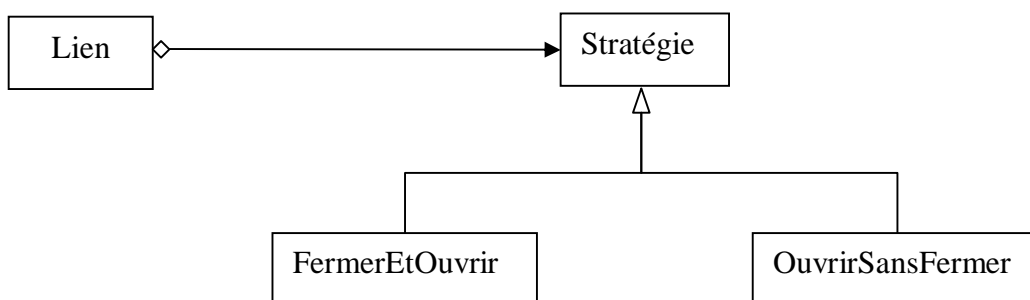
Une fois l'application hypermédia a été construite, il faut avoir des algorithmes permettant de définir les stratégies de navigation entre les composants de cette application. Dans [Gamma, 1995], il existe un patron de conception permet de définir une famille d'algorithmes. Nous avons adapté ce patron pour définir des algorithmes de navigation. Le patron adapté est défini comme suit :

<b>Nom :</b> Stratégie de navigation.
<b>Intention :</b>  Définir deux familles d'algorithmes permettant de donner à l'utilisateur le choix de la manière d'obtention du nœud suivant.
<b>Motivation :</b>  Dans des cas particuliers, un utilisateur veut garder des nœuds dans un état ouvert pour les comparer avec les nœuds suivants. Ce patron donne la main à l'utilisateur de décider lors de l'activation d'un lien soit de fermer le nœud en cours, soit de le garder dans un état ouvert.

**Indication d'utilisation :**

Le patron Stratégie de navigation est utilisé lorsque un auteur veut construire une application hypermédia qui invoque l'utilisateur dans le choix de la stratégie de navigation.

**Structure :**



**Figure 16 :** Structure du patron de conception "Stratégie de navigation".

**Constituants :**

- Stratégie :** la classe Stratégie déclare une interface commune à tous les algorithmes représentés dans le patron. La classe Lien utilise cette interface pour faire un appel à l'un des algorithmes qui sont définis par les sous classes OuvrirSansFermer et FermerEtOuvrir.
- Lien :** la classe Lien implémente un algorithme qui permet l'activation d'un lien et la gestion d'une référence à un objet Stratégie. Elle peut définir une interface qui permette à la Stratégie d'accéder à ces données.
- OuvrirSansFermer :** cette classe implémente un algorithme permettant de trouver le n° ud destination d'un lien activé par un utilisateur. Une fois ce n° ud a été trouvé, elle permet de l'ouvrir mais sans fermer le n° ud source de ce lien activé.
- FermerEtOuvrir :** cette classe implémente un algorithme permettant de trouver le n° ud destination d'un lien activé par un utilisateur. Une fois ce n° ud a été trouvé, elle permet de l'ouvrir et de fermer le n° ud source de ce lien activé.



**PDF Complete**  
Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

**Collaborations :**

- La Stratégie et le Lien interagissent dans l'implémentation de l'algorithme choisi. Un Lien peut fournir à la Stratégie toutes les données que requiert cet algorithme, lorsqu'il est appelé.
- Quand un lien est activé par un utilisateur, la classe Lien donne la main à la classe Stratégie d'invoquer l'utilisateur dans le choix de la façon d'obtention d'un nœud, par exemple : par une boîte de dialogue qui affiche les deux types de choix.

**Conséquences :**

Le patron Stratégie de navigation adapté pour l'hypermédia offre les bénéfices du patron de conception Stratégie proposé par E. Gamma [Gamma, 1995]. Ce dernier permet d'implémenter des algorithmes différents.

**b. Adaptation du patron "Décorateur" :**

La présentation de contenu dans les applications hypermédia est une tâche importante que l'auteur de développement des applications hypermédia doit la mettre en considération lors de la conception de ces applications. Le même contenu peut être présenté avec plusieurs vues différentes selon les types d'utilisateurs, par exemple : dans l'application "Département académique", présentée dans la section 2 de ce chapitre, plusieurs vues navigationnelles sont identifiées selon les types des utilisateurs de cette application. Ces vues ont été dérivées du même modèle, c'est-à-dire que ces vues ont le même contenu, mais ont des présentations différentes.

Dans le catalogue de patrons de conception proposé par E. Gamma [Gamma, 1995], il existe un patron qui offre la solution à cette situation rencontrée. L'intention de ce patron est l'attachement dynamique des responsabilités supplémentaires à un objet. Dans le domaine des applications hypermédia, le décorateur peut être adapté pour ajouter des vues différentes à un contenu existant. Dans ce qui suit, nous présentons la description de ce patron.

**Nom :** Décorateur.

**Intention :**

L'intention d'un décorateur est de fournir une alternative souple à la dérivation pour étendre les fonctionnalités. Concernant les applications hypermédia, l'idée d'un décorateur peut être adoptée pour ajouter des vues différentes à un contenu existant.

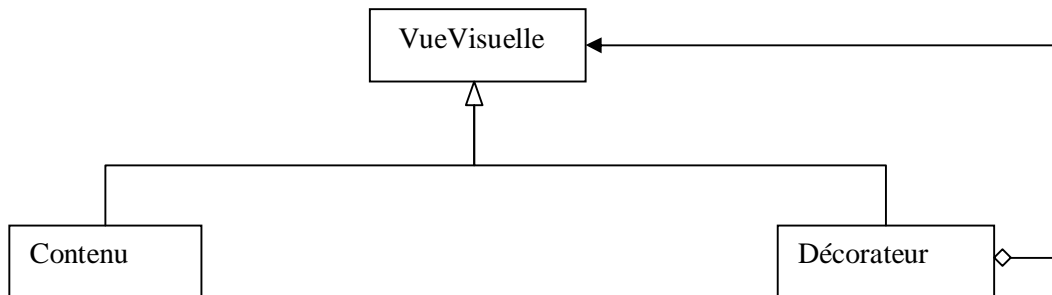
**Motivation :**

Supposons, par exemple, que nous disposions d'un objet Nœud qui contient seulement un objet VueTexte, ce dernier affiche un texte dans une fenêtre. Par défaut, l'objet VueTexte n'a pas d'ascenseurs de défilement ; ceux-ci peuvent ne pas être toujours nécessaires, quant ils le sont, nous pourrions les ajouter par l'intermédiaire d'un DécorateurDefile. Supposons que nous souhaitons également ajouter une épaisse Bordure noire autour de VueTexte. Nous pourrions également utiliser une DécorateurBord. Il s'agit, dans ce cas, de composer simplement des décorateurs avec l'objet VueTexte afin de produire le résultat escompté. Les classes DécorateurDefile et DécorateurBord sont deux sous-classes de la classe Décorateur. C'est une classe abstraite pour les composants visuels qui décore d'autres composants visuels.

**Indication d'utilisation :**

Le patron Décorateur est utilisé dans le développement d'une application hypermédia pour ajouter des vues différentes à un contenu existant.

**Structure :**



**Figure 17 :** Structure du patron de conception "Décorateur".



Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

**Constituants :**

- **VueVisuelle :** la classe VueVisuelle permet de définir l'interface des objets à décorer pour les représenter à un utilisateur.
- **Contenu :** cette classe permet de définir le contenu d'un nœud auquel des décors supplémentaires peuvent être adjoints.
- **Décorateur :** la classe Décorateur gère une référence à un objet VueVisuelle et définit une interface conforme à celle du VueVisuel.

**Collaborations :**

La classe Décorateur transmet des requêtes à ses objets VueVisuelle. Elle peut éventuellement effectuer des opérations supplémentaires avant et après la transmission de la requête.

**Conséquences :**

L'utilisation du patron Décorateur dans le développement d'une application hypermédia permet d'enrichir cette application par des nouvelles vues obtenues à partir des contenus existants.

**3.2.3 Réutilisation du patron "Observateur de navigation" :**

L'un des besoins que l'application hypermédia doit offrir à un utilisateur est l'historique de navigation. Cet historique doit enregistrer l'état de navigation et doit être automatiquement mettre à jour pendant la progression de navigation. La solution à cette situation peut être obtenue par l'utilisation d'un patron proposé dans la littérature par G. Rossi [Rossi, 1997]. La description de ce patron est représentée comme suit :

**Nom :** Observateur de navigation.

**Intention :**

L'Observateur de navigation permet de simplifier la construction de l'historique de navigation en séparant les composants de l'hypermédia (Nœud et Lien) des objets qui implémentent cet historique et leur apparence.

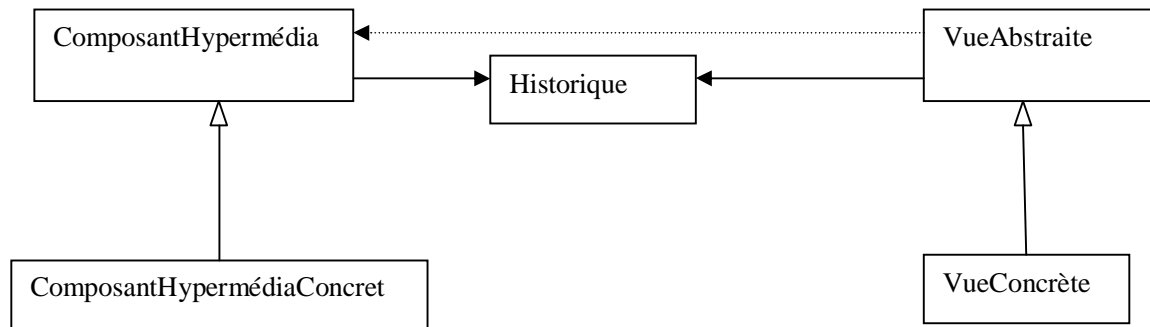
**Motivation :**

Un des besoins que l'application hypermédia doit offrir à un utilisateur est l'historique de navigation. Cet historique doit enregistrer l'état de navigation et doit être automatiquement mettre à jour pendant la progression de navigation. L'historique de navigation permet à un utilisateur de savoir quels nœuds visités par lui pendant son parcours de navigation. Il peut être présenté à un utilisateur par plusieurs vues, par exemple : il peut être présenté par une liste, un graphe ou une autre structure de présentation. La façon la plus commode d'implémenter cet historique est d'utiliser le patron de conception Observateur de navigation.

**Indication d'utilisation :**

L'Observateur de navigation est utile pour avoir un historique percevable des nœuds et des liens déjà visités par un utilisateur pendant son parcours de navigation. En outre, le patron Observateur de navigation rend cet historique indépendant d'eux, il autorise des changements de sa présentation selon les besoins et les préférences d'un utilisateur. Ce patron découple les nœuds de l'historique de navigation et l'historique de son interface (vues). Cela autorise la variation du type de vue ou de l'implémentation de l'historique sans affecter les nœuds.

**Structure :**



**Figure 18 :** Structure du patron de conception "Observateur de navigation".



Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

**Constituants :**

- **ComposantHypermédia :** un composant hypermédia peut être un nœud ou un lien.
- **Historique :** la classe Historique permet d'enregistrer l'historique de navigation.
- **VueAbstraite :** la classe VueAbstraite permet d'implémenter une interface commune pour les différents types des vues.
- **VueConcrète :** la classe VueConcrète permet d'implémenter l'opération de l'affichage de l'historique selon les préférences de l'utilisateur.

**Collaboration :**

Chaque fois un composant hypermédia est accédé, un message est envoyé à l'historique pour l'enregistrer.

**Conséquences :**

- Ce patron découple la navigation de leur historique et l'historique de leur présentation.
- Ce découplage permet de séparer les spécificités des applications et les fonctionnalités de navigation.

**4. Conclusion :**

Dans ce chapitre, nous avons présenté la structure de la démarche proposée pour construire un catalogue de patrons de conception spécifique aux applications hypermédia. Cette démarche permet d'organiser les diverses connaissances de domaine, identifiées lors de l'analyse d'un exemple d'une application hypermédia existante, dans des patrons dédiés au développement de ce type d'applications. Quatre patrons de conception sont définis :

- Le patron "Création d'applications hypermédia" pour assurer la conception, la structuration des différents composants d'une application hypermédia.
- Le patron "Stratégie de navigation" pour définir des algorithmes permettant d'identifier les stratégies de navigation dans une application hypermédia.



**PDF Complete**

*Your complimentary use period has ended. Thank you for using PDF Complete.*

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

pour ajouter des vues différentes à un contenu existant dans une application hypermédia.

- Le patron "Observateur de navigation" pour construire un historique de navigation pour une application hypermédia.

Les patrons de ce catalogue sont identifiés selon les problèmes les plus récurrents dans le domaine traité. Le catalogue proposé comporte les patrons de base nécessaires pour la conception d'une application hypermédia. Pour vérifier la réutilisabilité des patrons proposés, il faut les appliquer plusieurs fois dans des projets de développement des applications hypermédia, d'où l'intérêt est d'exprimer l'approche proposée à base de patrons sur un exemple concret. C'est l'objectif du prochain chapitre d'illustrer cette expérimentation.

## Chapitre 4 :

# Expérimentation du catalogue de patrons.

### 1. Introduction :

L'objectif de ce chapitre est de présenter l'expérimentation des concepts développés dans notre mémoire. Cette expérimentation porte sur trois aspects principaux :

- Le premier aspect consiste à valider la démarche proposée de spécification des applications hypermédia par réutilisation de patrons de notre catalogue. Nous montrons à un auteur de développement des applications hypermédia les étapes principales d'utilisation de ces patrons dans le processus de développement de ce type d'applications.
- Le deuxième aspect consiste à munir les patrons proposés d'une description XML afin de les formaliser pour les mieux gérer et organiser.
- Le troisième aspect consiste à montrer comment les patrons de notre catalogue peuvent être implémentés. Pour cela, nous prenons un exemple d'implémentation d'un patron de catalogue avec le langage Java.

Le présent chapitre est divisé en trois parties, selon trois aspects énoncés ci-dessus: la première partie illustre les étapes principaux d'utilisation des patrons de catalogue. Pour cela, nous les appliquons manuellement sur un exemple de modélisation relative aux applications hypermédia. La deuxième partie montre comment les patrons de catalogue peuvent être décrits par une description XML. La troisième partie présente un exemple d'implémentation de patron "Création d'applications hypermédia" que nous avons créés dans le catalogue. Cette partie a pour objectif d'illustrer la faisabilité logicielle de la démarche proposée.

### 2. Application du catalogue de patrons :

L'objectif de cette section est de montrer comment un auteur de développement des applications hypermédia peut utiliser les patrons constituant le catalogue proposé dans le processus de développement de ce type d'applications. Cette section illustre ainsi la deuxième phase dans l'ingénierie de ce type d'applications, c'est-à-dire le processus d'ingénierie par réutilisation.

Un des avantages de la réutilisation des patrons est leur parfaite indépendance vis-à-vis des technologies et des plateformes d'implémentation. Dans ce cadre là, un patron ne peut

à un outil particulier. L'inaccessibilité temporaire ou définitive de cet outil remettrait en cause le patron.

Afin d'illustrer l'utilisation des patrons proposés, nous allons maintenant présenter un exemple d'ordonnement des différentes étapes pour la spécification d'une simple application hypermédia.

## 2.1 Etapes d'utilisation de catalogue de patrons :

Les activités principales d'utilisation de notre catalogue de patrons pour le développement d'une application hypermédia sont:

1. La création des nœuds constituant cette application.
2. La création des liens qui relient ces différents nœuds.
3. La définition des stratégies de navigation de cette application.
4. L'association de plusieurs vues différentes à un contenu d'un nœud selon les préférences des utilisateurs.
5. La construction d'un historique de navigation des nœuds et des liens déjà visités par un utilisateur pendant son parcours de navigation.

Selon la nature de l'activité, nous faisons un appel au patron adéquat pour modéliser son déroulement :

- Pour modéliser les deux premières activités de création des nœuds et de création des liens qui relient ces différents nœuds, nous utilisons le patron "Création d'applications hypermédia" qui a pour but d'assurer la conception, la structuration des différents composants d'une application hypermédia.
- Pour modéliser l'activité de définition des stratégies de navigation, nous faisons un appel au patron "Stratégie de navigation". Ce patron permet de définir des algorithmes de navigation permettant de donner à l'utilisateur le choix de la manière d'obtention du nœud suivant.
- Pour modéliser l'activité d'association de plusieurs vues différentes à un contenu d'un nœud dans une application hypermédia, nous utilisons le patron "Décorateur" qui a pour but d'ajouter des vues différentes à un contenu existant.
- Enfin, pour modéliser l'activité de construction d'un historique de navigation, nous utilisons le patron "Observateur de navigation" qui permet de construire un historique de navigation. Le tableau suivant résume l'enchaînement de ces activités et le patron à imiter pour modéliser chacune d'elles.

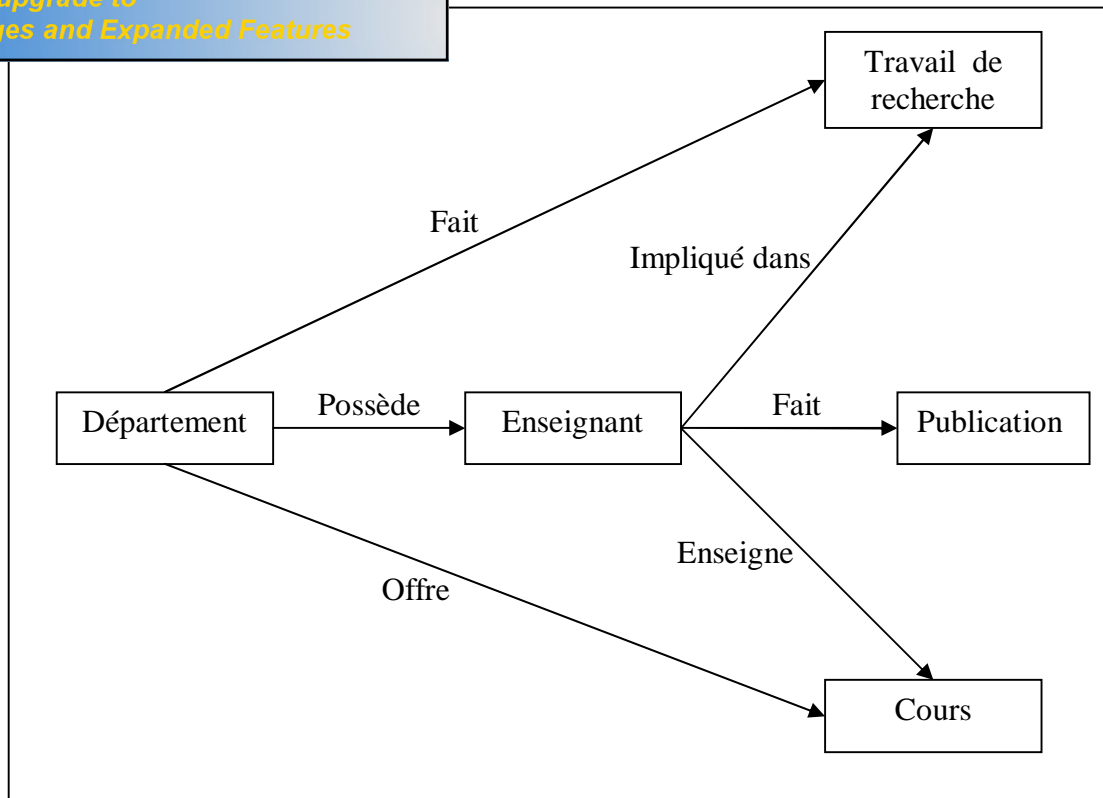
Activité	Patron à imiter	Intention
Création des nœuds.	Création d'applications hypermédia	Assurer la conception, la structuration des différents composants d'une application hypermédia.
Création des liens.		
Définition des stratégies de navigation.	Stratégie de navigation	Définir des algorithmes de navigation permettant de donner à l'utilisateur le choix de la manière d'obtention du nœud suivant.
Association de plusieurs vues différentes à un contenu d'un nœud.	Décorateur	Ajouter des vues différentes à un contenu existant.
Construction d'un historique de navigation.	Observateur de navigation	Construire un historique de navigation.

**Tableau 2 :** Activités pour la mise en place d'une application hypermédia.

## 2.2 Description de l'exemple utilisé :

Pour illustrer les principes de modélisation, nous allons prendre l'exemple de l'application "Département académique" que nous avons utilisé dans le chapitre 3 pour construire le "référentiel hypermédia". Nous avons utilisé cette application comme un exemple pour présenter le déroulement des activités que nous avons précisées précédemment, à cause de leur simplicité à comprendre.

Nous rappelons que l'application hypermédia "Département académique" est une application pour la gestion d'un département académique. Ce département se compose d'un département général d'information, des enseignants, des cours, des travaux de recherche et des publications. Le schéma conceptuel de cette application est représenté comme suit :



**Figure 19 :** Schéma conceptuel de l'application hypermédia "Département académique".

La figure 19 peut être traduit comme suit :

- Un département possède des enseignants, il offre des cours et il effectue des travaux de recherche.
- Un enseignant enseigne des cours et fait des publications. Il peut être impliqué dans des travaux de recherche.

### 2.3 Imitation des patrons :

Cette sous-section illustre l'imitation de divers patrons impliqués dans la spécification de l'exemple de l'application utilisée. Nous décrivons dans ce qui suit chacune des solutions offertes par ces patrons et quelques imitations possibles associées à ces solutions.

#### 2.3.1 Imitation du patron "Création d'applications hypermédia" :

Ce patron propose la solution suivante :

- Une application hypermédia est composée de deux types de composants qui sont: les nœuds et les liens.
- Un nœud est composé d'un contenu et des ancres.



**PDF Complete**  
Your complimentary use period has ended.  
Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

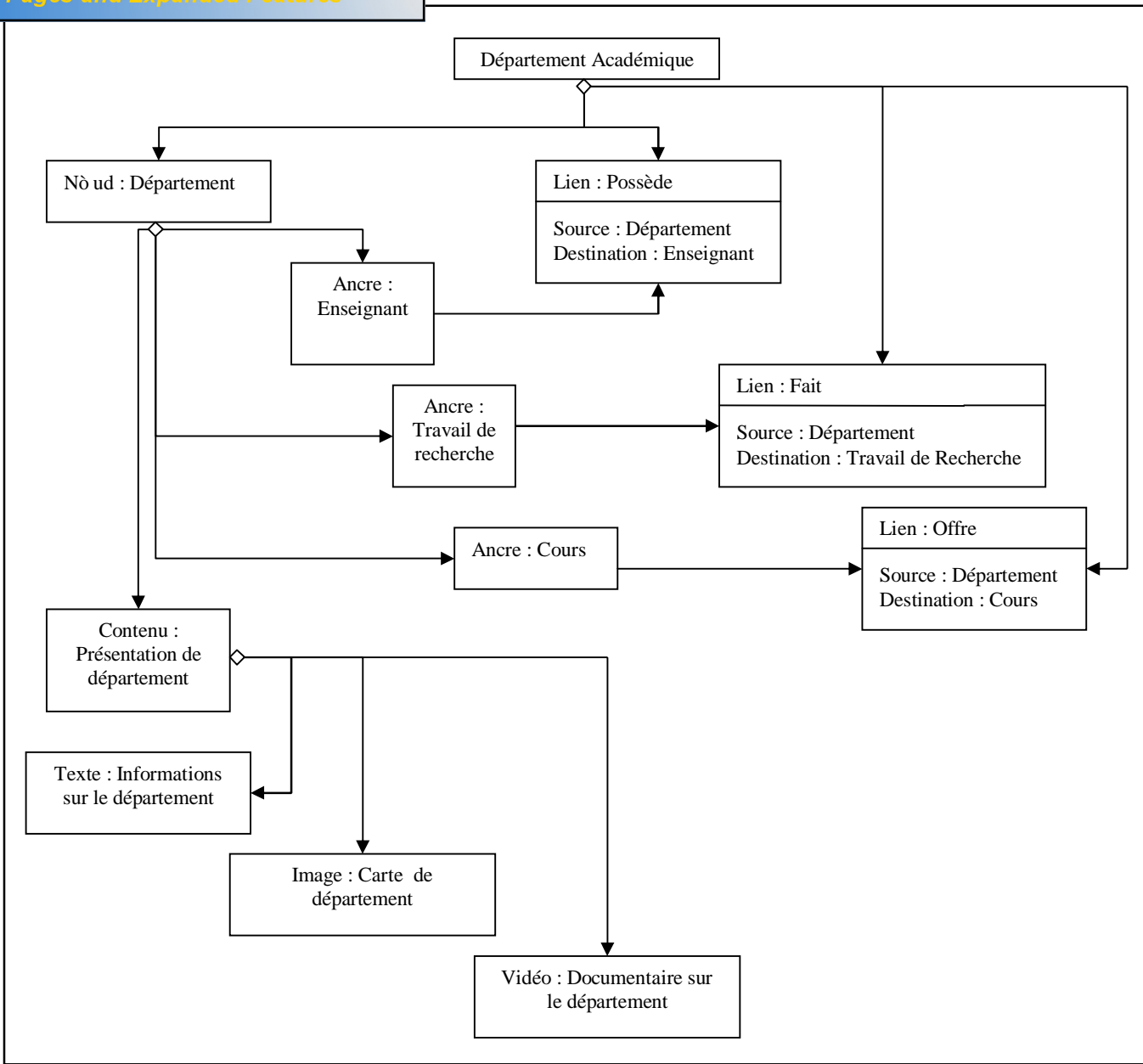
les différents types de médias que le nœud peut les contenir, c'est-à-dire les objets d'informations qui peuvent se trouver dans le nœud. Ces objets d'informations peuvent être : des textes, des images, des vidéos et des sons (audio).

- La classe Ancre contient toutes les ancres que le nœud peut les contenir. Une ancre peut être un mot, une phrase, une image, un bouton, etc.
- Toutes les informations nécessaires pour l'identification d'un lien sont définies dans la classe Lien. Parmi ces informations, nous pouvons trouver : la source, la destination et la description sémantique de chaque lien.
- Un événement sur une ancre permet d'activer le lien correspondant.

Nous essayons d'imiter ce patron pour les deux nœuds "Département" et "Enseignant" de l'application "Département académique".

#### **a. Pour le nœud "Département":**

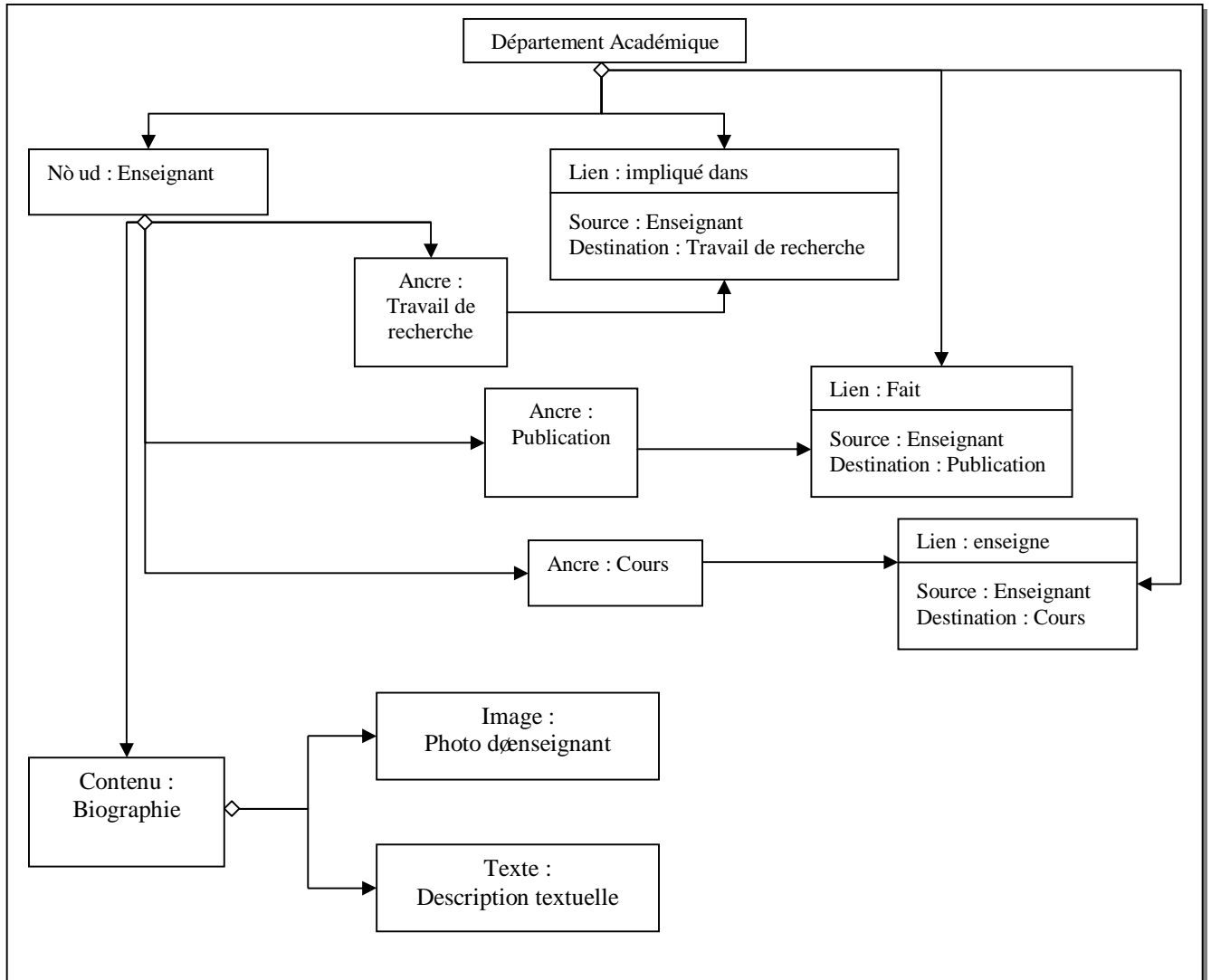
Le patron "Création d'applications hypermédia" permet de créer les nœuds et les liens dans une application hypermédia. Nous présentons dans ce qui suit l'imitation de ce patron pour la création du nœud "Département" et des liens associés à ce nœud.



**Figure 20 :** Imitation du patron "Création d'applications hypermédia" pour le nø ud "Département" et leurs liens associés.

Cette figure montre une imitation de patron "Création d'applications hypermédia" pour la création du nø ud "Département" et leurs liens associés : "Possède", "Fait" et "Offre". Le nø ud "Département" contient une "Présentation de département" et trois ancres "Enseignant", "Travail de recherche" et "Cours". Chacune de ces ancres permettant d'activer le lien correspondant, par exemple : l'ancre "Enseignant" permet d'activer le lien "Possède". L'activation de ce dernier permet à un utilisateur de passer au noeud "Enseignant". Le contenu du noeud "Département" se compose d'un texte qui contient des informations sur ce département, d'une image qui présente une carte graphique de ce département, et d'une vidéo qui présente un documentaire sur ce-ci.

La limitation du patron "Création d'applications hypermédia" pour la création du nœud "Enseignant" et des liens associés à ce nœud est présentée par la figure suivante.



**Figure 21 :** Imitation du patron "Création d'applications hypermédia" pour le nœud "Enseignant" et leurs liens associés.

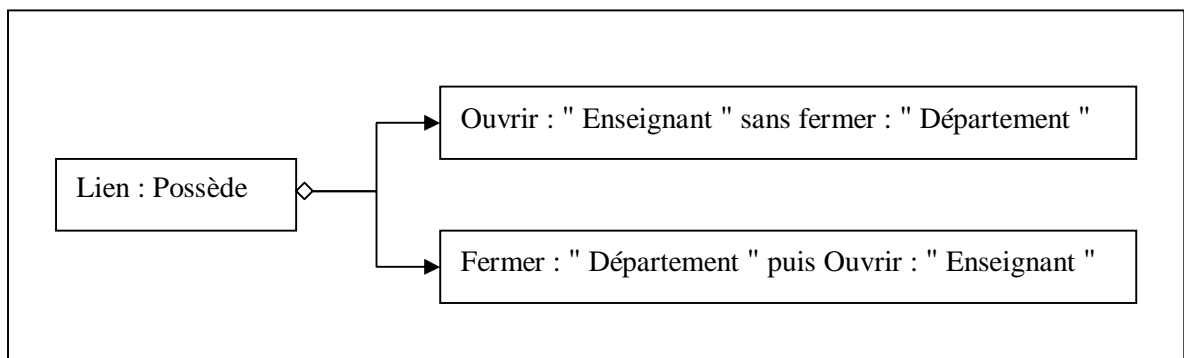
La figure 21 montre une imitation de patron "Création d'applications hypermédia" pour la création du nœud "Enseignant" et leurs liens associés : "Impliqué dans", "Fait" et "Enseigne". Le nœud "Enseignant" contient une "Biographie" de cet enseignant et trois ancres "Publication", "Travail de recherche" et "Cours". Chacune de ces ancres permettant d'activer le lien correspondant, par exemple: l'ancre "Cours" permet d'activer le lien "Enseigne". L'activation de ce dernier permet à un utilisateur de passer au nœud "Cours". La biographie d'un enseignant est présentée à un utilisateur comme une description textuelle dotée d'une photo de l'enseignant.

**"Stratégie de navigation" :**

Ce patron propose la solution suivante:

- La navigation dans l'application hypermédia se fait à travers d'activation d'une série des liens.
- Chaque lien est associé avec un algorithme permettant de définir une stratégie de navigation, c'est-à-dire comment les utilisateurs obtiennent les nœuds désirés.
- Ce patron propose deux algorithmes de navigation qui sont : "OuvrirSansFermer" et "FermerEtOuvrir". Le premier permet de trouver le nœud destination d'un lien activé par un utilisateur. Une fois ce nœud a été trouvé, il permet de l'ouvrir mais sans fermer le nœud source de ce lien activé. Le deuxième algorithme permet de trouver le nœud destination d'un lien activé par un utilisateur. Une fois ce nœud a été trouvé, il permet de l'ouvrir et de fermer le nœud source de ce lien activé.

Nous présentons ici une imitation de ce patron pour le lien "Possède" qui relie les deux nœuds : le nœud "Département" et le nœud "Enseignant".



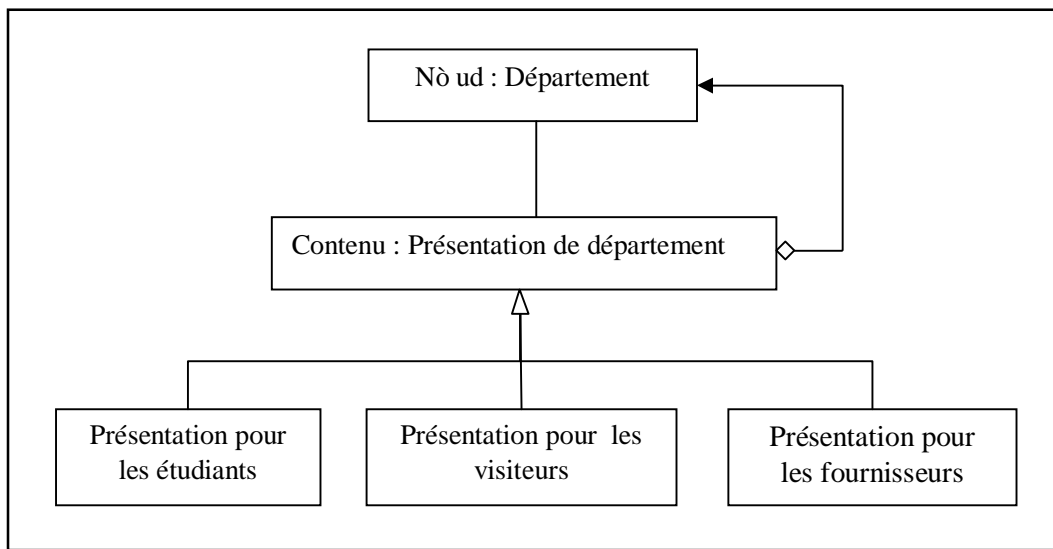
**Figure 22 :** Imitation du patron "Stratégie de navigation" pour le lien "Possède".

Cette figure présente une imitation de patron "Stratégie de navigation" pour le lien "Possède". Cette imitation montre que l'utilisateur a le choix entre deux manières de consultation du nœud "Enseignant". Il peut ouvrir ce nœud sans fermer le nœud "Département", comme il peut fermer ce nœud puis passer au nœud "Enseignant".

**2.3.3 Imitation du patron "Décorateur" :**

Le patron "Décorateur" permet d'associer plusieurs vues différentes à un nœud donné selon les types des utilisateurs. Nous avons vu dans le chapitre 3, que l'application hypermédia "Département académique" a trois types d'utilisateurs différents qui sont : les étudiants, les visiteurs et les fournisseurs. Chaque type d'utilisateur accède cette application d'une manière différentes que les autres utilisateurs. Nous avons donc trois vues navigationnelles possibles qui sont: la vue "Etudiant", la vue "Visiteur" et la vue "Fournisseur". Nous présentons dans la figure suivante une imitation de patron

er au contenu d'un département les trois vues de navigation de l'application "Département académique".



**Figure 23 :** Imitation du patron "Décorateur" pour le contenu du nò ud "Département".

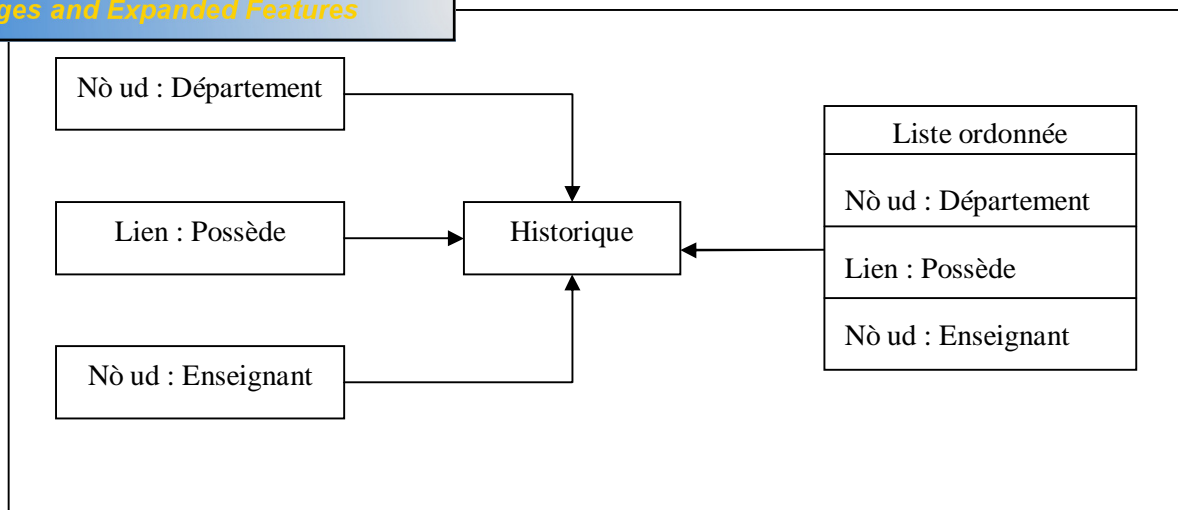
La figure 23 montre que le nò ud "Département" a trois vues de "Présentation de département" possibles : si l'utilisateur est de type étudiant, nous lui montrons une "Présentation pour les étudiants". Si l'utilisateur est de type visiteur nous lui montrons une "Présentation pour les visiteurs" et si l'utilisateur est de type fournisseur nous lui montrons une "Présentation pour les fournisseurs".

### 2.3.4 Imitation du patron "Observateur de navigation" :

Ce patron propose la solution suivante :

- Les nò uds et les liens déjà visités par un utilisateur sont enregistrés dans une classe Historique. Cette dernière permet d'enregistrer l'état de navigation.
- L'historique de navigation est automatiquement mettre à jour pendant la progression de navigation.
- L'historique peut être présenté par plusieurs vues. Il peut être présenté par une liste, un graphe, ou une autre structure de présentation.

Nous présentons dans la figure suivante un exemple d'une imitation de patron "Observateur de navigation".



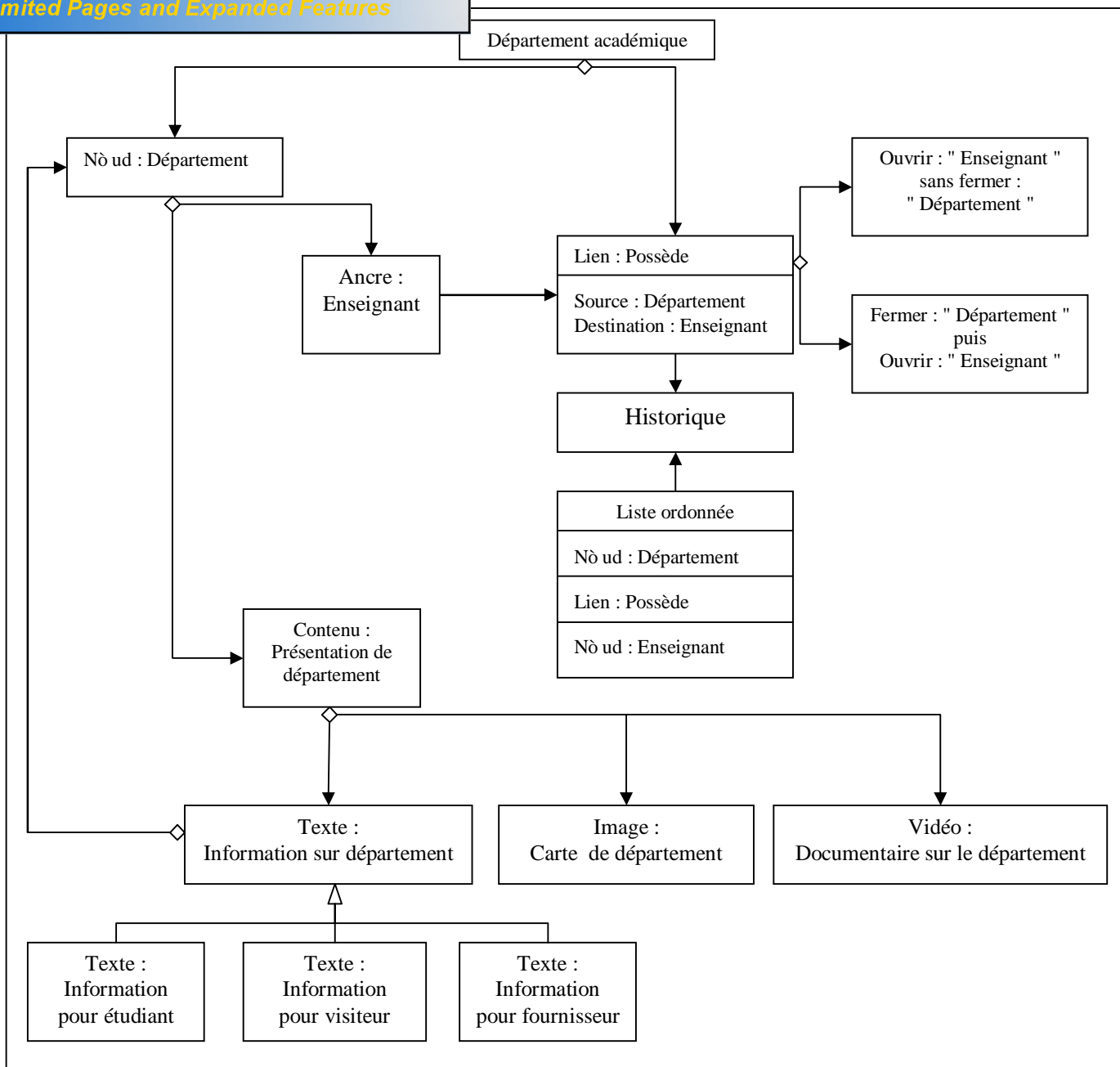
**Figure 24 :** Imitation du patron "Observateur de navigation".

Cet exemple d'imitation de patron "Observateur" est traduit comme suit :

- La consultation de nò ud "Département" permet d'enregistrer ce noeud dans la classe "Historique".
- L'activation de lien "Possède" permet d'enregistrer ce lien dans la classe "Historique".
- La consultation de nò ud "Enseignant" permet d'enregistrer ce noeud dans la classe "Historique".
- Tous ces éléments sont représentés à un utilisateur se forme d'une liste ordonnée.

#### 2.4 Intégration des imitations :

L'intégration des imitations nous donne le modèle final qui représente la spécification de l'enchaînement des différentes étapes pour la mise en place d'une application hypermédia. Nous présentons dans la figure 25 un exemple d'intégration des imitations des patrons de catalogue pour la création du nò ud "Département" et du lien "Possède".



**Figure 25 :** Exemple d'intégration des imitations des patrons de catalogue pour la création du nò ud "Département" et du lien "Possède".

La figure 25 montre l'intégration des quatre imitations suivantes :

- Imitation de patron "Création d'applications hypermédia" pour la création de nò ud "Département" et de lien "Possède".
- Imitation de patron "Stratégie de navigation" pour la définition des stratégies de navigation pour le lien "Possède".



Observateur" pour l'association des vues navigationnelles au contenu de no uir "Département".

- Imitation de patron "Observateur de navigation" pour l'enregistrement de ces deux éléments dans une liste ordonnée.

### 3. Formalisation des patrons à l'aide d'une notation XML :

Les patrons de conception dans le catalogue proposé sont décrits par le formalisme d'E. Gamma [Gamma, 1995]. Ce formalisme utilise le diagramme de classe UML pour représenter la structure d'un patron. Lors de notre étude, nous avons trouvé un travail de recherche qui exploite les 23 patrons de conception proposés par E. Gamma dans la construction d'un outil logiciel [Forge, 2009]. Ces 23 patrons ont été formalisés par une notation XML avant de leurs réutilisations dans cet outil.

Nous avons exploité cette idée pour formaliser les patrons de notre catalogue par une notation XML. Cette formalisation des patrons permet de mieux les gérer et les organiser. La description des patrons par la notation XML garantit la dimension de la réutilisabilité des patrons, parce que le langage XML n'est pas spécifique à une plateforme spécifique, un fichier XML peut être manipulé par n'importe quel type d'outil.

La description d'un patron par une notation XML, à partir de leur diagramme de classe UML, nécessite des règles de passage. Dans le travail de recherche cité au-dessus, ces règles de passage ne sont pas précisées, les patrons sont décrits directement en XML.

Dans notre travail, nous essayons de déduire les règles de passage en XML à partir d'une comparaison entre les diagrammes de classe des patrons d'E. Gamma et leurs descriptions XML dans l'outil logiciel cité précédemment. Avant de présenter ces règles de passage, nous essayons de présenter les notions générales d'XML.

#### 3.1 XML :

Le dernier standard important dans le domaine de la documentation structurée est l'XML, qui allie la simplicité d'HTML et la force de SGML (*Standard General Markup Language*)<sup>8</sup>. XML est l'acronyme de eXtensible Markup Language, cela signifie que XML n'est pas un langage sémantiquement figé comme peut l'être HTML, mais au contraire un langage ouvert. XML est une notation permettant de décrire un langage, sous la forme d'une grammaire d'arbre. XML est plus général et uniforme qu'HTML et plus simple que SGML dont il est un descendant direct.

Ce formalisme a été défini par le W3C (*World Wide Web Consortium*) [W3C, 2009] et regroupe sous l'acronyme XML une boîte à outils extensible dont le but est de simplifier la gestion, c'est-à-dire l'organisation, l'échange et la transformation des données. Il est important de noter qu'XML définit un ensemble de règles pour organiser les jeux de données, mais aussi un ensemble de technologies permettant de travailler avec ces données

<sup>8</sup>SGML (*Standard General Markup Language*) : est un langage non propriétaire et indépendant de la plateforme et du système sur lequel il est utilisé. Les documents créés grâce à ce langage sont utilisables sur toutes les plates-formes sans restriction de compatibilité.



Language Transformation), etc.

StyleSheet Language), XSLT (eXtensible StyleSheet

La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il permet de séparer le contenu de la présentation, ce qui permet, par exemple, d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que nous nécessitons de les représentés. En pratique, un fichier XML est un simple fichier texte, contenant des balises qui sont définies par l'utilisateur pour répondre à ces besoins [Badica, 2008].

L'XML à plusieurs avantages, parmi les quels nous citons :

- **La lisibilité** : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML.
- **La structure arborescente** : permettant de modéliser la majorité des problèmes informatiques.
- **L'universalité et la portabilité** : les différents jeux de caractères sont pris en compte.
- **La déployabilité** : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme HTTP (*HyperText Transfer Protocol*)<sup>9</sup>.
- **L'intégrabilité** : un document XML est utilisable par toute application pourvue d'un parseur (c'est-à-dire un logiciel permettant d'analyser un code XML).
- **L'extensibilité** : un document XML doit pouvoir être utilisable dans tous les domaines d'applications.

Le document XML a deux types qui sont : le document valide et le document bien formé. Le document valide est un ensemble des grammaires permettant de vérifier est ce que le document XML respect la syntaxe de données, par exemple : le DTD (*Document Type Définition*) et les schémas W3C. Le document bien formé est un document XML qui ne comporte pas des grammaires de vérification de conformité mais réponds aux règles de base du XML.

---

<sup>9</sup> http (*HyperText Transfer Protocol*) : est un protocole plus utilisé sur Internet depuis 1990. Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML).

Dans cette sous section, nous présentons l'ensemble des règles de passage permettant de décrire les patrons par une notation XML à partir de leurs diagrammes de classe UML.

- **Règle 1 :** le nom du patron est représenté par la balise < NomPatron > comme suit :

< NomPatron > Nom de patron </NomPatron >

Par exemple : pour le nom de patron "Création d'applications hypermédia" nous avons :

< NomPatron > Création d'applications hypermédia </NomPatron >.

- **Règle 2 :** une classe est représentée par la balise < Classe > comme suit :

< Classe > Nom de classe </Classe >

Par exemple : la classe Nøud dans le patron "Création d'applications hypermédia" est représentée comme suit :

< Classe > Noeud </Classe >

- **Règle 3 :** la relation d'agrégation est représentée par la balise <Agrégation>. Cette balise comporte trois balises pour représenter les éléments de cette relation. Ces trois éléments sont : la classe agrégative, la classe agrégée et l'ordre de la relation.

- La classe agrégative est représentée par la balise < ClasseAgrégative > comme suit :

< ClasseAgrégative >

Nom de la classe agrégative

</ ClasseAgrégative >

- La classe agrégée est représentée par la balise < ClasseAgrégée > comme suit :

< ClasseAgrégée > Nom de la classe agrégée </ ClasseAgrégée >

- La relation d'agrégation peut être plurielle ou singulière, cet ordre est représenté par la balise <OrderRelation > comme suit :

< OrderRelation > Pluriel ou singulier </OrderRelation >

relation d'agrégation devient comme suit :

< Agrégation >

< ClasseAgrégative > Nom de la classe agrégative </ ClasseAgrégative >

< ClasseAgrégée > Nom de la classe agrégée </ ClasseAgrégée >

< OrderRelation > Pluriel ou singulier </OrderRelation >

</Agrégation >

Par exemple : la relation d'agrégation dans le patron "Création d'applications hypermédia" est représentée comme suit :

< Agrégation >

< ClasseAgrégative > Application </ ClasseAgrégative >

< ClasseAgrégée > Noeud </ ClasseAgrégée >

< OrderRelation > Pluriel </OrderRelation >

</Agrégation >

- **Règle 4** : la relation de référence (ou la délégation) est représentée par la balise < Référence >. Cette balise comporte deux balises pour représenter les éléments de cette relation. Ces deux éléments sont : la classe référence et la classe référencée.

- La classe référence est représentée par la balise < ClasseRéférence > comme suit :

< ClasseRéférence >

Nom de la classe référence

</ClasseRéférence >

- La classe référencée est représentée par la balise < ClasseRéférénciée > comme suit :

< ClasseRéférénciée >

Nom de la classe référencée

</ ClasseRéféréncie >



relation de référence devient comme suit :

< Référence >

< ClasseRéférence > Nom de la classe référence </ClasseRéférence >

< ClasseRéférénciée > Nom de la classe référénciée </ ClasseRéférénciée >

</Référence >

Par exemple : La relation de référence dans le patron "Création d'applications hypermédia" est représentée comme suit :

< Référence >

< ClasseRéférence > Ancre </ClasseRéférence >

< ClasseRéférénciée > Lien </ ClasseRéférénciée >

</Référence >

- **Règle 5** : la relation d'héritage est représentée par la balise <Héritage>. Cette balise comporte deux balises pour représenter les éléments de cette relation. Ces deux éléments sont : la classe parente et la classe dérivée.

- La classe parente est représentée par la balise < ClasseParente > comme suit :

< ClasseParente > Nom de la classe parente </ClasseParente >

- La classe dérivée est représentée par la balise < ClasseDérivée > comme suit :

< ClasseDérivée >

Nom de la sous classe de l'héritage

</ ClasseDérivée >



relation d'héritage devient comme suit:

< Héritage >

< ClasseParente >

Nom de la classe parente de l'héritage

</ClasseParente >

< ClasseDérivée >

Nom de la sous classe de l'héritage

</ ClasseDérivée >

</Héritage >

Par exemple : La relation de l'héritage dans le patron "Observateur de navigation" est représentée comme suit :

< Héritage >

< ClasseParente > Composant Hypermédia </ClasseParente >

< ClasseDérivée > Nò ud </ ClasseDérivée >

</Héritage >

- **Règle 6** : la relation d'instanciation (ou de dépendance) est représentée par la balise <Instanciation>. Cette balise comporte deux balises pour représenter les éléments de cette relation. Ces deux éléments sont : la classe instanciateur et la classe instanciée.

- La classe instanciateur est représentée par la balise < ClasseInstanciateur > comme suit :

< ClasseInstanciateur >

Nom de la classe instanciateur

</ClasseInstanciateur >

- La classe instanciée est représentée par la balise <ClasseInstanciée> comme suit :

< ClasseInstanciée > Nom de la classe instanciée </ClasseInstanciée >



relation d'instanciation devient comme suit :

```
< Instanciation >
```

```
< ClasseInstantiateur >
```

```
    Nom de la classe instanciatrice
```

```
</ ClasseInstantiateur >
```

```
< ClasseInstanciée >
```

```
    Nom de la classe instanciée
```

```
</ ClasseInstanciée >
```

```
</ Instanciation >
```

Par exemple : La relation d'instanciation dans le patron "Observateur de navigation" est représentée comme suit :

```
< Instanciation >
```

```
    < ClasseInstantiateur > VueAbstraite </ ClasseInstantiateur >
```

```
    < ClasseInstanciée > ComposantHypermédia </ ClasseInstanciée >
```

```
</ Instanciation >
```

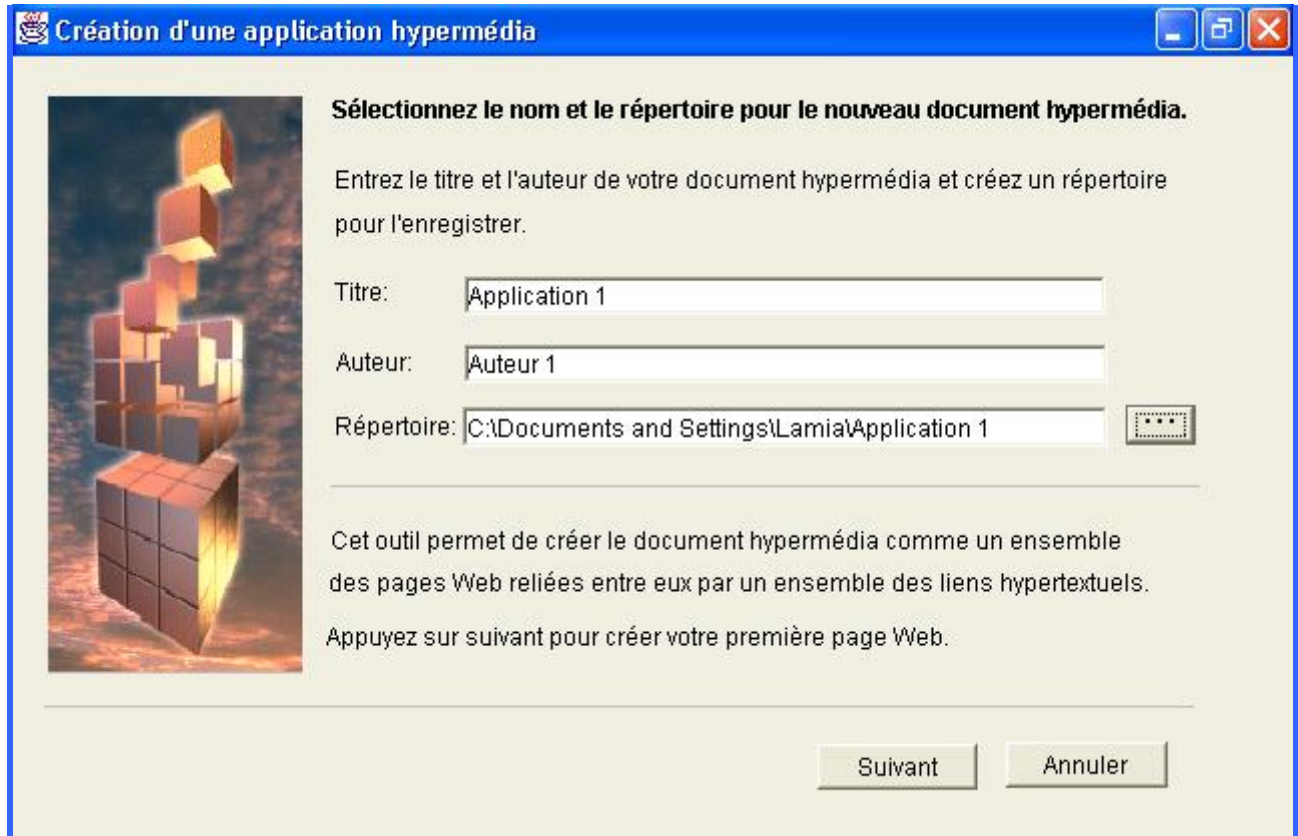
#### 4. Quelques aspects d'implémentation :

Notre catalogue de patrons comporte quatre patrons de conception spécifiques au développement des applications hypermédia. Pour monter la validité de notre catalogue, il faut l'appliquer dans plusieurs projets de développement de ce type d'applications. Dans notre mémoire, nous avons choisi d'implémenter le patron "Création d'applications hypermédia" (créés dans le catalogue) pour montrer sa validité. Le langage que nous avons utilisé pour l'implémentation de ce patron est le langage Java. Avant de présenter les aspects d'implémentation de ce patron, nous présenterons quelques notions du langage Java.

Le langage Java est un langage orienté objet. Contrairement aux langages de type procéduraux comme le C ou le Pascal, le langage Java ne divise pas l'espace de données (attributs) et l'espace de traitements (méthodes). Cet espace commun s'applique à une partie du problème à gérer sous la forme d'une classe. Une classe est une représentation abstraite décrivant comment faire fonctionner des objets. Les objets sont construits à partir de cette classe lors de l'exécution par un processus d'instanciation (par l'opérateur new). Chacune des déclarations dans une classe peut être limitée dans sa portée (portée locale ou privée, portée de groupe ou package, portée globale ou publique).

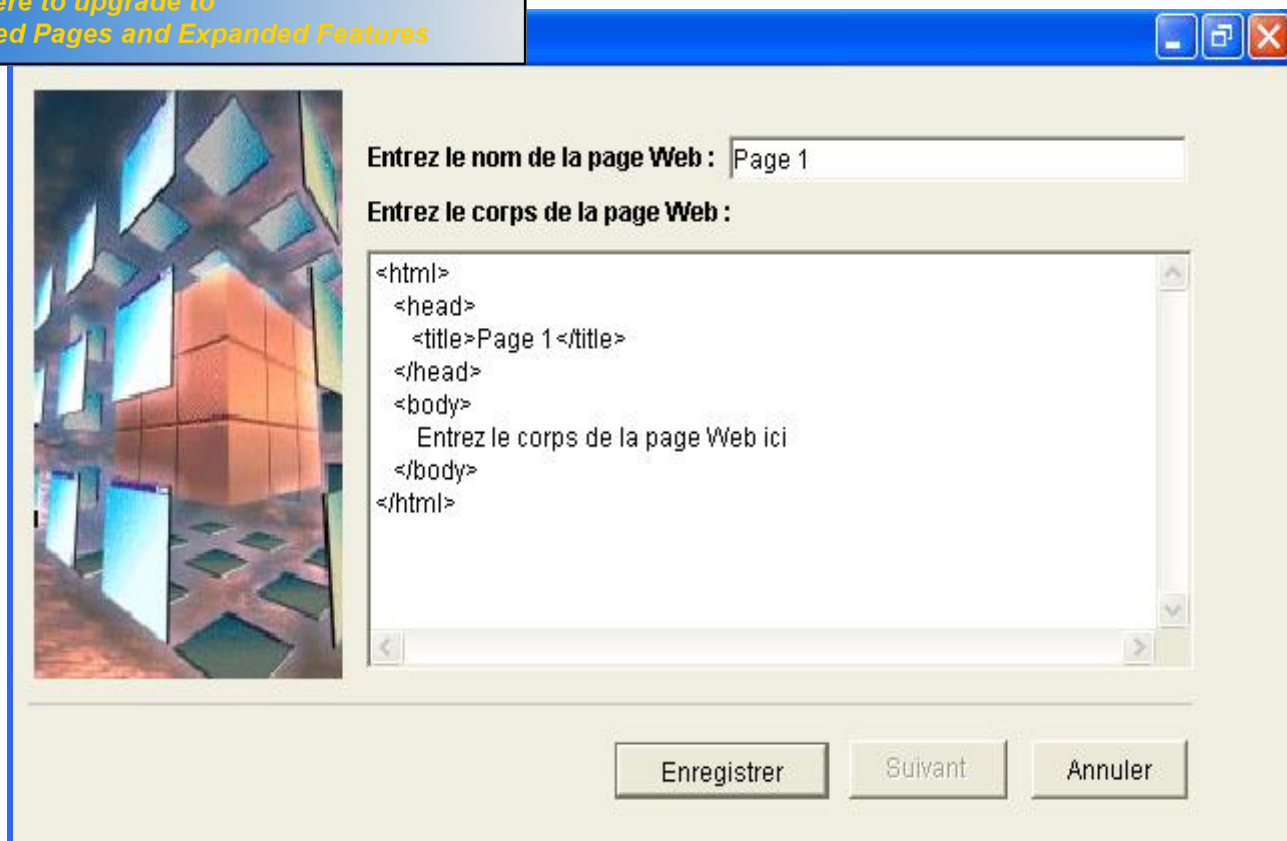
à d'autres classes pour faciliter la réutilisation. L'association la plus commune est l'héritage. L'héritage sert à spécialiser une classe existante (lien de généralisation ou de spécialisation) par la modification ou l'ajout de nouvelles méthodes et de nouvelles données. Cette spécialisation conduit à la construction de nouvelles classes (appelées aussi sous-classes) [Brillant, 2005].

Nous présentons maintenant quelques principes d'implémentation de patron "Création d'applications hypermédia". Ce patron est implémenté comme un outil graphique qui permet de créer les noeuds et les liens dans une application hypermédia.



**Figure 26 :** Interface pour la création d'une application hypermédia.

Cet outil permet à un auteur de développement des applications hypermédia de créer les noeuds comme des pages Web et de créer les liens comme des liens hypertextuels. Le contenu d'un noeud est spécifié donc par le langage HTML et les liens sont des liens hypertextuels de type "href". Les types d'un contenu sont spécifiés par les balises HTML, par exemple pour présenter une image dans une page Web, nous utilisons la balise HTML <IMG>.



**Figure 27 :** Interface pour la création des pages Web.

Cet outil peut être utilisé par n'importe quel utilisateur qui maîtrise le langage HTML. Il le guide dès la création du répertoire pour enregistrer les pages Web jusqu'à la création de la dernière page. Il permet aussi de modifier le contenu d'une page Web sélectionnée par un utilisateur.

### 5. Conclusion :

Ce chapitre a illustré d'une part l'expérimentation de la démarche de spécification des applications hypermédia par réutilisation de patrons que nous avons proposés, et d'autre part la formalisation de ces patrons par une notation XML et la présentation de quelques aspects d'implémentation du patron que nous avons créés dans le catalogue. L'expérimentation de la démarche a été menée en appliquant manuellement le catalogue de patron sur un exemple de modélisation relatif aux applications hypermédia. Cette application ne suffit certes pas à valider l'approche proposée. Dans un contexte plus large, l'objectif sera plutôt de confronter cette approche à des projets réels de développement et de vérifier ainsi les trois points suivant :

- La mise en œuvre de la démarche de spécification par réutilisation de patrons est facilement appréhendable par les concepteurs.



**PDF Complete**

*Your complimentary use period has ended. Thank you for using PDF Complete.*

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

---

répondent à des problèmes récurrents rencontrés dans des projets de développement des applications hypermédia sur le terrain industriel.

- Les modèles de spécification obtenus par réutilisation des patrons proposés sont conformes aux objectifs de développement.

Le catalogue de patrons proposé reste toutefois à enrichir par d'autres patrons spécialisant davantage des problèmes traités.

## Conclusion et perspectives

Ce travail se situe dans le cadre de domaine des applications hypermédia et vise à mettre en place un catalogue de patrons dédié à la conception de ce type d'applications. Cela nous a conduit à étudier les caractéristiques de ces applications hypermédia. Nous avons défini les caractéristiques principales suivantes : la capacité de représenter les informations non structurées, la capacité de relier n'importe quelle pièce d'information par n'importe quelle autre, la capacité de lier un grand nombre d'unités d'information pour les partager entre les groupes collaboratives, la capacité de combiner les connaissances formelles avec les connaissances informelles et la capacité de naviguer entre les différentes unités d'information liées par des liens hypertextuels.

Toutes ces propriétés impliquent l'augmentation avec un taux inattendu de nombre, de taille moyenne et de la complexité de telles applications. Dans ce contexte, plusieurs méthodes de conception ont été proposées dans la littérature pour résoudre ces problèmes. D'autre part, l'investigation dans la littérature récente des techniques de conception et des approches de réutilisation, susceptibles de réaliser les besoins de modélisation, permet de fixer les éléments clés sur lesquels se base le cadre méthodologique proposé. Il s'agit alors de développer une démarche de conception des applications hypermédia basée sur la réutilisation de patrons.

L'introduction des patrons dans une démarche d'ingénierie des systèmes hypermédia nécessite de mettre en œuvre deux processus complémentaires :

- Un processus pour la réutilisation dédié à l'ingénierie des patrons. Il consiste à identifier, spécifier et organiser les patrons à utiliser durant la spécification des applications hypermédia.
- Un processus par la réutilisation dédié à l'ingénierie des applications hypermédia, partant des spécifications des utilisateurs. Ce processus facilite la recherche, la sélection, l'adaptation et l'intégration de patrons pour spécifier une application hypermédia.

Le catalogue de patrons dédié au domaine des applications hypermédia que nous avons proposé est constitué de quatre patrons suivants:

- Le patron "Création d'applications hypermédia" : ce patron permet d'assurer la conception, la structuration des différents composants d'une application hypermédia.
- Le patron "Stratégie de navigation" : ce patron permet de définir des familles d'algorithmes pour identifier des stratégies de navigation.
- Le patron "Décorateur" : ce patron a pour but d'ajouter des vues différentes à un contenu existant.

de navigation" : ce patron permet de construire un historique de navigation.

Ces patrons permettent de réutiliser des fragments de modèles représentant divers aspects dans les applications hypermédia. Par ailleurs, le formalisme adopté pour représenter les patrons favorise la sélection et l'organisation des patrons.

Notre contribution est basée sur les deux apports suivants :

- La définition d'une démarche générale pour la construction des patrons de conception. Si les patrons commencent à devenir un sujet commun du domaine de l'ingénierie des systèmes d'information, leur ingénierie pose actuellement de nombreux problèmes et la mise en œuvre de cette forme d'ingénierie constitue une problématique de recherche.
- La proposition d'un catalogue de patrons dédié à la conception des applications hypermédia qui capitalise les savoirs et les savoir-faire mis en œuvre dans l'ingénierie des applications hypermédia, couvre la complexité des connaissances gérées dans ces applications et propose diverses structurations pour organiser ces connaissances et facilite leur gestion.

Ce travail ouvre la voie à notre sens vers diverses perspectives de recherche. Il serait intéressant, dans un premier temps, de compléter le catalogue de patrons proposé par d'autres patrons raffinant les patrons actuels. Dans un second temps, et en ce qui concerne la formalisation des patrons à l'aide d'une notation XML, il serait intéressant d'utiliser cette description XML comme un point de départ d'implémentation des patrons de catalogue. Dans ce cadre là, il serait intéressant de développer un outil logiciel qui permet de transformer les diagrammes de classes UML des patrons en une description XML, puis exploiter cette description dans l'implémentation de l'ensemble de ces patrons. En ce qui concerne l'expérimentation de la démarche proposée, il serait intéressant d'implémenter tous les patrons de catalogue proposé et les appliquer dans des projets industriels de spécification des applications hypermédia.

## Introduction générale

### 1. Contexte de travail :

L'émergence du WWW (*World Wide Web*) a encouragé l'apparition d'une nouvelle génération des systèmes d'information qui utilisent la notion de navigation. Cette dernière est obtenue à travers de la combinaison d'un espace des informations avec la nature structurée des données multimédia [Schwabe, 1995]. Le WWW introduit, du point de vue de conception des applications, le paradigme hypertexte (ou hypermédia)[Bruillard , 1995].

Les premières applications hypermédia, typiquement distribuées dans des CD-Roms, étaient pensées comme des applications inchangeables qui n'ont pas l'intention d'être maintenues ou d'être modifiées pour les nouveaux besoins. Cette considération peut être comme une conséquence à des propriétés physiques de leurs supports média. Avec l'évolution de technologie, notamment l'évolution de phénomène informatique, les applications ont le besoin d'être modifiables, elles doivent être enrichies avec des nouveaux services, et des nouvelles propriétés de navigation et d'interface sont ajoutées. Les bonnes applications basées Web devraient être, avant tout, des bonnes applications hypermédia. En outre le WWW pose certaines contraintes qui doivent être respectées principalement dans l'interface et l'architecture de l'implémentation.

Les méthodologies traditionnelles d'ingénierie des logiciels, ou les méthodologies de développement des systèmes d'information qui utilisent les bases de données ne contiennent pas des abstractions utiles capables de faciliter la tâche de spécification des applications hypermédia [Schwabe, 1998]. Par exemple, ils ne fournissent pas la notion de lien, et très peu de ces méthodologies montrent comment incorporer l'hypertexte dans l'interface.

Les applications hypermédia se caractérisent par plusieurs propriétés, elle permet de représenter les informations non structurées, elle permet de relier n'importe quelle pièce d'information par n'importe quelle autre, elle permet aussi de lier un grand nombre d'unités d'information pour les partager entre les groupes collaboratives et de combiner les connaissances formelles avec les connaissances informelles. Ces propriétés impliquent l'augmentation avec un taux inattendu de nombre, de taille moyenne et de la complexité de telles applications et rendent leur conception une tâche difficile, complexe et fastidieuse.

La dimension "réutilisation" prend aujourd'hui de plus en plus d'importance, « *elle consiste de prendre les avantages des efforts effectués dans des travaux précédemment réalisés pour réduire l'effort de réalisation d'un travail donné* » [Nanard, 1998]. Les recherches sur la réutilisation ont progressivement évolués ces dernières années, passant de la réutilisation de code à la réutilisation de connaissances et de démarches de raisonnement de différents niveaux. La réutilisation présente un enjeu supplémentaire pour garantir une meilleure qualité des développements. L'approche que nous adoptons, propose des éléments d'une part pour pallier les limites des méthodes traditionnelles concernant leur capacité à représenter les connaissances relatives aux applications hypermédia, d'autre

de nouvelles connaissances acquises par les développeurs d'applications des différents niveaux d'analyse des besoins.

## 2. Problématique et objectifs :

Aujourd'hui, les systèmes multimédia et hypermédia constituent un support incontournable dans le développement des applications dites avancées, telles que le e-commerce, les systèmes d'information basés Web, le e-learning, le e-government et bien d'autres. Ils constituent un moyen efficace pour l'utilisation et la manipulation des données.

Par ailleurs, la technologie des patrons de conception ou "Design Patterns" en anglais (les Design Patterns ont plusieurs noms francisés qui permettent de les désigner. Les plus courants sont "modèles de conception" ou encore "patrons de conception". La signification de ces termes est équivalente [Gamma, 1995]. Dans notre thèse, nous conservons le terme "patron de conception") a montré son efficacité dans le développement de logiciels dédiés à divers domaines. Cette technique du génie logiciel permet une définition ou une réutilisation des patrons tout en prenant en considération les principes et les moyens d'une bonne conception liée aux aspects d'ergonomie et de performance. Ces patrons offrent les caractéristiques suivantes : réutilisabilité, uniformité, facilité d'utilisation et fiabilité.

Malgré le grand nombre des patrons de conception proposés dans la littérature, il n'existe pas un catalogue complet de patrons permettant à un auteur de concevoir une application hypermédia qui prend en considération tous leurs caractéristiques. Il est donc important d'avoir un catalogue de patrons dédié à ces applications. Ce catalogue doit contenir les patrons de base qui permettent de résoudre les problèmes de conception dans ce domaine.

L'ingénierie des composants réutilisables reste un domaine de recherche récent, tout comme le domaine d'ingénierie à base de patrons, ce qui justifie l'absence d'approches d'ingénierie des patrons dédiées à la conception des applications hypermédia. Notre objectif consiste à proposer une démarche d'ingénierie des patrons de conception pour la conception des applications hypermédia, puis utiliser cette démarche pour identifier les patrons de conception que notre catalogue doit les contenir.

Les patrons de notre catalogue sont organisés selon trois processus principaux qui sont :

- La construction de patron "Création d'applications hypermédia" : ce patron permet d'assurer la conception et la structuration des différents composants d'une application hypermédia.
- L'adaptation de deux patrons proposés dans la littérature qui sont : le patron "Stratégie de navigation" qui permet de définir des stratégies de navigation dans cette application hypermédia, et le patron "Décorateur" qui permet de présenter le contenu de l'hypermédia à un utilisateur.

Le patron "Observateur de navigation" : ce patron permet de générer une application hypermédia avec un historique de navigation pour faciliter le parcours de l'utilisateur.

Nous essayons de formaliser ces patrons à l'aide d'une notation XML (*eXtensible Markup Language*) pour les mieux gérer et les organiser. En ce qui concerne la validation de la démarche proposée et la faisabilité logicielle de notre catalogue, nous essayons d'implémenter le patron "Création d'applications hypermédia" à l'aide de langage Java.

### 3. Plan de travail :

Notre mémoire se divise en quatre chapitres organisés comme suit :

- Le premier chapitre constitue un état de l'art des principales caractéristiques des applications hypermédia, ainsi qu'une présentation de quelques méthodes de conception d'applications hypermédia proposées dans la littérature.
- Dans le chapitre 2, nous présenterons la dimension de la réutilisation et les principes de bases des patrons, ainsi que les différents formalismes pour modéliser un patron.
- Dans le chapitre 3, nous présenterons la démarche d'ingénierie des patrons de conception que nous avons utilisé pour l'identification des divers patrons constituant notre catalogue dédié à la conception des applications hypermédia.
- Le chapitre 4 illustre la validation de notre démarche par l'expérimentation de notre catalogue sur un exemple de modélisation relative aux applications hypermédia. Ce chapitre illustre aussi la formalisation de patrons de catalogue proposé à l'aide d'une notation XML et présente quelques aspects d'implémentation de patron "Création d'applications hypermédia"

## Conclusion et perspectives

Ce travail se situe dans le cadre de domaine des applications hypermédia et vise à mettre en place un catalogue de patrons dédié à la conception de ce type d'applications. Cela nous a conduit à étudier les caractéristiques de ces applications hypermédia. Nous avons défini les caractéristiques principales suivantes : la capacité de représenter les informations non structurées, la capacité de relier n'importe quelle pièce d'information par n'importe quelle autre, la capacité de lier un grand nombre d'unités d'information pour les partager entre les groupes collaboratives, la capacité de combiner les connaissances formelles avec les connaissances informelles et la capacité de naviguer entre les différentes unités d'information liées par des liens hypertextuels.

Toutes ces propriétés impliquent l'augmentation avec un taux inattendu de nombre, de taille moyenne et de la complexité de telles applications. Dans ce contexte, plusieurs méthodes de conception ont été proposées dans la littérature pour résoudre ces problèmes. D'autre part, l'investigation dans la littérature récente des techniques de conception et des approches de réutilisation, susceptibles de réaliser les besoins de modélisation, permet de fixer les éléments clés sur lesquels se base le cadre méthodologique proposé. Il s'agit alors de développer une démarche de conception des applications hypermédia basée sur la réutilisation de patrons.

L'introduction des patrons dans une démarche d'ingénierie des systèmes hypermédia nécessite de mettre en œuvre deux processus complémentaires :

- Un processus pour la réutilisation dédié à l'ingénierie des patrons. Il consiste à identifier, spécifier et organiser les patrons à utiliser durant la spécification des applications hypermédia.
- Un processus par la réutilisation dédié à l'ingénierie des applications hypermédia, partant des spécifications des utilisateurs. Ce processus facilite la recherche, la sélection, l'adaptation et l'intégration de patrons pour spécifier une application hypermédia.

Le catalogue de patrons dédié au domaine des applications hypermédia que nous avons proposé est constitué de quatre patrons suivants:

- Le patron "Création d'applications hypermédia" : ce patron permet d'assurer la conception, la structuration des différents composants d'une application hypermédia.
- Le patron "Stratégie de navigation" : ce patron permet de définir des familles d'algorithmes pour identifier des stratégies de navigation.
- Le patron "Décorateur" : ce patron a pour but d'ajouter des vues différentes à un contenu existant.

de navigation" : ce patron permet de construire un historique de navigation.

Ces patrons permettent de réutiliser des fragments de modèles représentant divers aspects dans les applications hypermédia. Par ailleurs, le formalisme adopté pour représenter les patrons favorise la sélection et l'organisation des patrons.

Notre contribution est basée sur les deux apports suivants :

- La définition d'une démarche générale pour la construction des patrons de conception. Si les patrons commencent à devenir un sujet commun du domaine de l'ingénierie des systèmes d'information, leur ingénierie pose actuellement de nombreux problèmes et la mise en œuvre de cette forme d'ingénierie constitue une problématique de recherche.
- La proposition d'un catalogue de patrons dédié à la conception des applications hypermédia qui capitalise les savoirs et les savoir-faire mis en œuvre dans l'ingénierie des applications hypermédia, couvre la complexité des connaissances gérées dans ces applications et propose diverses structurations pour organiser ces connaissances et facilite leur gestion.

Ce travail ouvre la voie à notre sens vers diverses perspectives de recherche. Il serait intéressant, dans un premier temps, de compléter le catalogue de patrons proposé par d'autres patrons raffinant les patrons actuels. Dans un second temps, et en ce qui concerne la formalisation des patrons à l'aide d'une notation XML, il serait intéressant d'utiliser cette description XML comme un point de départ d'implémentation des patrons de catalogue. Dans ce cadre là, il serait intéressant de développer un outil logiciel qui permet de transformer les diagrammes de classes UML des patrons en une description XML, puis exploiter cette description dans l'implémentation de l'ensemble de ces patrons. En ce qui concerne l'expérimentation de la démarche proposée, il serait intéressant d'implémenter tous les patrons de catalogue proposé et les appliquer dans des projets industriels de spécification des applications hypermédia.





**PDF**  
Complete

*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

## Bibliographie

- [Acerbis, 2008]: R. Acerbis, A. Bongio, M. Brambilla, S. Butti, S. Ceri, P. Fraternali, « Web applications design and development with WebML and WebRatio 5.0 », In *Proceedings of TOOLS Europe 2008*, 2008.
- [Avgeriou, 2003]: P. Avgeriou, A. Papasalouros, S. Retalis, M. Skordalakis, « Towards a Pattern Language for Learning Management Systems », *Educational Technology And Society*, 6(2), (ISSN 1436-4522), 2003, pp. 11-24.
- [Adams, 1995]: M. Adams, J. Coplien, R. Gamoke, R. Hanmer, F. Keeve, K. Nicodemus, « Fault- Tolerent Telecommunication System Patterns», AT&T Bell Laboratories, 1995.
- [Alexander, 1977]: C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-king, S. Aangel, « A Pattern Language», Oxford University Press, New York, NY, 1977.
- [Alexander, 1979]: C. Alexander, « The Timeless Way of Building », Oxford University Press, New York, NY, 1979.
- [Ambler, 1998]: S. Ambler, « Process Patterns: Building Large-Scale Systems Using Object Technology », Cambridge University Press, 1998.
- [Arnaud, 2004]: N. Arnaud, A. Front, D. Rieu, « Une approche par méta-modélisation pour l'imitation des patrons», *Inforsid'04*, Biarritz, 2004.
- [Apple, 2006]: Apple Computers, « Apple Human Interface Guidelines », 2006.
- [Badica, 2008]: C. Badica, E. Popescu, « Using Logic Programming and XML Technologies for Data Extraction from Web Pages », *Intelligence Integration in Distributed Knowledge Management*, Information Science Reference, 2008.
- [Balpe, 1990]: J. Balpe, «Hyperdocuments, hypertextes, hypermédias », Paris : Eyrolles, 1990.
- [Balpe, 1996]: J. Balpe, A. Lelu, F. Papy, I. Saleh, « Techniques avancées pour l'hypertexte », Paris : Hermès, 1996.
- [Beck, 1994]: K. Beck, R. Johnson, « Patterns Generate Architecture », *ECOOP04*, 1994.
- [Beck, 1997]: K. Beck, « Best Practice Patterns », Vol: 1: Coding, Prentice Hall, NJ, 1997.

M. Djoudi , « Création dynamique de documents hypermédias adaptatifs », Session : Le document numérique scientifique à des fins pédagogiques, CIDE 10, 2008. Disponible sur : <http://172.16.128.67:50010/cide/index.php?id=281>.

- [Booch, 1997]: G. Booch, « Ingénierie du logiciel avec ADA: de la conception à la réalisation », InterEditions, 1997.
- [Brillant, 2005]: A. Brilliant, « Introduction aux Design Patterns en Java », Tutorial d'introduction aux Design Patterns en Java, 2005. Disponible sur : <http://www.developpez.net/forums/>
- [Buschmann, 1996]: F. Buschmann, « What is a pattern? », Object Expert, 1996, pp.17-18.
- [Bush, 1945]: V. Bush, « As we may think », Atlantic Monthly, 1945, pp. 101-108.
- [Calvary, 2003]: G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, A. Vanderdonckt, « Unifying Reference Framework for Multi-Target User Interfaces », In Interacting with Computers, 2003, pp. 289-308.
- [Cauvet, 1999]: C. Cauvet, F. Semmak, « La réutilisation dans l'ingénierie des systèmes d'information », Chapitre 1 du livre Génie Objet, analyse et conception de l'évolution, Edition Hermes, 1999, pp. 25-55.
- [Cauvet, 2001]: C. Cauvet, D. Rieu, A. Front-Conte, P. Ramadour, « Réutilisation dans l'ingénierie des systèmes d'information », Chapitre 5 du livre Ingénierie des systèmes d'information, Editions Hermes, 2001, pp. 115-147.
- [Ceri, 2000]: S. Ceri, P. Fraternali, A. Bongio, « Web Modeling Language (WebML): a modeling language for designing web sites », Ninth World Wide Web Conference, 2000, pp. 137-157.
- [Chatlain, 1996]: J. Chatlain, « Du parnasse à l'amérique : L'imaginaire de l'encyclopédie à la Renaissance et à l'âge classique, In tous les savoirs du monde », Bibliothèque nationale de France, Flammarion, 1996.
- [Coad, 1992]: P. Coad, « Object-Oriented Patterns », Communications of the ACM, 1992, pp.152-159.
- [Coad, 1997]: P. Coad, D. North, M. Mayfield, « Object Models: Strategies, Patterns, and Applications », Prentice Hall, 1997.
- [Conte, 2001]: A. Conte, M. Fredj, J. Giraudin, D. Rieu , P-Sigma, «Un formalisme pour une représentation unifiée de patrons », Inforsid'01, Genève, 2001, pp. 67-86.

, « Patterns, the Patterns White Paper », Congrès INFORSID, Bordeaux, 1996.

- [DeTroyer, 1997]: O. De Troyer, C. Leune, « WSDM - A user-centered design method for Web sites », 7th International World Wide Web Conference, 1997.
- [Educnet, 2009]: Educnet, « Hypermédia et apprentissage, Notions d'hypertexte et d'hypermédia », 2009. Disponible sur : <http://www.educnet.education.fr/dossier/hypermédia/>
- [Englebart, 1984]: D. Englebart, « Authorship provisions in AUGMENT », IEEE Comp-Con Proceedings, Spring, 1984, pp. 465-472.
- [Fleming, 1998]: J. Fleming, « Web Navigation: Designing the User Experience », O'REILLY, 1998.
- [Forge, 2009]: Sourceforge Web site, 2009. Disponible sur:<http://sourceforge.net/projects>.
- [Fowler, 1997]: M. Fowler, « Analysis Patterns», Reusable Object Models, Addison-Wesley, 1997.
- [Gamma, 1991]: E. Gamma, « Object-Oriented Software Developments based on ET++: Design Patterns », Class Library, Tools, PhD thesis, University of Zurich, 1991.
- [Gamma, 1995]: E. Gamma, R. Helm, R. Johnson, J. Vlissides, « Design Patterns, Elements of reusable Object-Oriented Software », Addison-Wesley Publishing Company, 1995.
- [Garzotto, 1995]: F. Garzotto, L. Mainetti, P. Paolini, « Hypermedia design analysis and Evaluation Issues», Communications of ACM, vol 38(8), 1995.
- [German, 2000]: D. German, D. Cowan, « Towards a unified catalogue of hypermedia design patterns », Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.
- [Gilles, 2005]: H. Gilles, « De la conception d'hypermédia à la conception d'applications Web », Centre de Recherche en Architecture et Ingénierie, Nancy, 2005. Disponible sur : [http://stcief.univ-lemans.fr/num/vol2005/stcief\\_2005\\_som.htm](http://stcief.univ-lemans.fr/num/vol2005/stcief_2005_som.htm).
- [Gomez, 2000]: J. Gomez, C. Cachero, O. Pastor, « Extending a Conceptual Modelling Approach to Web Application Design », 12th International Conference CAiSE 2000, Stockholm, Sweden, 2000, pp. 144-173.

- Si-Said, C. Rolland, « MENTOR : un environnement pour l'ingénierie des méthodes et des besoins », Congrès INFORSID, Bordeaux, 1996.
- [Gzara, 2000]: L. Gzara, « Les patrons pour l'ingénierie des Systèmes d'Information Produit », Thèse de doctorat de l'INPG, spécialité Génie Industriel, 2000.
  - [Halin, 1999]: G. Halin , J. Bignon, P. Humbert, « Designing Hypermedia : An Experience in Multimedia Catalogue of Building Products », ACM Hypertext'99 Workshop on Hypermedia Development, Design Pattern in Hypermedia, Darmstadt, Germany, 1999.
  - [Hess, 1990]: J. Hess, S. Cohen, B. Holibaugh, K. Kyang, A. Peterson, W. Novk, P. Carroll, « A Domain Analysis Bibliography Software Engineering Institute », Special report, 1990.
  - [Isakowitz, 1995]: T. Isakowitz, E. Stohr, P. Balasubramanian, « A methodology for the design of structured hypermedia applications », Communications of ACM, 1995, p. 34-44.
  - [Khayati, 2005]: O. Khayati, « Modèles formels et outils génériques pour la gestion et la recherche de composants », Thèse de doctorat de l'INPG, 2005.
  - [Koch, 2000] : N. Koch, « The Authoring Process of the UML-based Web Engineering Approach », 1st International Workshop on Web-Oriented Software Technology, 2000.
  - [Limbourg, 2005] : Q. Limbourg, J. Vanderdonckt, B. Michotte, L. Bouillon, « UsiXML: a Language Supporting Multi-Path Development of User Interfaces », InProceedings of 9th IFIP Working Conference on Engineering for Human-ComputerInteraction jointly with 11th International Workshop on Design, Berlin, 2005, pp. 200-220.
  - [Larman, 2002]: C. Larman, « UML et les designs patterns», Campus press, 2002.
  - [Lee, 1998]: H. Lee, C. Lee , C. Yoo, « A scenario-based object-oriented methodology for developing hypermedia information systems», 31st Annual Conference on Systems Science, 1998.
  - [Marangozova, 2002]: V. Marangozova, « Patrons de conception pour la duplication de composants », Projet SARDES - INRIA Rhône-Alpes, 655 avenue de l'Europe, onthonnot St Martin, 38334 St Ismier Cedex ó France, 2002.
  - [Mathieu, 2007]: G. Mathieu, «Introduction aux Design Patterns », 2007. Disponible sur : <http://www.design-patterns.fr/-Design-Patterns-.html>.

- y, « Mass-produced software components », Software Engineering Concepts and Techniques (1968 NATO Conference on Software Engineering), In J.M Buxton, P. Naur, and B. Randell (Editors), 1976, pp. 88-89.
- [Millard, 2006]: D. Millard, M. Ross, « Web 2.0: Hypertext by any other name? » in *Proceedings of the ACM Conference Hypertext'06*, New York : ACM Press, 2006, pp. 27-30.
  - [Nanard, 1994]: M. Nanard, « L'apport des travaux de recherche dans les hypertextes aux techniques éducatives », Séminaire Hypermédias, éducation et formation : contribution à la structuration d'un champ émergent, 1994, pp. 14-17.
  - [Nanard, 1998]: M. Nanard, J. Nanard, P. Kahn, « Pushing Reuse in Hypermedia Design: Golden Rules, Design Patterns and Constructive Templates », LIRMM, Univ, Montpellier, France, 1998, pp. 15-34.
  - [Nelson, 1965]: T. Nelson, « A file structure for the complex, the changing and the indeterminate », in *Proceedings of the 20th ACM National Conference*, New York, ACM Press, 1965.
  - [Nelson, 1970]: T. Nelson, « No More Teachers' Dirty Looks », *Computer Decisions*, September, 1970, pp. 191-210.
  - [Olavo, 2007]: C. Olavo, A. Derycke, « Concevoir des Scénarios Pédagogiques Exécutables avec des Patrons de Conception Pédagogiques », Laboratoire Trigone, Université des Sciences et Technologies de Lille, 59655, Villeneuve d'Ascq cedex, Franc, 2007.
  - [Olsina, 1998]: L. Olsina, « Building a Web-based information system applying the hypermedia flexible process modeling strategy », *Conference Hypertext'98*, 1998.
  - [Palkovic, 2009]: R. Palkovic, « Design Patterns for Dynamic Languages on the JVM Machine », *Computer science conference*, 2009.
  - [Paterno, 2003]: F. Paterno, C. Santoro, « A Unified Method for Designing Interactive Systems Adaptable to Mobile and Stationary Platforms », In *Interacting with Computers*, vol:15, 2003, pp.362-369.
  - [Pawlak, 2004]: R. Pawlak, J. Retraillé, L. Seinturier, « Programmation orientée aspect pour Java », Eyrolles, 2004, p 446.
  - [Pernin, 2009]: J. Pernin, D. Rieu, « Définition de patrons pour la conception de scénarios d'apprentissage », *Sujet proposé dans : M2R ICCA*, 2009.
  - [Phillipe, 1996]: Q. Phillipe, « Un hypermédia pédagogique efficace », *Revue électronique de technologie éducative*, Université Laval, Québec, 1996.

patrons de conception à base d'aspects pour l'ingénierie des systèmes d'information par réutilisation », Thèse de doctorat préparée au sein de l'équipe SIGMA du laboratoire LSR-IMAG , Université Joseph Fourier, Grenoble I, 2006.

- [Raskin, 2000]: J. Raskin, « Presenting information », in Jacobson, R. (ed.) Information design, Cambridge, MA, États-Unis : MIT Press, 2000.
- [Reyes, 2007]: E. Reyes, « L'objet technique hypermédia : repenser la création de contenu éducatif sur le Web », Sciences de l'information et de la communication, Thèse de doctorat, Université de Paris VIII, 2007.
- [Rhéaume, 2009]: J. Rhéaume, « Les hypertextes et les hypermédias », Faculté des sciences de l'éducation, Université Laval, 2009. Disponible sur : <http://www.fse.ulaval.ca/fac/ten/revenduc/html/vol1/no2/heth.html>.
- [Rieu, 1999]: D. Rieu, J. Giraudin, C. Saint-Marcel, A. Front-Conte, « Des opérations et des relations pour les patrons de conception », Actes d'InforSID, Lagarde, France, 1999.
- [Rising, 2000]: L. Rising, « The Pattern Almanac 2000 », Software Pattern Series, Addison-Wesley, 2000.
- [Rossi, 1997]: G. Rossi, D. Schwab, A. Garrido, « Design reuse in hypermedia application development », Hypertext 97, Southampton UK, 1997, pp. 60-61.
- [Rouet, 1999]: J. Rouet, « Interactivité et compatibilité cognitive dans les systèmes hypermédias », Revue des Sciences de l'Éducation, 1999.
- [Sahraoui, 2007]: S. Sahraoui, « Problèmes d'intégration des hypertextes/hypermédias dans l'apprentissage des langues », Université de Blida, Algérie, 2007, pp. 252-253.
- [Schmidt, 1995]: D. Schmidt, « Using Design Pattern to develop Reusable Object-Oriented Communication Software », Communications of the ACM, vol: 38, 1995, p. 10.
- [Schmucker, 1987]: K. Schmucker, « Object Oriented Programming for the Macintosh », Addison-Wesley, 1987.
- [Schwabe, 1995]: D. Schwabe, G. Rossi, « Building hypermedia application as navigational views of information models », In proceedings of the 28th Hawaii International conference on system science ( HICSS'95), 1995.
- [Schwabe, 1996]: D. Schwabe, G. Rossi, S. Barbosa, « Systematic hypermedia design with OOHDM », In proceedings of Hypertext'96 Conference, Washington DC, USA, 1996, pp. 16-20.

- ...e, G. Rossi, « An Object Oriented Approach to Web-Based Application Design », Dept of Informatics, PUC-RIO, Brazil, 1998.
- [Semmak, 1998]: F. Semmak, « Réutilisation de composants de domaine dans la conception des systèmes d'information », Thèse de doctorat, Université Paris I, 1998.
  - [Spanoudakis, 1996]: K. Spanoudakis, P. Constanpopoulos, « Elaborating analogies from conceptual models », *International Journal of Intelligent Systems*, Vol: 11, 1996, p. 11.
  - [Stash, 2007]: N. Stash, « Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System », PhD Thesis, Eindhoven University of Technology, Netherlands, 2007.
  - [Takahashi, 1997]: K. Takahashi, E. Ling, « Analysis and Design of Web-based Information Systems », *Proc of Sixth International World Wide Web Conf*, 1997, pp. 377-389.
  - [Thomson, 1998]: J. Thomson, J. Greer, J. Cooke, « Algorithmically detectable design patterns for hypermedia collections », *Workshop on Hypermedia development Process, Methods and Models Hypermedia*, 1998.
  - [W3C, 2009]: World Wide Web Consortium (W3C), 2009. Disponible sur: <http://www.w3.org/>



Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

## Abstract

*Today, the systems multimedia and hypermedia constitute a support that can not be ignored in the development of advanced applications such as: e-commerce, web based information systems, the e-learning, the e-government and as well as others. They constitute an efficient means for use and the manipulation of data. Otherwise, design patterns technology showed its efficiency in the development of software dedicated to various domains. This technique of the of the software engineering permits a definition or a reuse of patterns while taking in consideration principles and means of a good design bound to aspects of ergonomics and performance. These design patterns offer the following features: Reutilisability, Uniformity, Easiness of use and Reliability. Nevertheless, a lot of problems remain unsolved. Among these last are the management of data multimedia synchronization, the hold in account of the user's profile or the way a user is guided through it in navigation. This axis of research remains to explore, and the proposed solutions endure a lack of maturity. The proposition of new conceptual tools or the enrichment of the existing solutions becomes more than necessary. The main objective of this work is therefore to propose a catalog of design patterns permitting the development of hypermedia applications.*

**Key words:** Design patterns, Catalog of patterns, Reutilisability, Hypermedia applications.



## Résumé

*Aujourd'hui, les systèmes multimédia et hypermédia constituent un support incontournable dans le développement des applications dites avancées telles que le e-commerce, les systèmes d'information basés Web, le e-learning, le e-government et bien d'autres. Ils constituent un moyen efficace pour l'utilisation et la manipulation des données. Par ailleurs, la technologie des patrons de conception a montré son efficacité dans le développement de logiciels dédiés à divers domaines. Cette technique de génie logiciel permettent une définition ou une réutilisation des patrons tout en prenant en considération les principes et les moyens d'une bonne conception liée aux aspects d'ergonomie et de performance. Ces patrons de conception offrent des caractéristiques suivantes : Réutilisabilité, Uniformité, Facilité d'utilisation et Fiabilité. Néanmoins, beaucoup de problèmes restent à résoudre. Parmi ces derniers, se trouvent la gestion de synchronisation des données multimédia, la prise en compte du profil de l'utilisateur ou le suivi de navigation. Cet axe de recherche reste à explorer, et les solutions proposées souffrent d'un manque de maturité. La proposition de nouveaux outils conceptuels ou l'enrichissement des solutions existantes devient plus que nécessaire. L'objectif principal de ce travail est donc de proposer un catalogue de patrons de conception permettant le développement d'applications hypermédia.*

**Mots clés :** Patrons de conception, Catalogue de patrons, Réutilisabilité, Applications hypermédia.

## ملخص

تشكل أنظمة الميلايميديا و الهيبارميديا في وقتنا الحالي دعامة أساسية لإنشاء التطبيقات المتطورة مثل: التجارة الالكترونية، أنظمة المعلومات المتمركزة على الشبكة، التعليم عن بعد، الحكومة الالكترونية و أنظمة أخرى. هذه الأنظمة تشكل وسيلة فعالة من أجل استعمال واستغلال المعطيات. في مقابل ذلك، نجد أن تقنية نماذج التصميم برهنت فعاليتها في إنشاء البرمجيات في مختلف الميادين. هذه التقنية الخاصة بهندسة البرامج تضمن تعريف أو إعادة استعمال النماذج آخذة بعين الاعتبار المبادئ و الوسائل التي تسمح بإنشاء تصميم جيد من ناحية المردودية و الناحية الجمالية. هذه النماذج الخاصة بالتصميم توفر الخصائص التالية: إعادة الاستعمال، التناسق، سهولة الاستعمال والفاعلية. مع ذلك تبقى مجموعة من الإشكاليات تحتاج إلى حلول من بينها: تسيير مزامنة معطيات الميلايميديا، أخذ جانب المستعمل في عين الاعتبار و تتبع ملاحظته في الشبكة الالكترونية. هذا المجال مازال مفتوحا للبحث و الحلول الموجودة تعاني من النقصان، إذن اقتراح أدوات تصميم جديدة أو تجميع الحلول الموجودة أصبح ضروريا. الهدف الأساسي لهذا العمل هو اقتراح فهرس خاص بنماذج التصميم يسمح بإنشاء تطبيقات الهيبارميديا.

**الكلمات المفتاحية:** نماذج التصميم، فهرس النماذج، إعادة الاستعمال، تطبيقات الهيبارميديا.