



HAL
open science

Adaptabilité dans les systèmes d'information sur le web : modélisation et mise en oeuvre de l'accès progressif

Marlène Villanova-Oliver

► To cite this version:

Marlène Villanova-Oliver. Adaptabilité dans les systèmes d'information sur le web : modélisation et mise en oeuvre de l'accès progressif. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 2002. Français. NNT: . tel-00006759

HAL Id: tel-00006759

<https://theses.hal.science/tel-00006759>

Submitted on 27 Aug 2004

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

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

N° attribué par la bibliothèque

□□□□□□□□□□□□□□

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Spécialité : Informatique

préparée au laboratoire Logiciels, Systèmes, Réseaux

dans le cadre de l'École Doctorale

Mathématiques, Informatique, Sciences et Technologie de l'Information

présentée et soutenue publiquement

par

Marlène VILLANOVA-OLIVER

le 18 décembre 2002

**Adaptabilité dans les systèmes d'information sur le web :
Modélisation et mise en œuvre de l'accès progressif**

—————
Directeur de Thèse : Hervé Martin
—————

JURY

| | |
|-----------------------|--------------------|
| M. Paul Jacquet, | Président |
| Mme Florence Sèdes, | Rapporteur |
| M. Michel Léonard, | Rapporteur |
| M. Hervé Martin, | Directeur de Thèse |
| M. Jérôme Gensel | Co-encadrant |
| Mme Danielle Ziébelin | Examineur |

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Spécialité : Informatique

préparée au laboratoire Logiciels, Systèmes, Réseaux

dans le cadre de l'École Doctorale

Mathématiques, Informatique, Sciences et Technologie de l'Information

présentée et soutenue publiquement

par

Marlène VILLANOVA-OLIVER

le 18 décembre 2002

**Adaptabilité dans les systèmes d'information sur le Web :
Modélisation et mise en œuvre de l'accès progressif**

—————
Directeur de Thèse : Hervé Martin
—————

JURY

| | |
|-----------------------|--------------------|
| M. Paul Jacquet, | Président |
| Mme Florence Sèdes, | Rapporteur |
| M. Michel Léonard, | Rapporteur |
| M. Hervé Martin, | Directeur de Thèse |
| M. Jérôme Gensel | Co-encadrant |
| Mme Danielle Ziébelin | Examineur |

REMERCIEMENTS

Je tiens à remercier

Monsieur Paul Jacquet, Professeur et Président de l'Institut National Polytechnique de Grenoble pour l'honneur qu'il me fait de présider mon jury de thèse. Je double mes remerciements à son égard pour les encouragements et la disponibilité dont il a fait preuve en tant que directeur du laboratoire Logiciels, Systèmes, Réseaux au sein duquel j'ai été accueillie durant ces trois années de thèse.

Madame Florence Sèdes, Professeur à l'Université Paul Sabatier de Toulouse, pour l'intérêt qu'elle a témoigné pour mon travail et pour avoir accepté d'être rapporteur de ma thèse. Chacune de nos rencontres est pour moi un enrichissement et un plaisir.

Monsieur Michel Léonard, Professeur à l'Université de Genève, pour s'être montré particulièrement intéressé par mon travail et pour avoir accepté de le rapporter. Je le remercie pour les remarques et les conseils qu'il a formulés à propos de mon manuscrit.

Madame Danielle Ziébelin, Maître de Conférences à l'Université Joseph Fourier, pour avoir accepté de participer à mon jury de thèse. Je lui suis très reconnaissante de s'être intéressée à mes travaux.

Monsieur Hervé Martin, Professeur à l'Université Joseph Fourier, pour m'avoir permis de faire mes premiers pas dans le monde de la recherche et pour m'avoir incitée à poursuivre dans cette voie. Je le remercie pour son soutien sans faille dans la direction de cette thèse et pour avoir guidé ce travail en conjuguant habilement disponibilité, conseils et critiques constructives tout en laissant libre cours à ma créativité. Je sais avoir beaucoup appris grâce à lui.

Monsieur Jérôme Gensel, Maître de Conférences à l'Université Pierre-Mendès France, pour avoir fait preuve d'une très grande disponibilité et montré un réel intérêt pour mon travail. Grâce à nos discussions, j'ai *accédé progressivement* au niveau de confiance nécessaire à la rédaction d'une thèse. Je le remercie pour en avoir relu minutieusement les lignes me permettant ainsi d'en améliorer le contenu.

Tous ceux et toutes celles qui par leur présence, leur sourire, leur soutien et leur confiance ont fait que je n'étais pas seule pour parcourir ce chemin. J'ai ainsi une pensée pour :

Les personnels administratifs et autres membres du laboratoire LSR pour leur gentillesse et leurs encouragements de tous les instants.

Les membres de l'équipe SIGMA qui ont toujours été présents et prêts à m'apporter leur aide. Je ne peux les citer tous, mais que chacun d'eux sache combien je l'apprécie et combien il m'est agréable de faire partie de cette équipe. Un merci tout particulier à Christelle et Marius pour leur investissement dans le développement du prototype KIWIS.

Tous mes ami(e)s, et notamment :

Caroline, pour avoir préservé notre amitié ces derniers mois malgré mon indisponibilité et avoir été un coach exceptionnel rythmant l'avancement de cette thèse par un « *combien t'as de pages ?* » hebdomadaire redoutable.

Flora, complice d'une précieuse amitié de plus de 18 ans (*quel coup de vieux !...*), pour avoir toujours été là, au-delà de cette thèse, dans la vie tout simplement.

Hervé et Jérôme pour être, aussi et surtout, les amis dont je partage, entre autres, le bureau, l'effervescence des jours de dead-line, les fous-rires et peut être bientôt une choucroute. Face au jury de sélection qu'ils formaient lors de notre première rencontre en 1997, je n'ai pas imaginé le destin nous réunirait pour une thèse. Je suis heureuse qu'il en ait été ainsi.

Je tiens également à remercier ma p'tite Martine, Ibtissem (*à toi de jouer !!*), Paule-Annick (*« Je te ramène quelque chose à manger ?! »*), Jean-Pierre (*« Merci chef ! »*), Dominique (*« Tu es en train de me dire que tu es humaine ?!! »*), mes deux Gisèle préférées (*« Salut ma belle ! »* x 2), Christophe (*génial ré-équilibrateur énergétique*), Jérôme (*« Une vraie mère pour moi ! »*), les Gallina (*petits et grands*), Rolande, Jean et Brigitte (*ma très belle belle-famille*) et Hervé (*« Salut c'est ton frère ! Alors cette thèse, ça avance ? »*).

Je remercie affectueusement Sylviane et Bruno qui m'ont toujours soutenue dans mes choix. Je les sais fiers de moi. Qu'ils sachent en retour que je suis fière d'être leur fille.

Enfin et surtout, Loïc, pour avoir su composer remarquablement et patiemment avec mon caractère de fin de thèse, pour avoir à chaque fois trouvé les mots que j'avais besoin d'entendre et pour tous les *« véritables petits moments de bonheur »* que nous partageons.

Il y a un peu de chacun de vous dans cette thèse.

TABLE DES MATIÈRES

| | |
|--|-----------|
| 1. INTRODUCTION | 1 |
| Contexte de ce travail | 1 |
| Problématique | 2 |
| Aperçu de la proposition | 3 |
| Organisation du document | 5 |
| | |
| ÉTAT DE L'ART | 7 |
| | |
| 2. L'ADAPTATION DANS LES SIW | 9 |
| 2.1. Systèmes d'Information basés sur le Web | 9 |
| 2.1.1 L'influence des hypertextes et hypermédias | 9 |
| 2.1.2 Typologie des Applications basées sur le Web | 11 |
| 2.1.3 SI traditionnels versus SIW | 13 |
| 2.2. Les dimensions de l'adaptation | 14 |
| 2.2.1 Adaptation du contenu délivré | 14 |
| 2.2.2 Adaptation de la navigation | 15 |
| 2.2.3 Adaptation de la présentation | 17 |
| 2.2.4 Adaptation des fonctionnalités | 18 |
| 2.2.5 Gestion des conditions matérielles | 18 |
| 2.3. Les différents processus d'adaptation | 19 |
| 2.4. La modélisation des utilisateurs | 21 |
| 2.4.1 Classification des modèles de l'utilisateur | 22 |
| 2.4.2 Techniques de représentation des modèles utilisateur | 23 |
| 2.5. L'adaptation dans différents domaines | 24 |
| 2.5.1 Recherche d'Information | 24 |
| 2.5.2 E-Business | 25 |
| 2.5.3 Enseignement à distance | 26 |
| 2.6. Conclusion sur l'adaptation dans les SIW | 28 |
| | |
| 3. MÉTHODES DE CONCEPTION D'APPLICATIONS BASÉES SUR LE WEB | 29 |
| 3.1. RMM : Relationship Management Methodology | 30 |
| 3.1.1 Aperçu général de RMM | 30 |
| 3.1.2 Le modèle RMDM | 31 |
| 3.1.3 Les étapes de la méthode | 31 |
| 3.1.4 Limites et extensions de la version originale de RMM | 32 |
| 3.1.4.1 Limite des <i>slices</i> et leur évolution : les <i>m-slices</i> | 32 |
| 3.1.4.2 Redéfinition du processus de conception | 32 |
| 3.1.5 Conclusion sur RMM | 33 |

| | |
|--|-----------|
| 3.2. Projet Hera | 34 |
| 3.2.1 Aperçu général de Hera | 34 |
| 3.2.2 Conception de la présentation | 34 |
| 3.2.2.1 Le niveau logique | 34 |
| 3.2.2.2 Le niveau présentation | 35 |
| 3.2.3 Conception des adaptations | 35 |
| 3.2.4 Conclusion sur Hera | 36 |
| 3.3. OOHDM : Object Oriented Hypermedia Design Methodology | 36 |
| 3.3.1 Aperçu général de OOHDM | 36 |
| 3.3.2 Collecte des Besoins | 37 |
| 3.3.3 Modélisation conceptuelle | 38 |
| 3.3.4 Conception de la navigation | 38 |
| 3.3.4.1 Modélisation de la navigation correspondant à une tâche | 38 |
| 3.3.4.2 Production du diagramme de contexte navigationnel de l'application | 39 |
| 3.3.4.3 Spécification du modèle de navigation de l'application | 39 |
| 3.3.5 Conception de l'interface abstraite | 39 |
| 3.3.6 Implémentation | 39 |
| 3.3.7 Conclusion sur OOHDM | 40 |
| 3.4. WSDM: Web Site Design Method | 40 |
| 3.4.1 Aperçu général de WSDM | 40 |
| 3.4.2 Modélisation du public | 42 |
| 3.4.3 Modélisation conceptuelle | 42 |
| 3.4.3.1 Modélisation objet et modélisation du domaine | 42 |
| 3.4.3.2 Conception Navigationnelle | 42 |
| 3.4.4 Modélisation en vue de l'implémentation | 43 |
| 3.4.5 Conclusion sur WSDM | 43 |
| 3.5. Adaptive Web-based Information System - Method – AWIS-M | 43 |
| 3.5.1 Aperçu général de AWIS-M | 43 |
| 3.5.2 Les modèles de produits | 44 |
| 3.5.2.1 Les modèles de produits de niveau conceptuel | 44 |
| 3.5.2.2 Les modèles de produits de niveau logique | 47 |
| 3.5.3 La démarche proposée | 50 |
| 3.5.4 Conclusion sur AWIS-M | 51 |
| 3.6. UML-based Web Engineering – UWE | 51 |
| 3.6.1 Aperçu général d'UWE | 51 |
| 3.6.2 Analyse des besoins | 52 |
| 3.6.3 Modélisation Conceptuelle | 53 |
| 3.6.4 Modélisation de la Navigation | 53 |
| 3.6.4.1 Modélisation de l'espace de navigation | 53 |
| 3.6.4.2 Modélisation de la structure de navigation | 54 |
| 3.6.5 Modélisation de la Présentation | 56 |
| 3.6.6 Modélisation des utilisateurs | 57 |
| 3.6.7 Modélisation de l'Adaptation | 57 |
| 3.6.8 Conclusion sur UWE | 58 |
| 3.7. Web Modeling Language – WebML | 58 |
| 3.7.1 Aperçu général de WebML | 58 |
| 3.7.2 Le modèle structurel | 59 |
| 3.7.3 Le modèle de dérivation | 59 |
| 3.7.4 Le modèle de composition | 59 |
| 3.7.5 Le modèle de navigation | 60 |
| 3.7.6 Extensions apportées aux modèles de composition et de navigation | 60 |

| | | |
|-------------|--|-----------|
| 3.7.7 | La définition de la présentation | 60 |
| 3.7.8 | La personnalisation | 61 |
| 3.7.9 | Processus de conception avec WebML | 61 |
| 3.7.10 | Conclusion sur WebML | 62 |
| 3.8. | Conclusion sur les méthodes présentées | 62 |
| 4. | SYNTHÈSE DE L'ÉTAT DE L'ART ET AXES D'ÉTUDE | 65 |
| 4.1. | Modélisation des utilisateurs | 65 |
| 4.1.1 | Critères d'évaluation | 65 |
| 4.1.2 | Évaluation des méthodes | 66 |
| 4.2. | Dimensions adaptées | 68 |
| 4.2.1 | Critères d'évaluation | 68 |
| 4.2.2 | Évaluation des méthodes | 68 |
| 4.3. | Modélisation et mise en œuvre de l'adaptation | 70 |
| 4.3.1 | Critères d'évaluation | 70 |
| 4.3.2 | Évaluation des méthodes | 70 |
| 4.4. | Capacités d'adaptation du SIW conçu | 71 |
| 4.4.1 | Critères d'évaluation | 71 |
| 4.4.2 | Évaluation des méthodes | 72 |
| 4.5. | Conclusion et aperçu des principaux axes de notre proposition | 73 |
| 4.5.1 | Prise en compte de l'adaptation et modélisation multidimensionnelle | 74 |
| 4.5.2 | Adaptation minimale, adaptabilité et adaptativité du SIW | 74 |
| 4.5.3 | Contribution de l'accès progressif aux capacités d'adaptation du SIW | 75 |
| 4.5.4 | Un environnement de conception supportant l'approche | 75 |
| | PROPOSITION | 77 |
| 5. | MODÈLE D'ACCÈS PROGRESSIF | 79 |
| 5.1. | Description du Modèle d'Accès Progressif | 80 |
| 5.1.1 | Notions pour l'accès progressif | 80 |
| 5.1.1.1 | Entité Masquable et Représentations d'Entité Masquable | 80 |
| 5.1.1.2 | Stratification | 82 |
| 5.1.1.3 | Définitions de fonctions pour l'accès progressif | 82 |
| 5.1.2 | Modélisation UML du MAP | 85 |
| 5.2. | Protocole d'exploitation du MAP | 86 |
| 5.2.1 | Opérations de définition d'une stratification | 87 |
| 5.2.1.1 | Création d'une stratification | 87 |
| 5.2.1.2 | Suppression d'une stratification | 88 |
| 5.2.2 | Opérations d'accès progressif aux stratifications | 89 |
| 5.2.2.1 | Sélection d'une stratification | 89 |
| 5.2.2.2 | Masquage et dévoilement d'une stratification | 90 |
| 5.2.3 | Opérations de réorganisation d'une stratification | 91 |
| 5.2.3.1 | Importation d'un EEM dans une REM | 91 |
| 5.2.3.2 | Suppression d'un EEM d'une REM | 92 |
| 5.2.3.3 | Ajout d'une REM | 92 |
| 5.2.3.4 | Suppression d'une REM | 93 |
| 5.2.3.5 | Fusion de REM | 93 |
| 5.2.3.6 | Fractionnement d'une REM | 94 |
| 5.2.3.7 | Déplacement de REM | 95 |

| | |
|--|------------|
| 5.3. Application du MAP à un modèle de données | 95 |
| 5.3.1 Descriptions en intension et en extension du modèle de données | 97 |
| 5.3.2 Stratifications applicables à un modèle de données | 99 |
| 5.3.2.1 Stratifications en intension <i>versus</i> en extension | 99 |
| 5.3.2.2 Combinaison de stratifications | 100 |
| 5.3.3 Stratification en intension d'une EM | 101 |
| 5.3.3.1 Description | 101 |
| 5.3.3.2 Exemples de stratification en intension | 102 |
| 5.3.4 Stratification en extension d'une EM | 104 |
| 5.3.4.1 Description | 104 |
| 5.3.4.2 Exemples de stratification en extension | 106 |
| 5.3.5 Règles de visibilité du contenu en fonction d'une stratification | 108 |
| 5.3.5.1 Visibilité des valeurs de rôles d'association | 108 |
| 5.3.5.2 Gestion des hiérarchies de classes et d'associations | 109 |
| 5.4. Conclusion sur le Modèle d'Accès Progressif | 110 |
| 6. MODÈLES DE CONCEPTION POUR L'ACCÈS PROGRESSIF | 113 |
| 6.1. Modèle du Domaine | 114 |
| 6.2. Le Modèle des Fonctionnalités | 114 |
| 6.2.1 Les concepts du Modèle des Fonctionnalités | 115 |
| 6.2.2 Modèle des Fonctionnalités et Accès progressif | 116 |
| 6.2.3 Fonctionnalité de consultation avec accès progressif | 117 |
| 6.2.3.1 Modélisation de la requête utilisée par une fonctionnalité de consultation | 117 |
| 6.2.3.2 Stratification du résultat de la requête | 119 |
| 6.2.3.3 Stratifications types : définitions et exemples | 120 |
| 6.2.4 Conclusion sur le Modèle des Fonctionnalités | 124 |
| 6.3. Le Modèle de l'Hypermédia | 124 |
| 6.3.1 Le Modèle de Structuration & Navigation des Accès Progressifs | 125 |
| 6.3.1.1 Les concepts pour la structuration | 125 |
| 6.3.1.2 Les concepts pour la navigation | 126 |
| 6.3.1.3 Exemple | 128 |
| 6.3.1.4 Augmentation du modèle par le concepteur | 129 |
| 6.3.2 Modélisation de la présentation de l'Hypermédia | 129 |
| 6.3.2.1 Chartes de composition | 130 |
| 6.3.2.2 Charte graphique | 132 |
| 6.3.3 Conclusion sur le Modèle de l'Hypermédia | 133 |
| 6.4. Le Modèle des Utilisateurs | 133 |
| 6.4.1 Distinction entre Groupes et Utilisateurs | 134 |
| 6.4.1.1 Motivations | 134 |
| 6.4.1.2 Une distinction basée sur une approche fonctionnelle | 134 |
| 6.4.2 Profils de groupes et d'utilisateurs | 135 |
| 6.4.3 Adaptabilité des stratifications : les Profils d'Accès Progressif | 136 |
| 6.4.3.1 Le lien entre le MAP et le Modèle des Utilisateurs | 136 |
| 6.4.3.2 Description des Profils d'Accès Progressif | 137 |
| 6.4.3.3 Impact de l'adaptabilité sur la gestion des stratifications | 138 |
| 6.4.4 Adaptabilité de la présentation : les Profils d'Accès Progressif et de Préférences | 139 |
| 6.4.4.1 Adaptabilité de la composition résultant des profils d'Accès Progressif | 139 |
| 6.4.4.2 Adaptabilité de l'apparence grâce aux profils de Préférences | 141 |
| 6.4.5 Adaptativité du SIW : les profils de Comportement. | 143 |
| 6.4.6 Conclusion sur le Modèle des Utilisateurs | 144 |
| 6.5. Conclusion | 144 |

| | |
|--|------------|
| 7. CONCEPTION DE SIW A ACCÈS PROGRESSIF | 147 |
| 7.1. Relations entre les modèles de conception | 147 |
| 7.2. Démarche de conception | 149 |
| 7.3. Rôle central du MAP dans la conception de SIW | 151 |
| 7.4. Comparaison de la proposition avec l'état de l'art | 152 |
| 7.4.1 Modélisation de l'utilisateur | 153 |
| 7.4.2 Dimensions adaptées | 154 |
| 7.4.3 Modélisation et mise en œuvre de l'adaptation | 155 |
| 7.4.4 Capacités d'adaptation du SIW | 156 |
| 7.4.5 Comparaison des Concepts d'Accès Progressif et d'autres concepts pour la représentation des données | 157 |
| 7.5. Conclusion sur la conception de SIW à Accès Progressif | 158 |

RÉALISATION 159

| | |
|--|------------|
| 8. L'ENVIRONNEMENT KIWIS | 161 |
| 8.1. Architecture du prototype | 161 |
| 8.1.1 Architecture Logicielle | 161 |
| 8.1.1.1 Intégration d'outils existants | 162 |
| 8.1.1.2 Le prototype KIWIS | 163 |
| 8.1.1.3 Stockage des données | 164 |
| 8.1.2 Architectures Fonctionnelles | 164 |
| 8.1.2.1 Architecture fonctionnelle de conception | 164 |
| 8.1.2.2 Architecture fonctionnelle d'utilisation | 166 |
| 8.2. L'environnement du concepteur | 167 |
| 8.2.1 Connexion et page d'accueil du concepteur | 167 |
| 8.2.2 Création d'un nouveau SIW | 169 |
| 8.2.3 Interface de définition du Modèle du Domaine | 171 |
| 8.2.4 Interface de définition du Modèle des Fonctionnalités | 172 |
| 8.2.5 Interface de définition du Modèle des Utilisateurs | 174 |
| 8.2.6 Interface de définition des profils d'Accès Progressif | 175 |
| 8.2.7 Interface de définition du Modèle de l'Hypermédia | 176 |
| 8.2.7.1 Page de définition du Modèle de Structuration & Navigation. | 177 |
| 8.2.7.2 Page de définition de la charte de composition | 178 |
| 8.2.7.3 Page de définition de la charte graphique | 179 |
| 8.2.8 Génération du SIW | 179 |
| 8.3. Utilisation d'un SIW généré par KIWIS | 180 |
| 8.3.1 Navigation dans un SIW à Accès Progressif | 180 |
| 8.3.1.1 Dévoilement et Masquage de l'Espace Fonctionnel | 181 |
| 8.3.1.2 De l'Espace Fonctionnel à un Rôle Fonctionnel | 182 |
| 8.3.1.3 Dévoilement et Masquage d'un Rôle Fonctionnel à une Fonctionnalité | 182 |
| 8.3.1.4 Du Rôle Fonctionnel à une Fonctionnalité | 183 |
| 8.3.2 Gestion du profil d'Accès Progressif | 185 |
| 8.4. Conclusion sur l'environnement KIWIS | 186 |

| | |
|---|------------|
| CONCLUSIONS | 189 |
| 9. BILAN ET PERSPECTIVES | 191 |
| Bilan du travail réalisé | 191 |
| Perspectives | 194 |
| Amélioration du prototype KIWIS | 194 |
| Capacités d'Adaptation avancées pour l'Accès Progressif | 195 |
| BIBLIOGRAPHIE | 197 |
| ANNEXES | 205 |

TABLE DES ILLUSTRATIONS

FIGURES

| | |
|---|-----|
| Figure 2.1 – Classification des sites et applications basées sur le Web | 12 |
| Figure 2.2 – Exemple d'un profil CC/PP décrit en XML | 19 |
| Figure 2.3 – Différents types de processus d'adaptation : de l'adaptabilité à l'adaptativité | 21 |
| Figure 2.4 – Représentation de la proposition de standard PAPI Learner | 27 |
| Figure 3.1 – Le processus de conception de RMM | 33 |
| Figure 3.2 – Les étapes de OOHDM dans sa version proposée en 2000 | 37 |
| Figure 3.3 – Les différentes étapes de la méthode WSDM | 41 |
| Figure 3.4 – Les étapes de la démarche dans AWIS-M | 44 |
| Figure 3.5 – Exemples de Représentation de Catégories d'Utilisateurs dans AWIS-M | 46 |
| Figure 3.6 – Expression logique d'un paramètre de profil dans AWIS-M | 48 |
| Figure 3.7 – Descriptions conceptuelle et logique d'un profil dans AWIS-M | 48 |
| Figure 3.8 – Les modèles construits selon la méthode d'analyse et de conception de UWE | 52 |
| Figure 3.9 – Du modèle de l'espace de navigation au modèle de structure de la navigation selon UWE | 55 |
| Figure 4.1 – Typologie retenue pour les capacités d'adaptation des SIW | 72 |
| Figure 5.1 – Entité Masquable et Représentations d'Entité Masquable. | 81 |
| Figure 5.2 – Mécanismes de masquage et de dévoilement appliqués aux REM | 84 |
| Figure 5.3 – Profil UML proposé pour la stratification en vue de l'accès progressif | 85 |
| Figure 5.4 – Notations graphiques des cas d'utilisation pour le MAP | 87 |
| Figure 5.5 – Diagramme de Cas d'Exploitation du MAP : Création d'une Stratification | 87 |
| Figure 5.6 – Diagramme de séquence simplifié de la création d'une stratification | 88 |
| Figure 5.7 – Diagramme de Cas d'Exploitation du MAP : Opérations pour l'accès progressif | 89 |
| Figure 5.8 – Extrait du MAP: introduction des attributs et opérations des classes Stratification et REM | 90 |
| Figure 5.9 – Exemple de Stratification d'une EM en REM | 91 |
| Figure 5.10 – Réorganisation d'une stratification par ajout d'une nouvelle REM | 92 |
| Figure 5.11 – Exemple de réorganisation d'une stratification par fusion de REM | 94 |
| Figure 5.12 – Exemple de réorganisation d'une stratification par fractionnement d'une REM | 95 |
| Figure 5.13 – Exemple de modèle de données : Gestion Documentaire d'Inondations | 96 |
| Figure 5.14 – Exemples d'instances et de tuples définis pour le modèle de données 'GDI' | 96 |
| Figure 5.15 – EM potentielles et REM correspondantes pour stratifier en intension une Classe et une Association du Modèle de Données | 102 |
| Figure 5.16 – Exemples de stratification d'une Classe considérée comme une Entité Masquable. | 103 |
| Figure 5.17 – Visibilité des instances d'une classe stratifiée en intension | 103 |
| Figure 5.18 – EM potentielles et REM correspondantes pour stratifier en extension un Modèle de données, une Classe, Association et Classe-Association de ce modèle. | 105 |
| Figure 5.19 – Exemple de stratification en extension d'une EM 'Modèle de données ' | 106 |
| Figure 5.20 – Visibilité des instances pour une stratification en extension d'une classe | 107 |
| Figure 5.21 – Exemples pour la gestion des associations | 109 |

| | |
|---|-----|
| Figure 5.22 – Exemples pour la gestion des hiérarchies de classes _____ | 110 |
| Figure 6.1 – Concepts du Modèle des Fonctionnalités obtenus à partir des cas d'utilisation _____ | 116 |
| Figure 6.2 – Modélisation d'une Fonctionnalité de Consultation avec Accès Progressif _____ | 118 |
| Figure 6.3 – Stratifications en intension et en extension d'une EM Résultat en référence au MAP _____ | 119 |
| Figure 6.4 – Stratification d'une projection avec imbrication et mécanismes de masquage/dévoilement _____ | 123 |
| Figure 6.5 – Les concepts de structuration du Modèle de Structuration & Navigation _____ | 126 |
| Figure 6.6 – Représentation du Modèle de Structuration & Navigation _____ | 127 |
| Figure 6.7 – Représentation du Modèle S&N dérivé à partir du Modèle des Fonctionnalités instancié pour un utilisateur particulier. _____ | 128 |
| Figure 6.8 – Représentation graphique de la charte de composition par défaut _____ | 131 |
| Figure 6.9 – Modèle des Utilisateurs : distinction entre Groupes et Utilisateurs basée sur les concepts du Modèle des Fonctionnalités _____ | 135 |
| Figure 6.10 – Le Modèle des Utilisateurs : profils décrits pour les groupes et les utilisateurs _____ | 136 |
| Figure 6.11 – Modélisation pour une stratification adaptée aux groupes et aux utilisateurs _____ | 137 |
| Figure 6.12 – Contexte extrait du Modèle S&N correspondant à un Rôle Fonctionnel _____ | 140 |
| Figure 6.13 – Contexte pour un profil d'Accès Progressif différent _____ | 141 |
| Figure 6.14 – Deux exemples de pages créées à partir du même contexte d'un Modèle S&N _____ | 142 |
| Figure 7.1 – Relations entre les différents modèles de conception _____ | 148 |
| Figure 7.2 – Étapes de la démarche de conception préconisée _____ | 149 |
| Figure 7.3 – Relations entre les modèles de conception et le MAP _____ | 152 |
| Figure 8.1 – Architecture de l'environnement KIWIS. _____ | 162 |
| Figure 8.2 – Architecture fonctionnelle de KIWIS : Relations des Gestionnaires lors de la Conception _____ | 165 |
| Figure 8.3 – Architecture Fonctionnelle de KIWIS lors de l'appel à une fonctionnalité du SIW _____ | 167 |
| Figure 8.4 – Page de connexion à l'environnement KIWIS _____ | 168 |
| Figure 8.5 – La page d'accueil du concepteur dans KIWIS _____ | 169 |
| Figure 8.6 – La page de création d'un nouveau SIW _____ | 169 |
| Figure 8.7 – Structure du répertoire créé pour tout nouveau SIW _____ | 170 |
| Figure 8.8 – Page « Wizard » présentant la démarche de conception _____ | 170 |
| Figure 8.9 – Interface de création d'une classe du Modèle du Domaine _____ | 171 |
| Figure 8.10 – Interface de création d'une association du Modèle du Domaine. _____ | 172 |
| Figure 8.11 – Interface de création d'une fonctionnalité. _____ | 173 |
| Figure 8.12 – Interface de définition des Groupes du Modèle des Utilisateurs _____ | 174 |
| Figure 8.13 – Extrait de l'interface générée lorsqu'un qu'un groupe n'est pas associé à un Rôle Fonctionnel _____ | 175 |
| Figure 8.14 – Interface de gestion des Profils d'Accès Progressifs _____ | 176 |
| Figure 8.15 – Page de présentation du Modèle de Structuration & Navigation _____ | 177 |
| Figure 8.16 – Page de définition des chartes de composition - Définition d'une page type _____ | 179 |
| Figure 8.17 – Page d'accueil de KIWIS : présentation des SIW accessibles à un utilisateur _____ | 180 |
| Figure 8.18 – Page d'accueil d'un utilisateur : Présentation de son espace fonctionnel _____ | 181 |
| Figure 8.19 – Pages d'accès à un rôle fonctionnel : exemple de navigation entre deux niveaux de détail de ce rôle _____ | 183 |
| Figure 8.20 – Page dédiée à la Fonctionnalité de consultation des Rapports. _____ | 184 |
| Figure 8.21 – Extrait de la page obtenue suite au dévoilement de la Fonctionnalité de consultation des Rapports _____ | 185 |
| Figure 8.22 – Extrait de la page obtenue suite au dévoilement d'une stratification combinée _____ | 185 |
| Figure 8.23 – Gestion de son profil d'Accès Progressif par l'utilisateur. Exemple de réorganisation d'une stratification existante. _____ | 186 |

TABLEAUX

| | |
|---|-----|
| Tableau 4.1 – Critères d'évaluation relatifs à la modélisation des utilisateurs _____ | 66 |
| Tableau 4.2 – Évaluation des méthodes concernant la modélisation des utilisateurs. _____ | 67 |
| Tableau 4.3 – Critères d'évaluation relatifs aux dimensions adaptées _____ | 68 |
| Tableau 4.4 – Évaluation des méthodes concernant les dimensions adaptées _____ | 69 |
| Tableau 4.5 – Évaluation des méthodes concernant la modélisation et la mise en œuvre de l'adaptation _____ | 71 |
| Tableau 4.6 – Critères d'évaluation relatifs aux capacités d'adaptation du SIW _____ | 72 |
| Tableau 4.7 – Évaluation des capacités d'adaptation des SIW conçus avec les méthodes étudiées. _____ | 73 |
| Tableau 7.1 – Évaluation de notre proposition concernant la modélisation des utilisateurs. _____ | 154 |
| Tableau 7.2 – Évaluation de notre proposition concernant les dimensions adaptées _____ | 155 |
| Tableau 7.3 – Évaluation de notre proposition concernant la modélisation et la mise en œuvre de l'adaptation _____ | 156 |
| Tableau 7.4 – Évaluation des capacités d'adaptation des SIW conçus avec notre proposition. _____ | 156 |

1. INTRODUCTION

Contexte de ce travail

L'avènement du *Web*, au début des années 90, a constitué une révolution permettant d'envisager différemment la diffusion et le traitement de l'information. En tant qu'application reposant sur l'Internet, le Web offre un espace d'information à la mesure des possibilités offertes par cette infrastructure. A l'origine, sa mise en œuvre résulte de travaux autour du concept d'*hypertexte* dont il reprend les principes d'organisation non-linéaire des contenus informationnels hébergés sur des machines distantes et accessibles par de multiples chemins de navigation.

Les hypertextes et les hypermédias, dans lesquels les informations ne sont pas seulement de type texte, mais également de type image, son, vidéo ou encore multimédia, ont vocation à *présenter* l'information. Dans les *applications basées sur le Web*, l'information est présentée selon une approche hypermédia, mais de plus, il est possible d'effectuer des *traitements* à travers l'interface Web. Ces applications peuvent constituer une composante du Système d'Information (SI) d'une organisation. On parle alors de Systèmes d'Information basés sur le Web (que nous désignons dans ce mémoire par SIW) [Taka97][Isak98a].

Tout comme un SI traditionnel, un SIW forme un ensemble constitué de *ressources humaines* et de *moyens techniques*, de *règles de fonctionnement*, de *données* et de *procédés* pour acquérir, mémoriser, transformer, rechercher, communiquer et restituer ces données [Roll88]. Dans ce cadre, le Web constitue un support pour la mise en œuvre de systèmes qui sont donc, avant tout, des SI.

Un élément déterminant de l'approche SIW par rapport à une approche SI traditionnelle influence cependant nettement le processus de conception. Les SIW mettent en œuvre, comme les hypermédias, une organisation non-linéaire d'informations reliées entre elles par plusieurs chemins, ces derniers multipliant les possibilités d'accès à une même information.

La conception des SIW doit, en conséquence, porter sur les deux dimensions suivantes :

- une activité d'organisation de l'information en vue de sa *présentation* qui, en tant que support de la *navigation*, ne peut être ignorée dans le développement des SIW,
- une activité consacrée spécifiquement aux aspects fonctionnels, c'est-à-dire à la conception des services auxquels doit répondre le SIW, est nécessaire.

Cependant, par les caractéristiques qui distinguent les SIW des SI, le recours aux méthodes traditionnelles de conception est à la fois difficile et inadéquat. En effet, les aspects relatifs à la présentation et à la navigation ne sont pas une priorité dans un processus de développement de SI non basés sur le Web. A l’opposé, les méthodes de conception d’hypermédias ont porté une attention particulière à la fois à la structuration des pages, à l’élaboration des liens entre celles-ci pour la navigation, et à leur apparence lors de la présentation. Néanmoins, la conception d’un SIW ne peut pas être envisagée comme une activité triviale de manipulation de médias et de création de présentations. Pourtant, Zelnick met en évidence combien cela est observé dans les faits à travers une étude visant à évaluer les applications sur le Web [Zeln98]. Aussi, bien que majoritairement utilisées pour concevoir des SIW, les approches de conception d’hypermédias n’abordent pas de façon systématique la dimension fonctionnelle du système.

En plus de la dimension hypermédia qui distingue les SIW des SI traditionnels, une autre différence résulte de l’ouverture sur le monde offerte par le Web. En effet, le public cible d’un SIW est potentiellement plus large que celui d’un SI traditionnel, même s’il est toutefois possible de restreindre l’accès d’un SIW à une population spécifique en ayant recours à des politiques de sécurité et/ou en s’appuyant sur une infrastructure réseau privée (intranet). Dans le cas d’un SIW ouvert, la population d’utilisateurs est plus importante et souvent plus diversifiée. L’hétérogénéité des utilisateurs se constate à plusieurs niveaux relatifs par exemple à leurs buts, leurs connaissances, leurs compétences, leurs préférences en terme de présentation ou encore aux conditions matérielles dans lesquelles ils accèdent au SIW Il est donc nécessaire que les SIW s’adaptent à leurs utilisateurs.

Doter les SIW de capacités d’adaptation vise ici l’objectif suivant : *faire en sorte que l’utilisateur ait l’impression que le système a été développé spécialement pour lui*. Comme le souligne Colette Rolland, il est préjudiciable d’abandonner à la phase de développement des décisions qui, si l’on veut tendre vers un système de qualité, doivent être abordées dès la conception [Roll00]. En conséquence, des modèles et une démarche pour l’adaptation constituent, selon nous, la troisième dimension de la conception de SIW qui vient s’ajouter aux deux axes mentionnés précédemment (i.e. organisation de l’information et aspects fonctionnels).

Le contexte de cette thèse est donc celui des SIW adaptables à l’utilisateur, depuis leur conception jusqu’à leur mise en oeuvre.

Problématique

En tant que système basé sur le Web, le SIW propose une interface utilisateur reposant sur des caractéristiques hypermédia. De par la nature non-séquentielle de l’accès à l’information et aux fonctionnalités induites par ces caractéristiques, l’utilisateur bénéficie d’une certaine flexibilité dans l’utilisation du SIW. Néanmoins, cette facilité de naviguer dans la structure hypermédia comporte deux risques majeurs pour l’utilisateur : le syndrome de désorientation (“*lost in hyperspace syndrome*”) et le phénomène de surcharge cognitive [Conk87].

La *désorientation* de l’utilisateur est une conséquence de la structure hypermédia d’un SIW. Un hypermédia tente d’émuler la pensée humaine en associant des nœuds d’information les uns aux autres, *via* des liens reproduisant les mécanismes d’associations d’idées. La difficulté de simuler *a priori* de tels mécanismes est une des raisons qui peuvent conduire au développement d’un hypermédia « spaghetti »¹, caractérisé par une structure de nœuds et de liens dans laquelle il est difficile de se repérer et de trouver l’information recherchée. La structure est, par ailleurs,

¹ Par analogie au code spaghetti en programmation.

d'autant plus complexe qu'il n'existe pas une seule porte d'entrée pour un hypermédia, mais une multitude de points d'accès. En conséquence, la navigation dans un tel espace se solde parfois par des digressions dans le suivi des liens qui perdent l'utilisateur. Il peut ainsi être entraîné hors du champ de ses intérêts initiaux. A cela s'ajoute le second risque inhérent aux structures hypermédiées. La *surcharge cognitive* est définie comme l'effort d'attention et de concentration que l'utilisateur doit nécessairement produire pour explorer l'espace d'information mis à sa disposition. Plus la masse informationnelle proposée à l'utilisateur est conséquente, plus la surcharge cognitive est marquée. Et, avec la facilité d'échanges d'informations que permet le Web, la quantité d'informations disponibles est, en effet, croissante.

Il est d'autant plus important d'éliminer, sinon de réduire ces risques, que le SIW offre des services pour la réalisation de tâches qui peuvent mettre en jeu la pérennité de l'organisation. Nous pensons ici au rôle central du SI dans la théorie des systèmes [LeMo77]. Même basé sur le Web, le SI constitue le lien indispensable entre d'une part, le système opérationnel (*i.e.* l'ensemble des moyens chargés de la production qui répond à la finalité de l'organisation), et d'autre part, le système de décision (qui dirige l'organisation et prend les décisions permettant d'atteindre les objectifs choisis). Les conséquences d'une désorientation ou d'une surcharge cognitive subie par un acteur de l'organisation peuvent être encore plus dommageables que dans le cas d'un utilisateur *lambda* navigant sur le Web à titre personnel.

Un objectif de l'adaptation dans les applications Web est de maîtriser la complexité inhérente à la structure hypermédia qu'elles mettent en œuvre. La technique d'adaptation la plus courante consiste à réduire la taille de l'application en termes de nombre de pages et de liens. Cette forme d'adaptation revient essentiellement à cacher ce qui n'est pas pertinent pour l'utilisateur. Ceci contribue effectivement à limiter les risques d'égaré de l'utilisateur dans l'hyperespace et de surcharge cognitive, l'espace offert à l'utilisateur étant plus réduit par rapport à celui que décrit l'application entière. Néanmoins, cette solution n'est, selon nous, pas entièrement satisfaisante. Même limité à ce qui est pertinent pour l'utilisateur, cet espace peut être vaste. De plus, à l'intérieur d'un espace donné, tout n'est pas forcément de pertinence égale. Nous posons donc l'hypothèse suivante, sur laquelle repose notre proposition : l'utilisateur n'a pas *tout le temps* besoin d'accéder à *toute* l'information et à *toutes* les fonctionnalités mises à sa disposition par le SIW.

Aperçu de la proposition

Dans le cadre de cette thèse, nous visons l'objectif suivant : tenter de résoudre, à travers une approche de conception intégrant la spécification d'adaptations multiples à l'utilisateur, les problèmes de perte dans l'hyperespace et de surcharge cognitive inhérents à la dimension hypermédia des SIW. Destinée à la conception d'applications Web qui sont avant tout des SI, cette approche traite, en outre, la dimension fonctionnelle qui n'est pas toujours considérée par les approches de conception d'hypermédias.

Partant de l'hypothèse posée précédemment, une réflexion sur la façon d'organiser plus efficacement les informations et les fonctionnalités du SIW en vue de leur utilisation est nécessaire. Dans ce but, nous proposons une approche permettant de définir les informations et les fonctionnalités auxquelles l'utilisateur a accès de façon immédiate (*i.e.* les plus pertinentes pour lui), tout en lui laissant la possibilité de contrôler la mise à disposition progressive d'informations et de fonctionnalités supplémentaires jugées par lui moins pertinentes. Ces principes reposent sur la notion d'*Accès Progressif* dont la définition, la modélisation et la mise en œuvre forment la contribution majeure de ce travail.

L'Accès Progressif est supporté par un modèle, appelé *Modèle d'Accès Progressif* (MAP) qui en décrit les concepts et les mécanismes à un niveau générique (*i.e.* indépendamment de ce à quoi va être appliqué l'Accès Progressif). Le MAP définit le cadre de référence dans lequel s'inscrit notre approche pour la conception de SIW. Le processus de conception d'un SIW est généralement caractérisé par la description du système selon différentes dimensions relatives aux données du domaine d'application, à l'hypermédia qui constitue la face visible du SIW, aux fonctionnalités offertes par le SIW, et aux utilisateurs.

Les modèles que nous proposons supportent ces différentes dimensions. Nous rejoignons ainsi la plupart des approches de conception d'applications Web qui s'accordent sur la nécessité de modéliser, au moins, ces quatre dimensions. Cependant, nos modèles s'en distinguent par le fait qu'ils sont tous axés sur l'Accès Progressif. Nous proposons ainsi une démarche de conception qui intègre les spécificités de l'Accès Progressif dans chacun d'eux :

- Le *Modèle du Domaine* peut faire l'objet de multiples organisations qui autorisent un Accès Progressif aux données du domaine d'application sous-jacent au SIW qu'il décrit ;
- Le *Modèle des Fonctionnalités* décrit notamment comment l'Accès Progressif est intégré, dès la phase de conception, dans la spécification des fonctionnalités proposées par le SIW (*i.e.* consultation, modification, ajout d'informations, etc.), pour exploiter les données du Modèle du Domaine. Les fonctionnalités les plus pertinentes sont accessibles immédiatement, d'autres de façon progressive, en fonction des besoins de l'utilisateur.
- Le *Modèle de l'Hypermédia* permet de construire un espace d'informations et de fonctionnalités reposant sur une structure et des mécanismes spécifiques de l'Accès Progressif.
- Le *Modèle des Utilisateurs*, à travers ses connexions avec les autres modèles, garantit la conception d'un SIW adapté à ses utilisateurs selon les différentes dimensions considérées. En particulier, sa connexion avec le Modèle des Fonctionnalités assure un Accès Progressif personnalisé aux fonctionnalités, ainsi qu'au contenu du Modèle du Domaine. Celles qu'il entretient avec le Modèle de l'Hypermédia garantissent à la fois une structure, des possibilités de navigation et une apparence des pages également personnalisées.

Afin de guider le concepteur, étape par étape, dans la définition des modèles mentionnés ci-dessus, nous proposons une démarche pour la conception de SIW à Accès Progressif.

Enfin, un environnement, appelé KIWIS (acronyme de *Knowledge for Improving Web Information Systems*), a été développé. Cet environnement intègre et valide notre approche établissant un Accès Progressif à l'information dans les SIW. D'une part, il supporte la démarche de conception et la génération des SIW. D'autre part, il constitue l'environnement d'utilisation de tels SIW.

Dans KIWIS, les capacités d'adaptation mises en œuvre dans un SIW à accès progressif s'expriment à travers la génération de systèmes qui ont la faculté de s'adapter de façon dynamique à leurs utilisateurs. En effet, les pages de l'hypermédia du SIW sont générées par KIWIS à la demande de l'utilisateur, en fonction de sa représentation dans le système. Notamment les demandes relatives à l'accès progressif, qu'il s'agisse de *dévoiler* l'hyperespace (*i.e.* voir plus d'informations, plus de fonctionnalités) ou inversement de le *masquer* (*i.e.* voir moins d'informations, moins de fonctionnalités) sont traitées en tenant compte du profil de l'utilisateur.

Organisation du document

Ce document est organisé en trois parties présentant respectivement, l'état de l'art, notre proposition et enfin la réalisation effectuée.

L'état de l'art débute par une étude de l'adaptation dans les SIW présentée dans le chapitre 2. Nous décrivons les différentes dimensions pouvant faire l'objet d'adaptations dans ces systèmes, en montrant que celles-ci résultent de la double composante SI et hypermédia des SIW. Nous identifions les différents processus permettant la mise en œuvre de l'adaptation. Nous montrons la place prépondérante d'une modélisation de l'utilisateur en vue de l'adaptation. Enfin, nous donnons quelques exemples de domaines dans lesquels l'adaptation est particulièrement présente.

Dans le chapitre 3, nous poursuivons l'état de l'art avec la présentation des principales méthodes de conception d'applications Web. Les approches décrites, majoritairement issues de la communauté hypermédia, ont souvent donné lieu à des évolutions qui permettent la prise en compte d'une infrastructure basée sur le Web et, parfois de caractéristiques fonctionnelles.

Le chapitre 4 présente une synthèse de l'état de l'art dont l'objectif est de mettre en relation les études menées dans les deux chapitres précédents. Nous montrons ainsi comment les méthodes décrites lors du chapitre 3 prennent en compte les différents aspects de l'adaptation mis en évidence par le chapitre 2.

Nous exposons ensuite notre proposition. Nous débutons par la présentation du Modèle d'Accès Progressif auquel le chapitre 5 est entièrement consacré. Nous décrivons tout d'abord les notions sur lesquelles il repose, avant d'en donner une représentation au moyen de notations UML étendues [Booc99][Mull99][OMG01][Rumb99]. Le protocole d'exploitation de ce modèle est ensuite décrit. Il s'agit d'un ensemble d'opérations permettant d'exploiter les différents concepts de l'Accès Progressif. Un exemple d'application des principes de l'Accès Progressif à un modèle de données exprimé en UML est donné.

La description des quatre modèles de conception que nous avons mentionnés dans cette introduction constitue le cœur du chapitre 6. Nous décrivons ainsi, le Modèle du Domaine, le Modèle des Fonctionnalités, le Modèle de l'Hypermédia et le Modèle des Utilisateurs.

Le chapitre 7 aborde la conception de SIW à Accès Progressif. Dans un premier temps, nous explicitons les relations qu'entretiennent entre eux les modèles décrits au chapitre 6. De ces relations, nous déduisons une démarche de conception. Nous mettons ensuite en évidence le rôle sous-jacent du MAP dans la conception des SIW. Enfin, nous comparons notre proposition avec les approches étudiées au chapitre 3 en nous appuyant sur la synthèse de l'état de l'art présentée au chapitre 4.

Le chapitre 8 constitue la troisième partie de ce document, et traite de la réalisation effectuée. Nous présentons le prototype KIWIS, implémenté pour supporter l'approche par Accès Progressif, en décrivant tout d'abord son architecture. L'environnement de conception est ensuite présenté pour illustrer les étapes de la démarche suivie par le concepteur lors de la création d'un SIW. L'environnement d'exécution que comprend également KIWIS est mis en évidence. Nous décrivons un SIW à Accès Progressif tel qu'il est mis à la disposition d'un utilisateur et utilisé par celui-ci.

Enfin, le chapitre 9 conclut ce mémoire. Il dresse le bilan et donne les perspectives de ce travail autour de l'Accès Progressif et l'adaptabilité dans les Systèmes d'Information basés sur le Web.

ÉTAT DE L'ART

2. L'ADAPTATION DANS LES SIW

L'objectif de ce chapitre est de présenter un état de l'art sur la notion d'adaptation dans les SIW. Dans ce but, une première section est consacrée à la description des évolutions qui, à partir des hypertextes, ont ouvert la voie aux SIW. Nous mettons en évidence les caractéristiques des SIW qui sont issues d'une part de l'approche hypermédia, et, d'autre part, du domaine des Systèmes d'Information. Cette mise en relief permet de mieux comprendre ce sur quoi porte l'adaptation dans ces systèmes.

La deuxième section est consacrée à la présentation des différentes dimensions qui dans les SIW peuvent faire l'objet d'adaptations. Nous montrons que celles-ci portent sur le contenu et les fonctionnalités mise à disposition par le système d'information, mais également sur des dimensions héritées de l'approche hypermédia telles que la structure de l'hypermédia, les modes de navigation ou encore l'apparence des pages.

Dans la troisième section, la notion d'*adaptation* d'un système, en tant que processus, est précisée. A partir des différents termes employés dans la littérature autour de l'adaptation, nous identifions plusieurs processus d'adaptation. Ils diffèrent principalement dans la façon dont ils mettent en œuvre les adaptations : le processus est plus ou moins dynamique, et laisse une plus ou moins grande place au contrôle à l'utilisateur.

Dans la quatrième section, nous montrons comment l'émergence de systèmes capables d'adaptations s'accompagne d'une modélisation de plus en plus fine de l'utilisateur. Une classification des différentes approches pour la représentation des utilisateurs est proposée. Les techniques utilisées pour représenter les utilisateurs sont ensuite présentées.

Nous proposons ensuite, avant de conclure, quelques exemples de modélisation d'utilisateurs et d'adaptations pratiquées dans différents domaines : recherche d'information, commerce électronique et éducation.

2.1. Systèmes d'Information basés sur le Web

2.1.1 *L'influence des hypertextes et hypermédi*

Les termes *hypertexte* et *hypermédia* désignent des approches pour la gestion de l'information qui partagent le fait d'être basées sur un réseau de nœuds reliés entre eux par des

liens. Lorsque l'information portée par les nœuds est uniquement textuelle, on parle d'hypertexte ; l'hypermédia correspond à un réseau de nœuds véhiculant des informations de différents types, comme du texte, des images, des données audio ou vidéo, éventuellement combinées. Dans ce mémoire, nous emploierons le terme *hypermédia* pour couvrir les deux concepts.

Dès 1945, Vannevar Bush établit les principes d'un hypermédia dans un article intitulé "*As we may think*" [Bush45]. Il met en évidence l'inadéquation des systèmes informatiques par rapport au fonctionnement du cerveau humain. Dans ces systèmes, la rigidité et la linéarité des techniques de stockage et de récupération des informations s'opposent à la façon dont pense l'être humain. Bush évoque notamment la tendance de l'individu à raisonner par associations d'idées et la complexité des chemins empruntés alors par la pensée, ce que n'autorisent pas les systèmes de l'époque. Conscient de la difficulté à reproduire artificiellement et avec exactitude un tel processus mental, Bush préconise toutefois de s'en inspirer et expose quelques unes de ses idées à travers la description d'un système qu'il nomme « *memex* ». Le terme *hypertexte* est inventé en 1965 par Ted Nelson [Nels65] dont les travaux poursuivent ceux de Bush. Selon lui, une telle approche ressemble, plus que toute autre technologie, à la pensée humaine dans le sens où elle permet d'explorer un espace d'informations selon de multiples axes. Ces différents chemins d'exploration permettent à chaque individu d'emprunter celui qui lui convient le mieux dans sa quête d'informations (i.e. qui le chemin qui correspond le plus à son processus mental de réflexion).

Bien que ses origines remontent à plus de 50 ans, l'approche hypermédia a en réalité tardé à s'imposer. Selon [Lang01a], deux succès technologiques ont joué un rôle de facteur déclenchant dans le développement de l'approche.

D'une part, en 1987, Apple lance l'outil auteur *HyperCard* et le distribue largement en le fournissant pendant plusieurs années à l'achat d'un Macintosh. *HyperCard* est un "organisateur d'informations" qui s'inspire de l'approche hypermédia. L'utilisateur divise ses informations en plusieurs unités appelées *cartes*, et établit des liaisons entre elles au moyen de scripts écrits en *HyperTalk*, un langage de programmation intégré. Les cartes peuvent contenir du texte, des sons, des images fixes ou animées (avec l'apparition de la technologie *QuickTime*). Le succès d'*HyperCard* (un million d'exemplaires vendus lors de la première année de commercialisation) contribue à promouvoir l'approche hypermédia auprès des utilisateurs d'outils informatiques qui disposent alors d'un logiciel pour créer leurs propres applications.

D'autre part, Lang [Lang01a] considère la création du Web comme le second facteur technologique expliquant l'essor de l'approche hypermédia. Bien que tout hypermédia ne soit pas forcément basé sur le Web, c'est à travers ceux mis en ligne que de nombreux utilisateurs expérimentent pour la première fois le concept d'hypermédia [Bieb97]. Le Web a ainsi été l'occasion de rendre populaire et universel le concept d'hypermédia et d'en démontrer toute la puissance.

L'influence de ce concept est telle que les applications développées en vue d'une mise à disposition *via* le réseau en exploitent très largement les principes. Dans la section suivante, nous précisons ce que désigne l'expression '*applications basées sur le Web*' (ou '*applications Web*') à partir de l'étude des différents points de vue exprimés dans la littérature. Cette expression couvrant un spectre relativement large, quelques auteurs ont proposé de classer les différents types d'application. Nous présentons et commentons ces propositions.

2.1.2 Typologie des Applications basées sur le Web

Conallen [Cona00] propose une définition des applications basées sur le Web qui, d'une part, les positionne par rapport à une infrastructure technologique, et d'autre part, les distingue des *sites Web*. Il explique que « *les applications Web résultent de l'utilisation de sites et de systèmes Web* » ([Cona00], p13). Un *système Web* fait référence à l'infrastructure permettant un échange d'informations entre ordinateurs. Les éléments essentiels² d'un système Web sont donc :

- le réseau, qui relie physiquement les ordinateurs,
- un poste client doté d'un navigateur, *i.e.* un logiciel permettant d'afficher localement les documents hébergés sur d'autres ordinateurs du réseau,
- un poste hôte sur lequel un service (ou démon), appelé *serveur Web*, permet de localiser et de retourner les informations demandées par le client au moyen de requêtes.

Les informations sont organisées dans des pages reliées par des liens constituant ainsi un hypertexte. Un tel hypertexte correspond à un *site Web* qui peut être consulté au moyen d'un système Web.

Une application Web exploite également un système Web. Cependant, elle apporte une dimension supplémentaire à cette infrastructure à travers la mise en œuvre d'un serveur d'application. Ce dernier rend possible le traitement d'actions de l'utilisateur qui vont au-delà de simples requêtes de navigation. Ainsi, pour Baresi & al [Bare00], les sites Web présentent deux dimensions essentielles, celle des informations et celle de la navigation, alors que les applications Web, en revanche, en intègrent une troisième : celle des opérations. Cette position est partagée par Conallen [Cona00], pour qui, une application Web se différencie d'un site Web en cela que l'application offre à l'utilisateur des mécanismes lui permettant de réaliser des tâches plus complexes. Par ailleurs, le résultat de ces actions se traduit par une modification de l'état applicatif sur le serveur : des données sont ajoutées, modifiées, supprimées, des actions deviennent disponibles ou ne le sont plus, etc.

Au regard de cette définition, il apparaît clairement que de nombreuses applications peuvent être qualifiées de *basées sur le Web*. Ainsi, dès lors qu'une interface permet de saisir des informations (par exemple pour s'enregistrer en tant qu'utilisateur), le serveur d'application voit son état changer puisque les données fournies sont stockées et puis prises en compte dans la logique applicative. Comme nous le verrons par la suite (cf. section 2.1.3), des applications Web proposent toutefois des actions bien plus avancées : c'est notamment le cas de celles qui mettent en œuvre les *fonctionnalités métier* spécifiques d'une organisation.

Dans [Mecc99], la classification proposée peut être rapprochée de celle de Conallen, bien qu'elle distingue différentes catégories de ce que les auteurs appellent *sites Web*. Leur proposition est basée sur une analyse de la complexité des sites en regard de deux critères : l'un est relatif aux données gérées, et l'autre relève de ce que les auteurs nomment l'*application* et qui correspond aux services offerts par le site. Nous retrouvons donc, en référence à [Cona00], la notion de *tâche* qui caractérise une application Web. Le croisement de ces deux critères – complexité des données et complexité des services – permet à [Mecc99] de distinguer quatre catégories :

² Bien que non décrites ici, certaines technologies sous-jacentes aux éléments présentés sont indispensables au fonctionnement d'une telle infrastructure. On citera par exemple les protocoles TCP/IP (Transmission Control Protocol / Internet Protocol), HTTP (*HyperText Transfert Protocol*), les langages de publication de pages Web HTML (HyperText Markup Language), XML (eXtended Markup Language), etc.

- les *sites de présence sur le Web* sont caractérisés par un faible degré de complexité tant en terme de données que de services. Leur objectif concerne principalement la mise en ligne d'informations, à des fins publicitaires par exemple, ou résultant d'une volonté de diffuser des informations diverses (activités d'une association, présentations de manifestations, informations personnelles, etc.). Généralement constitués d'un petit nombre de pages, ces sites sont principalement construits à l'aide d'éditeurs HTML et ne s'appuient quasiment pas sur des techniques de développement *ad-hoc*.
- les *sites "catalogues"* ou à *forte intensité de données (data-intensive Web sites)* sont caractérisés par le fait qu'ils publient une masse importante de données, selon une structure hypertexte complexe. Ils n'offrent en revanche pas ou peu de services.

Nous assimilons ces deux catégories à la notion de sites Web telle qu'elle est envisagée par Conallen. Les deux suivantes peuvent être comparées à la notion d'applications Web :

- les *sites orientés services* ont vocation à fournir des services spécifiques tels que ceux offerts par les moteurs de recherche³ ou les applications de messagerie électronique. De tels sites peuvent reposer sur des bases de données de taille conséquente, mais la structure des données et de l'hypertexte reste simple. La complexité est donc davantage liée aux applications sous-jacentes qui garantissent les services, et c'est en général autour de ces aspects que s'oriente le développement de tels sites.
- les *Systèmes d'Information basés sur le Web (SIW)* constituent la quatrième catégorie de sites Web. Ils combinent la mise à disposition et la gestion de données complexes avec des services interactifs sophistiqués. Les applications de commerce électronique et les SI des organisations entrent dans cette catégorie.

La Figure 2.1 illustre le positionnement de ces catégories selon deux axes relatifs à la complexité des données (axe vertical) et à celle des services (axe horizontal). Inspirée de [Mecc99], cette représentation fait également apparaître la distinction établie dans [Cona00] entre sites et applications Web, où seule la dimension applicative intervient.

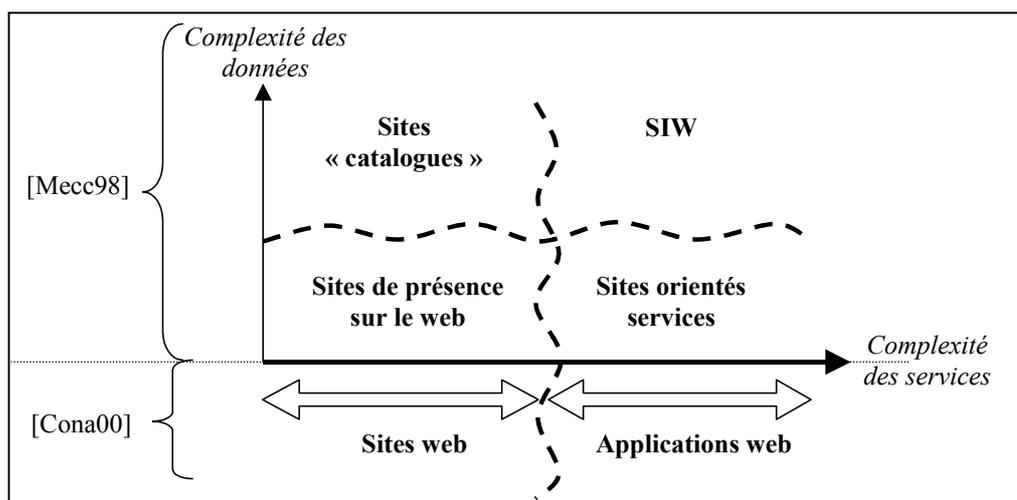


Figure 2.1 – Classification des sites et applications basées sur le Web
D'après [Mecc99] et [Cona00].

³ Notons que, pour Conallen, ([Cona00], p 23) les moteurs de recherche ne sont pas considérés comme des applications Web dans la mesure où ils ne sont généralement pas caractérisés par une modification de l'état applicatif suite aux actions (requêtes) de l'utilisateur.

La complexité des services offerts est donc un dénominateur commun aux propositions que nous avons présentées. Cette notion est reprise dans la section suivante qui vise à positionner les systèmes d'informations basés sur le Web par rapport aux applications Web et aux Systèmes d'Information non basés sur le Web, que nous qualifions de *Systèmes d'Information traditionnels* (désignés par SI dans la suite).

2.1.3 *SI traditionnels versus SIW*

Au regard de la section précédente, un SIW est à considérer comme une application Web, qui repose sur un système Web et qui met en œuvre des services relativement complexes. Ces services ont ceci de particulier qu'ils doivent répondre aux besoins spécifiques de l'organisation. Appelés *fonctionnalités, logique métier* [Ross01] ou encore *travail* [Isak98a][Denn98], ces services sont du même ordre que ceux offerts par des SI traditionnels. Les SIW rejoignent ainsi les SI dans la nécessité de spécifier, conformément au fonctionnement de l'organisation, les aspects fonctionnels de l'application [Frat99]. De façon triviale, nous pouvons alors définir un SIW comme un SI reposant sur un système Web. Toutefois, pour Dennis, il convient de noter qu'un SIW est *d'abord* un Système d'Information (traditionnel), et *ensuite* un système basé sur le Web [Denn98].

Se pose alors la question de déterminer l'impact de la technologie Web dans les SIW. En d'autres termes, baser un SI sur cette technologie donne-t-il lieu ou non à une nouvelle catégorie d'applications, différentes des SI traditionnel ? Un premier élément de réponse peut être apporté en se référant à la définition d'un SI proposée dans [Roll88] (*cf.* aussi [Boda83]). Colette Rolland définit un SI comme « *un ensemble formé :*

- *de collection de données, représentation partielles, en partie arbitraires mais nécessairement opératoires, d'aspects pertinents de la réalité de l'organisation sur lesquels on souhaite être renseigné. Ces collections inter-reliées, aussi cohérentes que possible, sont mémorisées et communiquées dans le lieu, le moment et la présentation appropriées aux acteurs qui en ont l'usage ;*
- *de collections de règles qui fixent le fonctionnement informationnel. Ces règles traduisent ou sont calquées sur le fonctionnement organisationnel. Parties intégrantes du SI, elles doivent être connues des acteurs qui utilisent le SI et leur sont nécessaires pour l'interprétation et la manipulation des collections de données ;*
- *d'un ensemble de procédés pour l'acquisition, la mémorisation, la transformation, la recherche, la communication et la restitution des renseignements ;*
- *d'un ensemble de ressources humaines et de moyens techniques intégrés dans un système, coopérant et contribuant à son fonctionnement et à la poursuite des objectifs qui lui sont assignés. »*

Cette définition, bien qu'antérieure à l'apparition des SIW, peut leur être appliquée, dès lors que l'on considère la dimension Web comme implicitement mentionnée dans le quatrième paragraphe, à travers la référence aux moyens techniques. Néanmoins, nous pensons qu'il est réducteur de restreindre le rôle de la dimension Web à un choix technologique d'implémentation. Deux facteurs au moins vont dans ce sens.

Les SIW partagent avec les hypermédias (et donc le Web) une organisation non-linéaire des contenus informationnels et la spécification de chemins multiples de navigation reliant ces contenus. D'ailleurs, ces notions constituent pour Fraternali des éléments déterminants de l'approche SIW par rapport à une approche SI traditionnelle [Frat99]. Par extension, l'organisation de l'information en vue de sa *présentation* et en tant que support à la *navigation*, ne peut être ignorée dans le développement de tels SIW. Dans la section 2.2, nous verrons que

ces dimensions interviennent de façon notable dans les méthodes proposées pour la conception d'applications et de SIW.

Une autre caractéristique du Web est son ouverture sur le monde. Le fait que le public cible d'un SIW soit potentiellement plus large que celui d'un SI traditionnel est pour Isakowitz [Isak95] un critère de différenciation des deux types de systèmes. Cette ouverture du SIW à une population d'utilisateurs plus grande entraîne une hétérogénéité que l'on peut constater à plusieurs niveaux. Les utilisateurs présentent ainsi des buts, des connaissances et des compétences divers, ont des préférences en terme de présentation variées, ou encore accèdent aux SIW dans des conditions matérielles différentes.

Déjà un enjeu dans les SI traditionnels, il devient incontournable de doter les SIW de *capacités d'adaptation à leurs utilisateurs* si l'on veut accroître le sentiment de satisfaction du plus grand nombre. L'adaptation aux utilisateurs dans un SIW s'exprime pour différentes dimensions qui résultent de la double filiation de ces systèmes, issus d'un côté, du courant hypermédia et de l'autre, du domaine des systèmes d'informations traditionnels. Les différentes dimensions de l'adaptation dans les SIW sont explorées dans la section suivante.

2.2. Les dimensions de l'adaptation

Dans cette section, nous présentons les différents axes qui peuvent faire l'objet d'adaptations dans un SIW. Cette présentation s'appuie principalement sur les travaux de la communauté hypermédia adaptatifs. Nous en proposons une synthèse dans la mesure où tous les auteurs ne considèrent pas exactement les mêmes dimensions.

Ainsi, dans [Brus97] [Brus98] et [Raad02] deux dimensions sont considérées : *adaptation du contenu* et *adaptation de la navigation*, alors que dans [Pate99][Koch00][Fras02] une troisième dimension est ajoutée: *adaptation de la présentation*. La terminologie employée par les différents auteurs prête parfois à confusion. Ainsi, pour [Brus97][Brus98] et [Raad02], l'adaptation du contenu est également appelée adaptation de la présentation. Cette expression ne fait cependant pas référence à l'adaptation de la présentation telle que considérée par les autres auteurs dont [Pate99][Koch00][Fras02]: il s'agit pour eux d'adapter les caractéristiques de mise en page (*i.e.* la forme ou apparence). Nous partageons cette seconde définition de l'adaptation de la présentation. Cette présentation est, par ailleurs, complétée par des propositions qui visent à considérer la dimension fonctionnelle des SIW ainsi que les adaptations basées sur la gestion des conditions matérielles d'exploitation.

Dans cette présentation, nous empruntons la terminologie proposée par [Brus97][Brus98] et reprise dans [Koch00]. Une *méthode* d'adaptation désigne ici une idée conceptuelle, une manière d'envisager une adaptation. Une *technique* d'adaptation correspond aux moyens mis en œuvre pour implémenter une méthode. Une méthode peut être implémentée grâce à différentes techniques, et, inversement, une technique peut être utilisée pour différentes méthodes.

2.2.1 Adaptation du contenu délivré

L'adaptation du contenu consiste principalement à proposer une information qui corresponde aux connaissances de l'utilisateur ou aux activités qu'il doit mener à bien. Le contenu peut être adapté de plusieurs façons. Parmi les plus courantes, on distingue les méthodes visant à :

- fournir un supplément d'information par rapport à un contenu de base défini pour l'ensemble des utilisateurs ;
- cacher une partie de l'information jugée non pertinente pour un utilisateur donné ; cette méthode peut être vue comme l'inverse de la précédente ;

- choisir, parmi plusieurs alternatives prédéfinies pour proposer une information, celle qui est la plus adéquate pour un utilisateur donné.

Plusieurs techniques existent pour mettre en œuvre de telles adaptations de contenu. Elles portent majoritairement sur des données de type texte, mais peuvent être transposées à des données multimédias [Koch00]. Des exemples de systèmes utilisant les techniques ci-dessous peuvent être trouvés dans [Brus97][Brus98]. Les termes en italique sont repris de l'auteur.

Proposer des informations supplémentaires ou, au contraire, en cacher, peut être réalisé par l'association de conditions aux éléments d'information (*conditional text*). Ces conditions expriment généralement le(s) critère(s) requis pour y accéder. Par comparaison avec les valeurs affectées à l'utilisateur pour ces critères, le système décide de montrer ou non l'information. Le même objectif est atteint par le recours à une technique basée sur un principe d'expansion/réduction du texte dans une page Web (*stretchtext*). Une partie de texte (en général un mot ou groupe de mots correspondant à un concept) est associée à une information additionnelle qu'il est possible de faire apparaître. Le système choisit de dévoiler ou non l'information en référence à des spécifications données par un modèle utilisateur ; parallèlement, l'utilisateur peut également activer les mécanismes d'expansion/réduction du texte.

La méthode basée sur le choix de contenus alternatifs est traduite de deux façons au niveau technique. La première consiste à créer autant de versions de pages Web que nécessaire (*page variants*). Cette technique, simple mais relativement coûteuse, n'est utilisée que dans les systèmes destinés à quelques groupes distincts d'utilisateurs. Une adaptation individuelle par ce biais est en effet difficilement envisageable. La seconde technique adopte un principe similaire mais à un niveau de granularité plus fin : différentes versions relatives à un concept sont créées (*fragment variants*). Une sélection de la version adéquate est opérée pour construire la page présentée à l'utilisateur.

Enfin, une technique est basée sur une organisation de l'information exploitant la notion de cadres (*frame-based technique*). Le principe est d'insérer dans de tels cadres, toujours en fonction de l'utilisateur, des représentations différentes de l'information à laquelle il vient d'accéder dans la fenêtre principale. Des informations supplémentaires reliées à celle-ci, ou liens vers ces informations, peuvent également être proposé(e)s. Le cadre peut être masqué à la demande de l'utilisateur.

Dans une approche hypermédia, l'adaptation du contenu se traduit essentiellement au niveau structurel par une modification des nœuds : ce sont en effet les pages qui délivrent l'information adaptée. La seconde dimension d'une structure hypermédia, celles des liens, peut également faire l'objet d'adaptation. Nous abordons ce point ci-dessous.

2.2.2 *Adaptation de la navigation*

L'adaptation de la navigation vise plusieurs objectifs qui, de façon générale, permettent d'éviter à l'utilisateur de suivre des chemins le menant à des informations non pertinentes [Brus97]. Nous décrivons les objectifs les plus courants et les méthodes associées d'après [Brus98]:

- le *guidage global* consiste à aider l'utilisateur à suivre le plus court chemin pour atteindre son « but⁴ » d'information. L'information est contenue dans un ou plusieurs nœuds de l'hyperespace et ne peut être accédée que par navigation. Deux méthodes sont envisagées.

⁴ Brusilovsky parle de but général d'information (*global information goal*). C'est à partir de ce but qu'est réalisé le processus d'adaptation.

L'une, appelée guidage direct, est de suggérer étape par étape (i.e. après chaque suivi de lien) le lien suivant à actionner. L'autre suppose de trier les liens potentiels par ordre décroissant de pertinence (*adaptive sorting*). La seconde méthode est équivalente à la première dès lors que l'utilisateur choisit le lien en tête de liste, mais lui laisse par ailleurs la liberté de suivre d'autres chemins.

- le *guidage local* vise à assister l'utilisateur dans le choix du lien à suivre à partir du nœud courant. Alors que le guidage global construit un chemin complet de navigation vers un but, le guidage local se résume à une suggestion élaborée à partir des préférences, connaissances, etc. de l'utilisateur. La méthode privilégiée à cet égard est également le tri des liens en fonction de leur pertinence, évaluée ici par rapport aux caractéristiques de l'utilisateur.
- l'*orientation globale* aide l'utilisateur à comprendre la structure complète de l'hyperespace et à situer sa position absolue actuelle dans cette structure. Des méthodes basées sur l'insertion de repères visuels ou d'affichage de la cartographie de l'hypermédia sont utilisées. Le concept de visites guidées, par la structuration en espaces plus petits qu'il suppose, participe également à l'orientation de l'utilisateur qui « appréhende » plus facilement l'espace dans lequel il navigue. Enfin, une méthode consiste à agrémenter les liens d'annotations qui introduisent une sémantique facilitant la compréhension de la structure. Les annotations de liens dans le domaine hypermédia sont principalement de nature visuelle [Koch00]. Un exemple classique est celui de la couleur utilisée pour un hyperlien textuel qui varie selon que le lien a été déjà visité ou non. Dans une démarche d'adaptation à l'utilisateur, la couleur peut être utilisée pour traduire la pertinence du lien pour un utilisateur donné.
- l'*orientation locale* porte sur l'introduction d'informations permettant à l'utilisateur de mieux percevoir sa position relative et surtout d'avoir une meilleure représentation de ce qu'il peut atteindre à partir de celle-ci. Les méthodes d'annotations de liens sont utilisées, ainsi que des méthodes qui cachent les informations non pertinentes (i.e. les liens vers des nœuds non pertinents).
- Les *vues personnalisées* ont comme objectif de réduire l'hyperespace à la partie que les utilisateurs exploitent réellement dans leur tâche quotidienne. Dans [Brus98], la création de telles vues est présentée comme relevant d'une méthode définissant la liste des liens vers les hyperdocuments qu'est amené à manipuler l'utilisateur. Il s'agit globalement de créer des structures de type *favoris* (ou *bookmarks*) qui permettent un accès rapide à un espace restreint, mais n'offrent pas un véritable support à la réalisation des activités. Ceci est, selon nous, envisageable en combinant les principes des vues personnalisées et d'un guidage global fondé sur un but traduisant le besoin de réaliser une tâche.

Les techniques utilisées pour implémenter les méthodes précédemment citées ont en commun le recours à des algorithmes d'évaluation et de tri des liens candidats. Elles exploitent également soit une représentation des buts de l'utilisateur, soit de ses caractéristiques, ou encore une combinaison des deux. Enfin, les techniques implémentent certains aspects spécifiques des méthodes tels que la définition des 'clés' visuelles utilisées dans les annotations de liens (spécification de la sémantique associée aux couleurs par exemple).

Le cas des adaptations par annotation de liens peut être considéré comme relevant d'une adaptation non pas de la navigation, mais de la présentation, puisqu'il repose sur des aspects visuels. Les objectifs de l'adaptation des présentations sont décrits ci-dessous, ainsi que les techniques employées à ces fins.

2.2.3 *Adaptation de la présentation*

La troisième dimension considérée comme objet d'adaptation dans les SIW est celle de la présentation. À ce niveau, ce sont les caractéristiques visuelles qui sont concernées : les adaptations portent donc sur la *forme* (ou *apparence*).

Les méthodes les plus courantes concernent l'adaptation des *caractéristiques graphiques* du document [Koch00][Fras02]. Il s'agit ici d'intervenir sur les choix de couleurs et formes des composants (qu'ils soient interactifs tels que les boutons de commandes, ou non tels que les bordures, les puces, etc.), de tailles, types et couleurs de police, d'arrière-plans, etc.

On rencontre également souvent dans la littérature des adaptations de la présentation à travers le *choix du langage* utilisé pour diffuser l'information [Koch00][Fras02].

Des méthodes portant sur l'adaptation de *l'organisation du document* sont aussi décrites. Celle-ci peut être spatiale ou temporelle. Dans ce cas, la page est décomposée en différentes régions qui peuvent être positionnées différemment dans la page et être activées selon des ordonnancements temporels variés. A un niveau de granularité plus fin, les caractéristiques spatiales et temporelles peuvent être affectées à des éléments d'affichage correspondant à des concepts du domaine d'application [Fras01].

Enfin, les *composants multimédias* d'une présentation peuvent également faire l'objet d'adaptation. Nous distinguons à ce propos des méthodes que nous appelons méthodes *d'altération de médias*, des méthodes *de substitution de médias*. Les méthodes d'altération conduisent à des adaptations qui modifient une information délivrée par un média sans changer de support : la réduction de la taille d'une image, le passage d'une image couleur à une image noir et blanc, le remplacement d'une vidéo par un résumé vidéo [Mart00], etc. en sont des exemples. Les méthodes de substitution de médias permettent de remplacer une information, supportée par un média donné, par une autre représentation de cette information en utilisant un autre type de média. Par exemple, la substitution d'une vidéo par une image extraite de celle-ci ou encore, le remplacement d'une bande audio par le texte transcrit de celle-ci illustrent ce second type de méthode.

Diverses techniques sont employées pour implémenter ces méthodes. Une première possibilité est de créer autant de versions de pages ou de régions de pages que nécessaire, à l'image de ce qui a été décrit pour les contenus alternatifs (*cf.* section 2.2.1).

Une technique moins rudimentaire est offerte à travers la possibilité de recourir à des feuilles de styles. Tout d'abord avec les CSS (Cascading Style Sheets [W3C98]) associées à HTML (HyperText Markup Language [W3C99b]), puis avec XSL (eXtensible Stylesheet Language [W3C01]) et les langages associés, proposés autour de XML (eXtensible Markup Language [W3C00]), la séparation du contenu et de sa présentation devient une procédure courante. Dès lors, la création de différentes feuilles de style – qui contiennent les spécifications des adaptations souhaitées – permet de produire différents documents bien que les feuilles de style soient appliquées à un même contenu.

Enfin, l'implémentation des méthodes d'adaptation de composants multimédias, peut faire appel à des programmes spécifiques de traitement d'images, de traitement des langues, de construction de résumés de vidéo, etc. Des techniques moins sophistiquées peuvent cependant être employées, comme la sélection d'une version d'information (i.e. version vidéo d'une information, sa représentation en images clés ou sous la forme d'une description textuelle du contenu) parmi un ensemble créé et stocké au préalable.

2.2.4 *Adaptation des fonctionnalités*

L'identification des besoins fonctionnels des utilisateurs et l'objectif d'apporter des réponses en adéquation avec ces besoins guident les démarches actuelles de développement de SI. Ainsi, une démarche s'appuyant sur UML et débutant par l'identification des cas d'utilisation, permet logiquement d'aboutir à un système que l'on peut qualifier d'adapté en regard de ce seul critère : le SI offre à l'utilisateur les moyens de remplir les tâches qui lui incombent. Néanmoins, le simple fait de rendre possible l'accomplissement d'une tâche ne suffit pas à rendre fonctionnellement adapté un SIW. Selon nous, l'adaptation des fonctionnalités dans un SIW comporte deux dimensions :

D'une part, elle suppose la définition d'un espace de travail adéquat qui, dans la lignée des objectifs visés par les hypermédias adaptatifs, évite à la fois une perte dans l'hyperespace et une surcharge cognitive. En d'autres termes, toutes les fonctionnalités supportées par le système n'ont pas à être mises à la disposition de l'utilisateur s'il n'en a pas l'usage. Les travaux relatés dans [Turk01][Turk02] sur les *hyperclasses* peuvent être analysés dans ce sens. Une hyperclasse est formée d'un ensemble de classes conceptuelles issues du schéma global d'une base de données (sous-jacente au SI). Il s'agit d'une unité ayant une sémantique précise qui est généralement associée à une fonction du SI. Une hyperclasse délimite le sous-domaine d'information (données et méthodes) nécessaire à la réalisation de la fonction associée. Un espace de travail déterminé à partir de la ou des hyperclasses qui ont un sens pour l'utilisateur permet d'atteindre un degré d'adaptation intéressant.

D'autre part, l'adaptation peut également se traduire par un suivi personnalisé des processus que doivent exécuter les utilisateurs. Les travaux de Gnaho [Gnah00], et notamment les aspects liés à la définition d'*unités sémantiques de navigation*, guident l'utilisateur vers son but. Après chaque étape de navigation, des choix relatifs à la poursuite de l'activité sont suggérés à l'utilisateur. Dans cette approche, il s'agit principalement de fournir des alternatives pour poursuivre l'activité engagée (*e.g.* dans le cas d'une recherche d'information, une recherche par mots-clés donnés par l'utilisateur ou par choix parmi plusieurs thèmes est proposée). Un de nos précédents travaux suggère un suivi des processus plus complet. Nous proposons un modèle pour le *contrôle* et la *coordination* des activités conduites dans les SIW [Vill02a]. Contrôle et coordination sont garantis par le recours à des opérateurs temporels (inspirés des propositions de Allen [Alle83]). Notre approche permet à tout instant d'informer l'utilisateur sur l'état des activités (activité terminée, en cours, à engager, etc.) et de le guider dans la réalisation des activités : celles à entreprendre sont automatiquement présentées à l'utilisateur, des mécanismes de rappel permettent de relancer un utilisateur ayant « oublié » une activité, etc. Proposée à l'origine dans le but d'assurer la cohérence des activités des différents acteurs, notre proposition peut évoluer vers une solution adaptée à chaque utilisateur. Ceci permettrait de fournir une aide personnalisée à la réalisation d'activités et augmenterait ainsi les capacités d'adaptation fonctionnelle des SIW.

2.2.5 *Gestion des conditions matérielles*

Plusieurs propositions tiennent également compte dans la représentation de l'utilisateur des conditions matérielles dans lesquelles il accède au système. Le dispositif utilisé constitue, en effet, le chaînon intermédiaire dans les interactions homme-machine.

Afin de considérer la diversité croissante des dispositifs d'accès exploités par les utilisateurs, il est nécessaire de proposer des solutions pour gérer les différences de capacités inhérentes aux caractéristiques techniques des machines utilisées. Les taille d'affichage, capacité de la mémoire, vitesse du réseau, etc., varient ainsi considérablement en fonction des conditions matérielles d'utilisation : PC, ordinateur portable, télévision interactive, téléphone WAP (*Wireless Access*

Protocol), PDA (*Personal Digital Assistant*), etc. La description des caractéristiques matérielles est utilisée dans le but de diffuser une information dans un format cohérent avec les capacités du dispositif de sortie.

Plusieurs travaux vont dans ce sens [Kapl01][Fras02][Cann02a][Cann02b]. Nous illustrons ce domaine par la proposition d'un format de données spécifiquement proposé par le W3C. Appelée CC/PP (*Composite Capability / Preference Profile*) [Klyn01], cette proposition est basée sur RDF (*Resource Description Framework* [W3C99a]) et les schémas associés [W3C02]. Elle offre un cadre générique pour la description des capacités des dispositifs d'accès. Ce qui est appelé *préférences des utilisateurs* dans CC/PP correspond en fait à la définition des caractéristiques particulières du dispositif utilisé par l'utilisateur. La Figure 2.2 illustre un profil CC/PP défini en XML. Un tel profil est décrit par trois composants qui permettent de caractériser respectivement la plate-forme matérielle (*TerminalHardware*), la plate-forme logicielle (*TerminalSoftware*) et une application telle qu'un navigateur (*TerminalBrowser*). Chacun de ces composants est lui-même décrit à l'aide d'attribut (*e.g. display, name, version, etc.*).

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#"
  <rdf:Description rdf:about="MyProfile">
    <ccpp:component>
      <rdf:Description rdf:about="TerminalHardware">
        <rdf:type rdf:resource="HardwarePlatform" />
        <display>320x200</display>
      </rdf:Description>
    </ccpp:component>
    <ccpp:component>
      <rdf:Description rdf:about="TerminalSoftware">
        <rdf:type rdf:resource="SoftwarePlatform" />
        <name>EPOC</name>
        <version>2.0</version>
        <vendor>Symbian</vendor>
      </rdf:Description>
    </ccpp:component>
    <ccpp:component>
      <rdf:Description rdf:about="TerminalBrowser">
        <rdf:type rdf:resource="BrowserUA" />
        <name>Mozilla</name>
        <version>5.0</version>
        <vendor>Symbian</vendor>
        <htmlVersionsSupported>
          <rdf:Bag>
            <rdf:li>3.0</rdf:li>
            <rdf:li>4.0</rdf:li>
          </rdf:Bag>
        </htmlVersionsSupported>
      </rdf:Description>
    </ccpp:component>
  </rdf:Description>
</rdf:RDF>

```

Figure 2.2 – Exemple d'un profil CC/PP décrit en XML

La définition du processus d'adaptation qui exploite une telle description sort du cadre de la proposition CC/PP. Il revient donc aux concepteurs et développeurs de définir et mettre en œuvre les modalités d'exploitation des profils CC/PP. Un exemple, donné dans la section 3.1.5, illustre l'utilisation de CC/PP en vue d'adapter la présentation d'un système basé sur le Web (projet Hera [Fras02]).

Nous avons décrit les différentes dimensions qui peuvent faire l'objet d'adaptations dans un SIW – contenu, navigation, présentation, fonctionnalités, conditions d'exploitation –. Nous abordons à présent la question de la mise en œuvre de ces adaptations. Différentes façons d'envisager le processus d'adaptation du système à l'utilisateur sont décrites dans la littérature. Nous en donnons une description dans la section suivante.

2.3. Les différents processus d'adaptation

La *personnalisation* et la *customisation* visent à répondre de façon adaptée aux besoins et caractéristiques uniques et particuliers de *chaque utilisateur* [Cing00]. Ces termes, issus du

domaine du commerce électronique, insistent donc sur une dimension individuelle de l'adaptation.

Le processus de personnalisation est généralement distingué de celui de *customisation*. La customisation est un processus contrôlé par l'utilisateur qui effectue un choix entre plusieurs options. Le choix des options guide complètement la réponse fournie par le système [Rose01]. L'utilisateur est donc activement engagé dans une interaction avec le système délivrant l'information : les messages qu'il envoie au système formulent explicitement ce qu'il attend. De plus, les options retenues par l'utilisateur restent identiques jusqu'à ce qu'il procède à de nouveaux choix.

Par opposition à la customisation, la personnalisation véhicule l'idée d'un processus guidé par le système lui-même. Le système gère de la connaissance relative à l'utilisateur (ses besoins, ses préférences, etc.) et exploite cette connaissance pour décider ce qui doit être présenté à l'utilisateur. Le processus de personnalisation est décrit par [Rose01] comme une technologie « cognitive » dans la mesure où il s'appuie sur un apprentissage, par la machine, de ce que veut l'utilisateur. Cet apprentissage est basé sur une observation de son comportement au cours des sessions. Le système a, par ailleurs, la capacité de s'adapter au cours du temps.

Les concepts véhiculés par les termes personnalisation et customisation sont également exploités depuis plusieurs années dans d'autres domaines, notamment en interface homme machine [Brow90][IFIP96], systèmes à buts éducatifs [Oppe94][Oppe97], hypermédias adaptatifs [Brus98]. Si, pour certains auteurs, personnalisation, customisation et adaptation sont synonymes [Moba00][Kapp00][Ross01], d'autres emploient une terminologie différente. Celle-ci oppose généralement les capacités d'*adaptabilité* et d'*adaptativité* (équivalent par lequel nous traduisons le terme anglais '*adaptivity*') des systèmes, qui sont respectivement qualifiés d'*adaptables* et d'*adaptatifs*. Les sens attribués à ces termes diffèrent selon les auteurs :

Pour [Step98a][Step98b], l'adaptabilité fait référence à un processus d'adaptation basé sur des connaissances (à propos de l'utilisateur, de l'environnement, etc.) disponibles ou acquises par le système *avant* que ne soient engagées les interactions utilisateur/système. Les adaptations sont donc réalisées lors de l'initialisation du système qui se présente dans une version adaptée à l'utilisateur. Les connaissances utilisées par le système sont, de plus, supposées rester inchangées au cours de la session d'utilisation. Inversement, l'adaptativité traduit une vision plus dynamique du processus d'adaptation. Les connaissances sont ici acquises ou modifiées par le système au cours des interactions, *via* des techniques de suivi de session. Le système procède à des adaptations *pendant* que l'utilisateur interagit avec lui. Cette vision est également partagée par [Fras02] et par Kappel & al. [Kapp00], mais ces derniers appellent l'adaptabilité '*adaptation statique*' et l'adaptativité '*adaptation dynamique*'.

Selon une autre approche, l'opposition des termes est basée sur le degré de contrôle que possède l'utilisateur dans le processus d'adaptation [Diet93][Oppe94][Kobs01a]. Nous citons un extrait de [Kobs01a] qui introduit les différents niveaux – initiation, proposition, sélection et exécution – auxquels peut s'exercer le contrôle de l'utilisateur, et définit sur cette base ce qui distingue l'adaptabilité de l'adaptativité :

« Systems where the user is in control of initiation, proposal, selection and production of adaptation (“in control” thereby meaning that the user can perform these functions, but can also opt to let the system perform some of them) are called adaptable. For instance, a user of a Website may see a need to introduce a shortcut for a Web page that is frequently visited by him or her but deeply buried in the site hierarchy (adaptation initiation). The user then “proposes” to himself or herself to introduce a new link on the lateral navigation bar of the site pages (assume this is possible) or to define a bookmark in the browser, selects the

shortcut link and performs the necessary steps to produce this adaptation. In contrast, systems that perform all steps autonomously are called adaptive. For instance, the AVANTI system [Fink98] automatically inserts such personalised shortcut links for pages that a user frequently visits. »

Pour les auteurs, un processus d'adaptation peut également relever de configurations qui combinent les caractéristiques de l'adaptabilité (i.e. contrôle émanant de l'utilisateur) et de l'adaptativité (i.e. contrôle émanant du système). La Figure 2.3 montre différentes possibilités de processus d'adaptation : l'adaptativité contrôlée par l'utilisateur et initiée par lui (respectivement *user-controlled adaptivity* et *user-initiated adaptivity* [Kobs01a]) sont des exemples de configurations mixtes. Les graphiques insérés dans la figure représentent l'implication de l'utilisateur et du système dans les différentes actions qui constituent le processus d'adaptation.

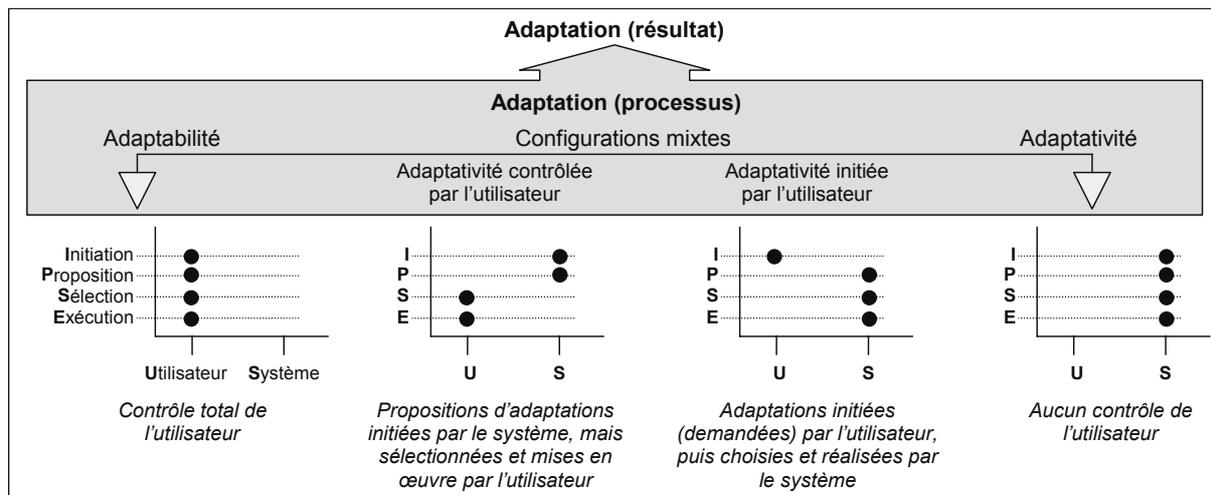


Figure 2.3 – Différents types de processus d'adaptation : de l'adaptabilité à l'adaptativité

L'adaptation du système est vue comme le résultat d'un processus pouvant relever de l'adaptabilité, de l'adaptativité, ou de processus qui mixent leurs caractéristiques. Les graphiques montrent qui, de l'utilisateur ou du système (axe horizontal), contrôle les actions (initiation, proposition, sélection, exécution – axe vertical) qui constituent les étapes du processus d'adaptation. Le contrôle est symbolisé par un cercle plein.

Les différents processus d'adaptation que nous avons décrits ont en commun, à des degrés divers, le fait d'exploiter une représentation de l'utilisateur comme source d'informations. La modélisation des utilisateurs est abordée dans la section suivante.

2.4. La modélisation des utilisateurs

La place de l'utilisateur dans une démarche d'analyse et de conception de SI est largement reconnue comme prépondérante. Le système est avant tout conçu pour des utilisateurs auxquels il est nécessaire de s'intéresser en vue d'anticiper leurs attentes à l'égard du système. Le domaine de l'ingénierie des besoins de façon générale et l'avènement d'UML avec, en particulier, le recours qu'il préconise aux cas d'utilisation, témoignent de l'intérêt particulier accordé à l'utilisateur. Ce dernier, bien qu'étant un interlocuteur privilégié du système, n'en est pas moins considéré comme un objet *extérieur* au système. En effet, l'utilisateur n'est généralement pas représenté comme une entité interne du système, comme le sont, par exemple, les concepts propres au domaine d'application décrits par le modèle de données.

Les problématiques de sécurité autour des systèmes, et notamment l'obligation de contrôler les accès aux ressources, ont rendu nécessaire la représentation interne des utilisateurs d'un système. Une telle représentation consiste essentiellement à stocker les informations utilisées pour décider d'une autorisation d'accès (identifiant, mot de passe, droits sur les données, etc.). Néanmoins, la mise en oeuvre de SI capables d'adaptations à l'utilisateur nécessite une description plus élaborée de l'utilisateur et de sa représentation en tant qu'objet à part entière du système⁵. Cette représentation interne de l'utilisateur vise à procurer au système les moyens de procéder aux adaptations souhaitées.

En amont de la représentation effective d'un utilisateur dans le système, et de l'exploitation de cette représentation, il est primordial de définir *comment* représenter l'utilisateur. En d'autres termes, il faut choisir les informations à retenir pour en construire un modèle. Plusieurs approches sont proposées pour guider l'élaboration de tels modèles et permettent de décrire une classification de ceux-ci.

2.4.1 Classification des modèles de l'utilisateur

Daniels [Dani86] oppose deux classes de modèle de l'utilisateur (ou modèle utilisateur). Les modèles *quantitatifs et empiriques* dont l'objectif est l'étude du comportement externe de l'utilisateur. Ces modèles se basent sur l'observation de ce que fait l'utilisateur et permettent d'étudier ses interactions avec le système. Les *modèles analytiques et cognitifs* s'intéressent à la modélisation du comportement interne de celui-ci. Ils favorisent l'identification des connaissances et processus cognitifs de l'utilisateur. La représentation de l'utilisateur doit, selon Daniels, tenir compte de ces deux catégories de modèle en raison de la complémentarité qu'elles offrent. Une telle approche permet en effet d'acquérir des informations relatives aux *savoir-faire* et *savoir* de l'utilisateur. Ces deux types d'information peuvent être exploités pour adapter le système.

Pour Rich [Rich99], la classification des modèles utilisateur est basée sur trois dimensions :

- un modèle *canonique* peut être opposé à une collection de modèles d'*utilisateurs individuels*. A des fins d'adaptation, le modèle canonique présente un intérêt modéré. Il consiste en effet à utiliser une représentation (des besoins, des caractéristiques, du fonctionnement cognitif, du mode de raisonnement, ou de tout autre trait, etc.), appelée *stéréotype*, que l'on considère représentative de l'ensemble des utilisateurs. Le recours à une approche individuelle nous semble plus approprié dans l'optique d'une adaptation à chaque utilisateur. Généralement, une approche intermédiaire consiste à définir différents modèles de groupes qui sont ensuite affinés individuellement [Koch00].
- les modèles basés sur des caractéristiques *persistantes* qui sont comparés aux modèles s'appuyant sur des caractéristiques *ponctuelles*. Ces dimensions permettent de distinguer les informations qui ont une validité à long terme, des informations valables ponctuellement, par exemple pour une session d'utilisation (les modèles en question sont respectivement désignés par *long-term* et *short-term models* dans [Koch00]).
- les modèles *explicites* qui sont confrontés aux modèles *implicites*. On différencie ainsi les modèles construits par l'utilisateur ou le concepteur, des modèles construits par le

⁵ Le terme 'système' désigne ici un système au sens large. La modélisation de l'utilisateur et la gestion de celle-ci peuvent en effet être déléguées par une application à un serveur uniquement dédié à ces activités (voir [Kobs01b] pour une revue des travaux autour des systèmes 'génériques' de modélisation de l'utilisateur (*i.e.* indépendants de toute application) et [Fink00] pour une comparaison des approches centralisées (*i.e.* la modélisation de l'utilisateur est embarquée dans l'application) et décentralisée (elle est déléguée à un serveur).

système sur la base du comportement de l'utilisateur. Les modèles implicites relèvent de la détection et de la gestion automatisées des *comportements externes* selon la terminologie de Daniels. On peut donc rapprocher ces modèles implicites des approches relevant de l'adaptativité telle que définie dans la section 2.1. Koch préfère d'ailleurs employer les termes de modèles *adaptatifs* (pour implicites) et *adaptables* (pour explicites) [Koch00].

La dernière classification possible que nous citons distingue les modèles des utilisateurs en fonction de la connaissance qu'ils véhiculent [Koch00]. Trois modèles sont ainsi généralement mis en évidence :

- les *modèles des connaissances du domaine* qui maintiennent la connaissance que l'utilisateur a du domaine ;
- les *modèles des connaissances indépendantes du domaine* qui concernent les connaissances générales ou non spécifiques au domaine, les centres d'intérêts, acquis (*background*) de l'utilisateur selon [Koch00]. Bien que non mentionnée par l'auteur, la connaissance de l'outil informatique nous paraît également entrer dans ce cadre.
- les *modèles psychologiques ou cognitifs* qui font référence aux préférences, capacités ou handicaps, et traits de personnalité de l'utilisateur. De tels modèles sont notamment utilisés dans le domaine éducatif pour représenter les styles d'apprentissage, types de raisonnement, capacités de concentration, etc., des utilisateurs).

Après avoir décrit différentes classifications de modèles utilisateur issues de la littérature, nous présentons dans la section suivante les techniques les plus couramment utilisées pour représenter de tels modèles.

2.4.2 Techniques de représentation des modèles utilisateur

Différentes techniques sont utilisées pour représenter les utilisateurs et structurer la connaissance à leur propos. Nous décrivons brièvement les approches les plus utilisées.

Dans les modèles de recouvrement (*overlay models*), les connaissances de l'utilisateur sont représentées par un sous-ensemble de connaissances 'idéales' qu'exploite le système. Ces connaissances idéales sont généralement exprimées par un modèle qui fait référence : celui-ci est préalablement établi par un expert du domaine, ou correspond plus simplement à l'ensemble des concepts manipulés par le système, ou encore définit ce que l'on attend de l'utilisateur (i.e. les connaissances qu'on souhaite le voir acquérir). Le modèle de l'utilisateur prend la forme d'un ensemble de paires concept-valeur, où le concept représente une connaissance élémentaire et la valeur associée indique le degré de maîtrise que l'utilisateur a de cette connaissance. Cette valeur est exprimée par un booléen, sur un domaine discret, ou encore à l'aide de probabilités qui expriment généralement une croyance du système à l'égard d'une connaissance.

Les profils utilisateurs sont des modèles également basés sur des couples (items, valeurs) mais ne font pas référence à un modèle de connaissances préétabli. Les items auxquels sont associées des valeurs (booléennes, discrètes, nominales, probabilités) permettent de représenter aussi bien des connaissances du domaine, que des préférences, des capacités, des actions précédentes, etc., de l'utilisateur.

De même, les modèles de stéréotypes sont fondés sur des paires items-valeurs. Ils introduisent toutefois une catégorisation inexistante dans les autres approches. Des combinaisons de couples items-valeurs définissent différents stéréotypes tels que *novice*, *avancé*, *expert*. L'utilisateur est donc associé à une des catégories élaborées et hérite de ses propriétés. L'utilisateur dispose, par suite, des adaptations réalisées pour le stéréotype. Ce manque d'individualisation de l'adaptation peut être contré par le recours à des approches mixtes qui

consistent à initialiser un modèle utilisateur à l'aide d'un stéréotype, puis à l'affiner en utilisant un modèle de recouvrement ou un profil. Un exemple est donné par les travaux de [Cann01a] dans lesquels l'association d'un utilisateur à un stéréotype est réévaluée, en fonction de ses actions, par le biais d'une méthode probabiliste.

L'exemple cité précédemment illustre un double enjeu de la représentation des connaissances à propos de l'utilisateur : gérer l'incertitude de cette représentation et supporter son évolution. Dans ce but, des travaux exploitant les réseaux de neurones tels que [Chen00] ou Bayésiens dans [Henz99] sont proposés.

L'acquisition et la gestion des modèles utilisateur reposent dans la plupart des travaux sur un module (ou composant) spécifique de modélisation et de gestion de l'utilisateur. Deux courants pour la mise en œuvre de ce module s'opposent (*cf.* [Fink00] pour une revue). D'un côté, une approche monolithique considère le composant dédié au modèle utilisateur comme un élément totalement intégré à l'application. De l'autre, des *serveurs* de modélisation de l'utilisateur, indépendants de toute application cible, sont proposés. Ces outils sont principalement rencontrés dans le domaine du commerce électronique (*e.g.* *Personalization Server* et produits complémentaires de l'*Art Technology Group*⁶).

Nous illustrons dans la section suivante comment sont représentés les utilisateurs à travers quelques exemples issus de différents domaines particulièrement concernés par l'adaptation.

2.5. L'adaptation dans différents domaines

Nous exposons quelques modes de représentation des utilisateurs, et façon d'utiliser celle-ci en nous intéressant à trois domaines : la recherche d'information, le commerce électronique et l'enseignement à distance.

2.5.1 Recherche d'Information

Dans les systèmes de recherche d'information basés sur le Web, les adaptations s'appuient principalement sur une représentation des centres d'intérêt de l'utilisateur [Bala95][Chan99].

Le profil d'intérêt est déterminé de façon plus ou moins complexe. Le profil utilisé dans Letizia [Lieb95] est constitué d'une liste de mots-clés qui représente les éléments pertinents pour un utilisateur. Dans [Boll99], le profil est constitué de multiples représentations de ce qu'est un « article intéressant » aux yeux d'un utilisateur (mots-clés, exemples d'URL, citations, etc.).

Dans Syskill & Webert [Pazz96], un ensemble de triplets constitués par un mot-clé, la probabilité que ce mot appartienne à un document pertinent, et la probabilité qu'il appartienne à un document non-pertinent sont déterminées pour chaque utilisateur. Cet ensemble constitue le profil et caractérise les documents pertinents et non-pertinents moyens pour un utilisateur.

Des travaux tels que ceux présentés dans [Pret99] proposent une représentation plus élaborée de l'intérêt de l'utilisateur qui s'appuie sur une hiérarchie de concepts construite à partir des pages visitées et du temps passé sur chacune d'elles. Certaines approches ajoutent une évaluation fournie explicitement par l'utilisateur auquel il est demandé de classer des pages visitées par ordre de préférence [Pazz96]. Jugeant cette façon de faire trop contraignante, Chan propose une technique d'estimation de l'intérêt d'une page pour un utilisateur basée sur l'observation de différents critères (*i.e.* rien n'est demandé à l'utilisateur) [Chan99]. Les critères retenus sont la fréquence à laquelle l'utilisateur accède à la page, le fait que cette page appartienne ou non aux

⁶ Art Technology Group, Dynamo Product Suite, <http://www.atg.com/products/highlights>

« favoris », la durée de la visite de cette page, la date de la dernière visite, le pourcentage de liens de cette page qu'a déjà visités l'utilisateur.

Le système Amalthea [Mouk97] combine jugement de l'utilisateur et détection automatique de ses centres d'intérêt. Il construit et fait évoluer un profil d'intérêt constitué de mots-clés pondérés aussi bien à partir des actions passées et en cours de l'utilisateur, des informations contenues dans les listes de sites préférés et ou d'historique, que d'un *feedback* explicite de la part de l'utilisateur.

Dans Profildoc [Lain99], la seule information qui concerne directement l'utilisateur est celle de son *niveau d'étude*. Les autres informations prises en compte sont le *champ disciplinaire* (e.g. Informatique, Agronomie, etc.), le *but de la recherche* (e.g. constitution d'une bibliographie, interprétation des données, etc.) et le *type de recherche* (i.e. pointue ou généraliste). Les exemples de valeurs mentionnées entre parenthèses constituent les choix que l'utilisateur peut faire lors de son entrée sur le système. L'exploitation de ce profil utilisateur est faite par un système d'aiguillage [BenA97] qui sélectionne les documents pertinents.

De façon générale, le profil défini en recherche d'information s'attache plus à caractériser le besoin d'information de l'utilisateur que l'utilisateur lui-même.

2.5.2 E-Business

Les objectifs essentiels des applications de commerce électronique sont évidemment l'accroissement de leur compétitivité et de leurs profits. Dans ce contexte, un utilisateur est avant tout un client. Dans [Fink00], quelques chiffres sont donnés pour illustrer les intérêts que trouvent les applications de commerce électronique à supporter l'adaptation. Une étude menée sur 25 sites dotés de telles capacités montrent que les sociétés ont vu leur nombre de nouveaux clients augmenter de 47% la première année, et leurs bénéfices s'accroître de 52%. Les adaptations offertes par de tels systèmes visent, comme dans tout autre domaine, à garantir la satisfaction de l'utilisateur, mais leur objectif essentiel est de fidéliser le client. Dans ce but, les applications de e-business cherchent :

- à épargner à l'utilisateur la re-saisie d'une information qu'il a déjà fournie au système.

Les techniques développées autour de la gestion du panier de l'utilisateur vont dans ce sens. Au cours d'une session, l'utilisateur navigue comme il le souhaite pour prendre connaissance des produits proposés, sélectionne/supprime des articles de son panier. Mais c'est surtout par la possibilité qui lui est offerte de passer commande en une seule fois (*one-click ordering*) de tout le contenu de son panier et à travers le maintien des informations le concernant en vue d'une commande ultérieure (nom, adresse, mode de paiement choisi, etc.) que ces systèmes facilitent la tâche de l'utilisateur. Le modèle de l'utilisateur combine ici des informations ponctuelles et persistantes [Rich99]. Le site gagne en efficacité car il est plus pratique d'utilisation, mais le bénéfice essentiel se situe dans la fidélisation du client qui hésitera à fréquenter un site concurrent où il devra procéder à nouveau à la saisie d'informations personnelles.

- à garantir que l'utilisateur ne passe pas à côté d'une information intéressante.

Dans ce but, l'envoi d'un mail contenant les informations susceptibles d'intéresser le client est de plus en plus répandu. Cette technique (*push email*) a été initialement proposée par le site d'offres d'emplois en ligne www.emailjob.com. Une autre façon de procéder consiste à insérer dans les pages du site une rubrique « nouvelle information pour vous » qui recense les informations non encore vues par l'utilisateur et potentiellement pertinentes par rapport à son profil. Cette option est, par exemple, mise en œuvre par la compagnie American Airlines sur son

site (www.americanairlines.com). Le modèle de l'utilisateur maintient donc dans ce cas un historique des pages ou informations accédées.

➤ à détecter les informations à même d'intéresser l'utilisateur.

Plusieurs services implémentés par les applications de commerce électronique permettent d'atteindre cet objectif [Fink00]. Une technique, aujourd'hui très souvent mise en œuvre, est due au libraire en ligne Amazon (www.amazon.com). Le *filtrage collaboratif* consiste à faire des recommandations à chaque utilisateur en se basant sur les achats effectués par des clients ayant un profil similaire. La gestion des affinités entre utilisateurs (*user-to-user affinities*) nécessite donc de maintenir l'historique des commandes des utilisateurs, mais aussi de pouvoir établir des comparaisons de profils. Une alternative à la comparaison de profils est la classification des utilisateurs à l'aide de stéréotypes.

De plus, le maintien de l'historique des achats dans un profil utilisateur permet ultérieurement de suggérer à l'utilisateur un nouvel achat (*action-to-item affinities*). Ainsi, lorsqu'un nouvel ouvrage sur un thème particulier est ajouté au catalogue, le système peut automatiquement le recommander aux utilisateurs ayant auparavant acquis des ouvrages sur le même thème.

Enfin, la mise en évidence d'affinités entre différents produits (*item-to-item affinities*) est également utilisée. Le principe est de recommander un produit ayant une relation (d'usage, de complémentarité, etc.) avec un produit qui semble intéresser le client. Par exemple, lorsque l'utilisateur ajoute un graveur de CD-ROM à son panier, le système lui suggère automatiquement l'achat de CD-ROM enregistrables. Notons toutefois que cette technique n'influe pas réellement sur le modèle utilisateur, les relations entre produits existant indépendamment des utilisateurs.

2.5.3 Enseignement à distance

Plusieurs travaux concernant la définition de métadonnées pour les matériels éducatifs sont en cours. Parmi eux, ceux de l'*IEEE Learning Technology Standardization Committee (LTSC)*⁷ ont porté sur une proposition, appelée *Learning Object Metadata (LOM)*, récemment élevée au rang de standard (le 13 juin 2002, par le *Standards Board of the IEEE Standards Association*) [LTSC02]. Il s'appuie sur des métadonnées de type Dublin Core [DCMI99] enrichies et affinées en vue de la description des données pour l'enseignement. Un des objectifs annoncés est la composition automatique et dynamique de cours personnalisés destinés à un apprenant donné. La prise en compte de cet apprenant est étudiée par différents groupes du LTSC.

Le *Learner Model Working Group*⁸ travaille à la définition d'un standard qui spécifie la syntaxe et la sémantique d'un modèle de l'apprenant, quels que soient son âge, ses acquis, sa situation géographique, ses moyens et sa situation scolaire. Ce travail consiste notamment à permettre la création et la construction d'un tel modèle, basé sur un standard national, exploitable tout au long du cursus scolaire des apprenants. La proposition du standard PAPI Learner (Public And Private Information for Learners) [LMWG00] décrit un sous-ensemble minimal d'informations sur l'apprenant (*cf.* Figure 2.4).

Six types d'informations sont définis par ce standard qui rend également possible l'extension de chacun d'eux. Un profil d'apprenant est donc défini par :

⁷ <http://ltsc.ieee.org/>

⁸ IEEE P1484.2 Learner Model Working Group, <http://ltsc.ieee.org/wg2/>

- des informations personnelles sur l'apprenant (PAPI Learner Personal). Il s'agit d'informations privées qui n'ont pas de lien direct avec l'apprentissage mais sont plutôt administratives (nom, adresse, numéro de téléphone, etc.)
- des informations relationnelles, relatives aux relations entretenues par l'apprenant avec les autres utilisateurs – professeurs, autres étudiants, etc. – (PAPI Learner Relations).
- des informations sur la sécurité – mot de passe, clés, etc. – (PAPI Learner Security).
- des informations sur la performance de l'apprenant (PAPI Learner Performance). Elles concernent l'histoire de l'apprenant, son travail en cours, et ses objectifs futurs.
- des informations « portfolio » qui constituent une collection représentative de travaux de l'apprenant utilisées en tant qu'illustrations de ses capacités (PAPI Learner Portfolio).
- des informations liées aux préférences de l'apprenant (PAPI Learner Performance) destinées à améliorer l'interaction homme – machine et à permettre l'adaptation automatique des systèmes aux besoins spécifiques de l'apprenant. Les préférences incluent des caractéristiques techniques, liées à l'interface, ou à la présentation des contenus. Ces préférences peuvent être explicitement identifiées par l'utilisateur ou être inférées à partir de son comportement

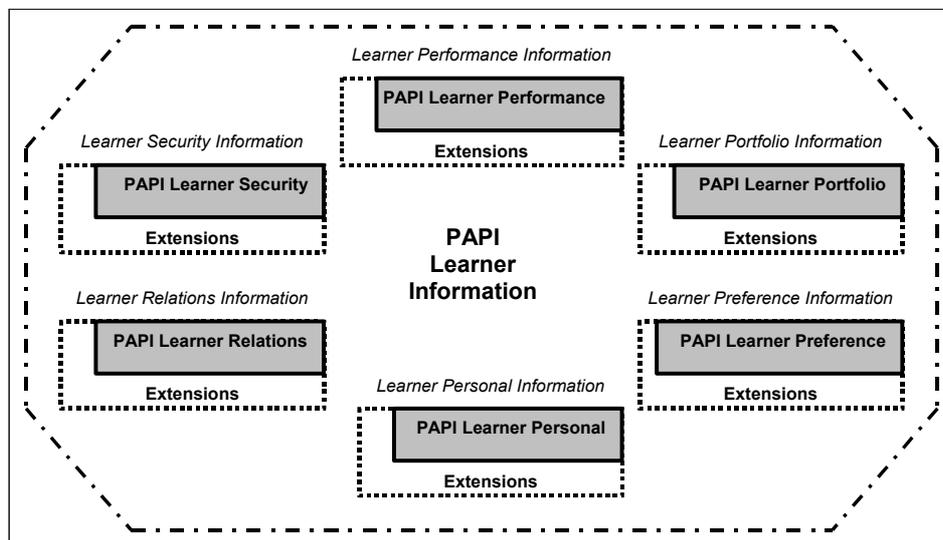


Figure 2.4 – Représentation de la proposition de standard PAPI Learner

Quelques travaux connexes à ceux précédemment cités complètent les travaux menés dans le cadre du LTSC. L'identification de l'étudiant est nécessaire à l'exploitation de son profil. Un groupe de travail⁹ du LTSC s'intéresse plus particulièrement à ces aspects. Un de ses objectifs est d'accroître les performances des systèmes en terme d'accès aux profils des utilisateurs.

Il existe à l'heure actuelle de multiples définitions pour des termes tels qu'« objectif d'apprentissage », « aptitude » ou encore « savoir ». Ce manque de consensus est un frein majeur dans une logique de définition de modèle de données réutilisables et d'échange de celles-ci. Un autre groupe de travail¹⁰ se consacre donc depuis décembre 1999 à la spécification d'un modèle de définition de compétences qui puisse faire référence.

⁹ IEEE LTSC Student Identifier Working Group - <http://ltsc.ieee.org/wg13/>

¹⁰ IEEE P1484.20 LTSC Competency Definitions Working Group - <http://ltsc.ieee.org/wg20/>

Les travaux menés dans le cadre de l'IEEE LTSC sont sans conteste les plus complets du point de vue de la prise en compte de l'utilisateur. Les informations recensées sont nombreuses et pertinentes pour le domaine d'application initial que constitue l'apprentissage. La portabilité et l'extensibilité des spécifications garantissent la réutilisation et l'adaptation des futurs standards à d'autres contextes.

2.6. Conclusion sur l'adaptation dans les SIW

Nous avons débuté ce chapitre en illustrant en quoi les SIW s'inscrivent dans la lignée des hypermédias d'une part et des Systèmes d'Information traditionnels d'autre part, héritant des caractéristiques de chacun d'eux. Nous avons ainsi mis en évidence la diversité des dimensions susceptibles d'être adaptées à l'utilisateur dans le SIW. Cinq dimensions principales en vue de l'adaptation ont été identifiées: le contenu, la navigation, la présentation, les fonctionnalités, et les conditions matérielles d'exploitation.

Nous avons ensuite fait le constat d'une terminologie riche associée à l'adaptation. Nous avons mis en relations les termes employés (adaptation, adaptabilité, adaptativité, personnalisation, customisation, etc.). Ceux-ci correspondant le plus souvent à des façons différentes d'envisager le processus visant à adapter le système à l'utilisateur (contrôle de l'utilisateur versus du système / aspects dynamiques, etc.). Parfois, deux termes peuvent être considérés comme des synonymes, la préférence allant à l'un ou à l'autre en fonction du domaine d'application qui l'emploie (le terme de personnalisation, par exemple, en matière de commerce électronique).

Tout processus d'adaptation, quel qu'il soit, repose sur un modèle de l'utilisateur, c'est-à-dire une représentation de ses caractéristiques dont le système tient compte dans la mise en œuvre des adaptations. Les différentes façons d'envisager une telle modélisation des utilisateurs ont été décrites.

Enfin, nous avons présenté certains travaux visant à illustrer l'ensemble des propos tenus dans ce chapitre. Nous avons ainsi exposé brièvement quelques propositions faites en matière d'adaptation dans les domaines de la recherche d'information, du commerce électronique et de l'enseignement à distance. Ces trois exemples nous ont notamment permis d'illustrer la diversité des représentations des utilisateurs, basées sur les buts de l'utilisateur, un historique de ses actions (tels que ses achats), ou encore son niveau de compétence, pour ne citer que quelques critères.

Avec ce chapitre, nous avons traité le premier aspect constituant l'état de l'art de cette thèse. En plus des questions relatives à l'adaptation dans les SIW, le contexte général de cette thèse est également celui de la conception de tels systèmes. Dans le chapitre suivant, ce second aspect de l'état de l'art est donc abordé à travers l'étude des méthodes proposées pour la conception d'hypermédias et d'applications basées sur le Web.

3. MÉTHODES DE CONCEPTION D'APPLICATIONS BASÉES SUR LE WEB

Dans ce chapitre, nous proposons une étude des approches de conception d'hypermédias et de SI basés sur le Web. Depuis 1993, date d'apparition de la première proposition en ce domaine [Garz93a], de nombreux travaux ont été publiés. Pour [Mecc99][Koch00], le caractère récent de la problématique explique en partie cet engouement.

Les propositions sont majoritairement issues de la communauté hypermédia. Le courant a été initié par HDM (*Hypermedia Design Model*), un modèle destiné à la conception topologique *globale* des applications (i.e. il n'aborde pas leur conception détaillée) [Garz93a][Garz93b]. HDM ne propose pas de démarche de conception, et à ce titre ne fait pas l'objet d'une présentation détaillée dans cet état de l'art¹¹. Néanmoins, cette proposition ayant donné lieu à différentes évolutions (dont RMM [Bala94][Isak95]) et étant très souvent référencée par les méthodes que nous présentons, nous en donnons un aperçu. HDM constitue le premier modèle conceptuel, basé sur un paradigme entité-relation [Chen76], enrichi pour supporter les caractéristiques des hypermédiâs. Les apports essentiels du modèle sont :

- la définition d'une structure interne complexe des entités qui permet d'offrir plusieurs représentations de leur contenu à travers la notion de perspective ;
- la spécification, en termes de liens, de relations entre les différents éléments structurels permettant de dériver les modalités de navigation dans l'hypermédia ;
- l'ajout de structures d'accès (index, visites guidées, collections) augmentant les possibilités de parcours dans l'hypermédia.

Cet état de l'art présente plusieurs travaux qui font, pour la plupart, référence dans la littérature et que nous avons retenus pour différentes raisons. RMM [Bala94][Isak95] s'appuie en partie sur HDM pour proposer la première méthode pour la conception d'applications

¹¹ On note cependant l'existence de plusieurs outils qui exploitent le modèle HDM, comme par exemple Jweb proposé par Bochicchio & al. [Boch99] ou encore HDM-Edit par Garzotto & al. [Garz01].

hypermédias. Hera [Fras01][Fras02] étend RMM dans le but d'introduire des capacités d'adaptation aux conditions matérielles d'utilisation. OOHDM [Schw96] innove en adoptant une conception orientée-objet. Sa version la plus récente ajoute une phase d'analyse des besoins [Güel00] en amont de la démarche initialement décrite. La proposition de WSDM [DeTr98] est la première à avoir préconisé une approche centrée utilisateur pour la conception des applications Web. AWIS-M [Gnah00][Gnah01] offre une démarche qui s'appuie notamment sur la modélisation des objectifs des utilisateurs [Gnah99] et de leurs profils pour guider la conception des sessions de navigation dans un SIW. UWE [Koch00a] a la particularité d'avoir initié le courant des approches basées sur le standard UML étendu par un profil spécifique. Enfin, WebML [Ceri01] est caractérisé par le recours à la technologie XML pour formaliser ses propositions.

Dans certains écrits (e.g. [Gnah00]), la présentation de ces méthodes suit une classification qui distingue les méthodes dédiées à la conception d'hypermédias de celles consacrées aux applications et SI basés sur le Web. Nous choisissons ici de ne pas faire cette distinction. En effet, la plupart des méthodes, à l'origine destinées aux applications non-basées sur le Web, ont souvent donné lieu à des évolutions ou extensions qui permettent la prise en compte d'une infrastructure basée sur le Web et qui intègrent parfois des caractéristiques relevant de l'approche SI.

Les méthodes présentées ne font pas toutes preuve de réelles capacités pour la spécification d'adaptation à l'utilisateur. C'est par exemple le cas pour RMM, OOHDM, et WSDM. Néanmoins, nous choisissons de les présenter dans la mesure où certains de leurs concepts ou extensions récemment proposées introduisent des caractéristiques d'adaptation. Les propositions les plus récentes, et notamment AWIS-M, UWE et WebML, intègrent de façon plus marquée cette dimension dans leur approche. L'étude de la prise en compte de l'adaptation dans les méthodes présentées fait plus particulièrement l'objet du chapitre 4.

3.1. RMM : Relationship Management Methodology

3.1.1 *Aperçu général de RMM*

RMM (*Relationship Management Methodology*) [Isak95] propose une méthode pour la conception et la construction des applications hypermédia. Le nom choisi par les auteurs reflète leur vision de l'hypermédia qui est envisagé comme un moyen de gestion des relations entre des éléments d'information.

L'intérêt de recourir à RMM pour la conception d'un hypermédia est évalué dans [Isak95] en regard de deux critères : la structuration de l'application et la volatilité de l'information (i.e. le fait que les données subissent des modifications au cours du temps). RMM est jugée non utile pour des applications à la structure faiblement marquée et au contenu évoluant peu (des applications artistiques par exemple). En revanche, l'utilité de l'approche est maximale pour les applications fortement structurées et dont la volatilité des données est très prononcée. C'est notamment le cas des applications interfaçant une base de données telles que les catalogues de produits. Pour les applications aux caractéristiques intermédiaires (i.e. faible structure/haute volatilité des données ou inversement), les apports de RMM sont estimés réduits.

RMM couvre une large partie du cycle de développement de l'application. Avant de décrire les différentes étapes du processus, nous présentons le modèle sur lequel repose principalement la méthode.

3.1.2 Le modèle RMDM

RMDM (*Relationship Management Data Model*) est le modèle utilisé dans l'approche pour décrire les informations et les mécanismes de navigation des hypermédias. Ce modèle, basé sur le modèle Entité-Relation (E-R) et sur HDM, offre trois sortes de primitives de modélisation.

Les *primitives du domaine* sont utilisées pour modéliser le domaine d'application en termes d'entités et d'attributs, et de relations associatives (1:1) ou (1:n) entre les entités. Le modèle conceptuel peut être augmenté par la définition de primitives appelées *slices* (part). Les *slices* sont utilisés pour procéder à des regroupements d'attributs appartenant à une même entité afin de présenter différentes versions de celle-ci. Enfin, RMDM propose différentes *primitives d'accès* pour modéliser la navigation. Les liens directionnels et bidirectionnels établissent des connexions entre les *slices* d'une même entité. Les possibilités de parcours inter-entités sont spécifiées grâce aux groupements, index et visites guidées. Les groupements sont des sortes de menus. Les index consistent en une liste d'items, chacun donnant l'accès direct à une instance d'entité. Les visites guidées élaborent un chemin d'accès linéaire et bidirectionnel aux items d'une collection. Les index et visites guidées portent des conditions (ou prédicats logiques) qui permettent de déterminer les instances à montrer.

3.1.3 Les étapes de la méthode

La méthode RMM porte sur la conception et la construction des hypermédias et s'insère dans un cycle complet de développement logiciel. Les phases d'analyse, en amont, et celles de tests et d'évaluation, en aval, ne sont pas les préoccupations premières de RMM. La conception est proposée en trois étapes, suivies de quatre autres dédiées au développement.

La première étape de conception est celle de la représentation du domaine d'application *via* un modèle E-R. La seconde étape est celle de la conception des *slices*. Elle vise à déterminer comment est présenté le contenu des entités aux utilisateurs et quelles sont les possibilités d'accès offertes. Un diagramme de *slices* est défini pour chaque entité et met en évidence les différentes représentations de celle-ci. L'artefact obtenu à l'issue de cette étape est appelé schéma E-R+ et correspond au modèle obtenu à l'étape précédente dans lequel chaque entité a été remplacée par son diagramme de *slices*. Dans l'hypermédia, chaque slice deviendra un nœud (i.e. une page HTML). L'étape suivante termine la phase de conception de l'hypermédia par la définition des aspects navigationnels. Les relations associatives du modèle E-R+ sont remplacées par des primitives d'accès (liens, groupement, index, visites guidées). La méthode donne quelques indications pour procéder à ce remplacement : établir un lien bi-directionnel pour une relation (1:1), dans le cas des relations (1:n), choisir une visite guidée ou un index, etc. A ce stade, le modèle RMDM est complètement défini et donne une représentation des modalités d'accès à implémenter. Les étapes suivantes sont spécifiques à la construction de l'hypermédia.

La quatrième étape débute le processus de développement. Elle consiste à définir un protocole de conversion permettant de transformer les éléments du modèle RMDM en objets de la plate-forme retenue pour le développement. Par exemple, le protocole pourra établir que les index seront implantés par des formulaires HTML. L'étape suivante concerne la conception de l'interface utilisateur. L'apparence et la localisation de chaque élément du modèle RMDM sont définies. La sixième étape est dédiée à la conception du comportement à l'exécution. Les décisions concernant l'implantation des mécanismes liés à la navigation et au suivi des actions de l'utilisateur sont prises. C'est également lors de cette étape que le choix de recourir à la génération dynamique (i.e. à la demande) des pages est fait si le degré de volatilité des données le justifie. Enfin, lors de la septième étape, le système est construit puis soumis à différents tests comme dans tout développement logiciel. La nature hypermédia de l'application nécessite, pour les auteurs [Isak95], de porter une attention particulière à l'évaluation des chemins de navigation.

3.1.4 Limites et extensions de la version originale de RMM

Quelques inconvénients de RMM ont été mis en évidence [Isak97a][Isak98b] tant en termes de possibilités de modélisation que de démarche. Nous présentons ici deux limitations formulées par les auteurs de la première version de RMM et qui ont donné lieu à des évolutions de leur proposition.

3.1.4.1 Limite des *slices* et leur évolution : les *m-slices*

La première insuffisance constatée de RMM est qu'elle ne permet pas la modélisation du contenu de pages Web complexes. Le concept de *slice*, s'il autorise la spécification de différentes représentations, est peu flexible dans la mesure où il ne peut être appliqué qu'à l'unité conceptuelle qu'est l'entité : la combinaison d'informations issues de plusieurs entités dans un même *slice* est impossible. Chaque *slice* (mono-entité) étant traduit dans l'hypermédia par un nœud, une succession de chemins de navigation est souvent nécessaire pour atteindre l'information souhaitée, avec le risque que cela implique de 'perdre' l'utilisateur.

Dans [Isak97a], un nouvel élément, appelé *m-slice*, est introduit pour pallier le manque de flexibilité des *slices*. Le préfixe '*m*' fait référence aux Matriochkas, les poupées russes emboîtables, dont le principe a inspiré les auteurs. Les *m-slices* permettent de regrouper des informations, issues de sources différentes, pour constituer des unités d'informations plus complexes intégrant par exemple des attributs appartenant à des entités différentes, des structures d'accès telles qu'un index, ou encore d'autres *m-slices*.

3.1.4.2 Redéfinition du processus de conception

Une autre limitation de RMM, dans sa première version, réside dans le fait que le processus de conception de l'hypermédia associé suit une approche qualifiée de descendante (*top-down*). Le modèle E-R défini au départ donne le squelette général de l'hypermédia qui est ensuite affiné par le remplacement des entités par leur diagramme de *slices*. Pour les auteurs, l'absence de primitives de modélisation permettant la construction de l'application à partir de composants plus petits est un inconvénient [Isak97b]. L'apport des *m-slices* décrits ci-dessus répond à cet inconvénient puisqu'ils sont construits à partir d'éléments de plus fine granularité (les attributs), et ensuite composés pour obtenir les nœuds de l'hypermédia. Toutefois, l'effet inverse est alors induit, l'approche de conception devenant totalement ascendante (*bottom-up*). L'inconvénient majeur réside dans la difficulté à produire un modèle complet de l'application donnant une représentation claire de l'hypermédia.

La solution apportée par les auteurs [Isak97b] introduit un nouveau diagramme et redéfinit le processus de conception. Le diagramme proposé est un *diagramme de l'application* qui remplace le modèle de navigation produit à l'étape 3 du processus original. S'il offre une vue globale de l'hypermédia comme le faisait le précédent modèle, c'est surtout dans la façon de l'obtenir que se situe le changement. En effet, le processus adopte ici un cycle alliant approches descendante et ascendante (*cf.* Figure 3.1) :

- une première version du diagramme de l'application est décrite pour mettre en évidence la structure générale attendue de l'hypermédia. Chaque unité du diagramme est alors décomposée en différents *m-slices* (en exploitant les possibilités de structuration décrites plus haut) conformément à une approche descendante.
- un diagramme d'application est ensuite reconstruit à partir des *m-slices* obtenus en suivant des étapes spécifiques [Isak97b]. Cette fois l'approche est donc ascendante.

L'intérêt de combiner les approches réside dans la possibilité de comparer les deux diagrammes obtenus et ainsi de mettre en évidence d'éventuelles divergences (*m-slices* oubliés, à

supprimer, liens manquants). A travers une démarche de conception itérative, le concepteur affine ces spécifications jusqu'à obtenir des diagrammes d'application (issus des approches ascendante et descendante) concordants.

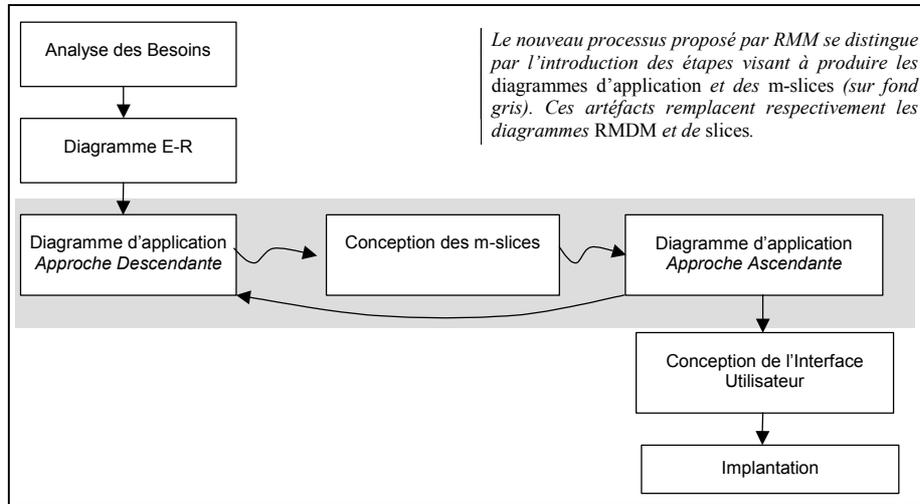


Figure 3.1 – Le processus de conception de RMM

Le processus décrit est celui redéfini dans [Isak98b]¹² par rapport à la première version présentée en section 3.1.3. (Reproduit d'après [Isak98b]).

3.1.5 Conclusion sur RMM

RMM est une démarche de conception et de développement particulièrement destinées aux applications hypermédias caractérisées par une structure stable et un contenu soumis à de fréquentes modifications.

A l'image de HDM sur lequel est basée RMM, le modèle E-R est utilisé pour spécifier le domaine. Le modèle obtenu est enrichi grâce aux concepts de *m-slices* et de relations structurelles pour lesquels des correspondances hypermédias sont préétablies. L'hypermédia est ainsi construit à partir du modèle RMDM, par une transformation des *m-slices* et entités en nœuds, et des relations (associatives et structurelles) en liens de navigation. Les contributions de RMM, en terme de modèle par rapport à HDM, relèvent principalement d'une description plus détaillée de l'utilisation qui doit être faite des différents éléments du modèle. Par ailleurs, RMM est la première proposition qui offre une démarche mettant en évidence les différentes étapes du processus de conception d'application hypermédia. Elle est de plus implémentée dans l'atelier RMCASE (Relationship Management Case Tool) [Diaz95] qui supporte les différentes phases de la méthode

La méthode RMM s'impose comme une référence dans le domaine. Elle est ainsi systématiquement référencée et comparée dans les travaux ultérieurs. Par ailleurs, RMM a fait l'objet de propositions qui visent à étendre les possibilités (spécification de liens de navigation

¹² Cette refonte du processus d'après [Isak98b] nous inspire quelques commentaires. L'étape d'analyse est mentionnée comme composante à part entière du processus ; toutefois, à notre connaissance, aucune proposition réelle n'est faite en la matière. Nous notons également la disparition des étapes 4 (définition du protocole de conversion) et 6 (définition du comportement à l'exécution). Or, les évolutions apportées à RMM ne portent pas sur ces aspects et ne permettent donc pas de s'en affranchir. Selon nous, leur absence dans ce nouveau processus n'est pas motivée (et n'est pas, à notre connaissance, justifiée par les auteurs).

dynamiques [Gabb98], aspects liés à la présentation [Fras01][Fras02]). En terme d'adaptation, la version originale de RMM est pauvre. Les *m-slices* ne sont pas exploités à des fins d'adaptation à différents utilisateurs : ils décrivent conceptuellement la façon dont l'information est présentée à tous les utilisateurs.

Si de nouveaux éléments de modélisation et une nouvelle version du processus de conception ont donc été proposés par les auteurs de la version originale de RMM, ils n'envisagent aucune évolution portant sur la conception de la présentation dans l'hypermédia. Nous décrivons dans la section suivante un travail plus récent [Fras01], basé sur RMM, qui vise à améliorer les spécifications relevant de la présentation.

3.2. Projet Hera

3.2.1 Aperçu général de Hera

Pour [Fras01], les possibilités de spécification offertes par RMM en terme de présentation sont insuffisantes. Il est principalement reproché à RMM le fait que quelques éléments de présentation sont introduits au niveau logique (i.e. dans le diagramme d'application) mais sans pour autant permettre une véritable spécification de la présentation. Dans le cadre d'un projet baptisé Hera [Houb00], les auteurs ajoutent une étape de *conception de la présentation*. De plus, ils introduisent une phase de conception des adaptations [Fras02] qui tient compte des caractéristiques matérielles, des préférences de l'utilisateur et de l'historique de ses navigations. Nous présentons les principales contributions de cette proposition en commençant par celles concernant la conception de la présentation.

3.2.2 Conception de la présentation

La conception de la présentation distingue un *niveau logique* et un *niveau présentation*. Au niveau logique, des caractéristiques de présentation plus avancées que les liens de navigation dans RMM, peuvent être exprimées grâce à la définition de nouveaux éléments de modélisation. Au niveau présentation, la production d'un véritable diagramme de présentation est préconisée afin d'établir un pont entre le niveau logique et l'implémentation.

3.2.2.1 Le niveau logique

Des extensions sont apportées au diagramme d'application sous la forme de relations entre *m-slices*. Ces relations traduisent à un haut niveau d'abstraction aussi bien des liens de navigation, que des relations temporelles ou spatiales.

Ces relations peuvent être simples (une source, une cible), ou multiples (plusieurs sources, plusieurs cibles) et dans ce cas, diverses configurations sont proposées sur la base des opérateurs AND et OR (par exemple, la configuration *multi-and target* indique que traverser le lien permet d'atteindre toutes les cibles [Fras01]).

Les relations entre *m-slices* peuvent être exprimées pour des *m-slices* emboîtés dans un autre (*internal m-slice relationship*). Ce type de relation interne nécessite que soit spécifié si le contenu du *m-slice* source reste ou non apparent lorsque le lien est franchi : on distingue pour cela les relations internes de préservation et de disparition de la source (resp. *preserving* et *vanishing source relationship*).

Les spécifications au niveau logique décrivent globalement la structure de la présentation sans donner une représentation concrète des choix en termes d'organisation navigationnelle, spatiale et temporelle. Ceci est abordé au niveau présentation.

3.2.2.2 Le niveau présentation

L'instauration d'un *diagramme de présentation* vise à décrire conceptuellement la présentation de l'hypermédia. Les éléments du niveau logique sont transformés en éléments de présentation : des correspondances sont établies entre les *m-slices* et des *régions*, les relations entre *m-slices* sont matérialisées par des *relations navigationnelles*, *temporelles* ou *spatiales*.

Une région est définie comme une collection d'attributs et d'éventuelles autres régions. Les attributs d'une région véhiculent l'information du domaine en faisant référence à l'entité dont est issue cette information. Chaque région est associée à une *aire* (une zone rectangulaire de l'écran, définie par une dimension et une position) dans laquelle elle sera affichée. La correspondance entre *m-slice* et région n'est pas forcément directe : il est possible d'affecter un même *m-slice* à différentes régions, ou différents *m-slices* à une même région. Selon nous, cette possibilité doit être utilisée avec précaution dans la mesure où elle ne doit pas conduire à une rupture de l'unité sémantique qu'un *m-slice* est supposé représenter.

La matérialisation des relations entre *m-slices* se traduit par l'expression de relations entre régions. Trois types de relation sont définis, chacun ayant une syntaxe et une sémantique différentes associées à sa réalisation. La *relation de navigation* représente les hyperliens classiques de navigation. Les *relations temporelles* expriment la notion de temps dans une présentation, que ce temps soit une durée intrinsèque à un élément d'information (e.g. une vidéo), ou introduit par le concepteur (e.g. lorsqu'il spécifie explicitement la durée d'affichage d'une information). Enfin, les *relations spatiales* sont utilisées pour décrire la mise en page des régions. La notion de lien n'a pas ici le sens d'un chemin parcouru mais décrit la position relative d'une région cible par rapport à une région source (e.g. en dessous, à droite, des coordonnées, etc.). Les trois types de relation que nous venons de décrire sont également caractérisés par un indice qui permet de les synchroniser.

La proposition de [Fras01] en vue de la spécification de la présentation d'un hypermédia augmente le processus de conception de RMM. Si des éléments de modélisation pour la présentation sont intégrés au niveau logique, c'est surtout via l'introduction d'un niveau spécifique que les aspects relatifs à la présentation sont pris en charge. Ce niveau conceptuel supplémentaire évite que les choix en terme de présentation ne soient relégués lors de la phase d'implémentation.

3.2.3 Conception des adaptations

Dans [Fras02], une phase de conception des adaptations est introduite. Les adaptations proposées conditionnent l'apparence des *m-slices* et la visibilité des relations entre *m-slices*. Cette proposition porte donc uniquement sur la dimension présentation. Les deux types d'adaptations définies dans la section 2.1, l'adaptabilité et l'adaptativité, sont envisagés. Nous rappelons que pour [Fras02] l'adaptabilité est vue comme un processus statique, *i.e.* la présentation est fixée avant que la navigation commence, et l'adaptativité comme un processus dynamique, *i.e.* la présentation change pendant que l'utilisateur navigue.

L'adaptabilité exploite des informations relatives aux capacités du dispositif d'accès (taille d'écran, capacité de la mémoire, vitesse du réseau, etc.) et aux préférences de l'utilisateur (mise en page souhaitée, langage, etc.). L'adaptabilité est basée sur la définition d'un *Profil* décrit en utilisant le vocabulaire CC/PP [Klyn01] (*cf.* également section 2.2.5). Outre les composants proposés par CC/PP pour la description des caractéristiques matérielles, logicielles et du navigateur, ce *Profil* inclut un composant dédié à stocker les préférences de l'utilisateur (*UserPreference*). Parallèlement, la description des *m-slices* est augmentée de conditions (*Slice Condition*) qui expriment ce qui est requis pour l'affichage du *m-slice* (ou d'une partie de celui-

ci). Enfin, des conditions d'adaptabilité (*Adaptability Condition*) établissent le lien entre le *Profil* et la description des *m-slices* : elles imposent que le *m-slice* ne soit intégré dans la page HTML que si les conditions associées sont satisfaites par le dispositif de sortie.

L'adaptativité est basée sur des informations concernant l'historique de la navigation d'un utilisateur. Ces informations sont stockées dans un modèle utilisateur et mises à jour au fur et à mesure des actions de l'utilisateur. Le modèle utilisateur définit un ensemble de couples attribut – valeur. Un attribut correspond à un concept du domaine, la valeur traduit l'intérêt de l'utilisateur pour ce concept. Les conditions d'adaptativité (*Adaptivity Condition*) définissent le seuil (*i.e.* une valeur minimale d'intérêt) au-delà duquel il est jugé utile de présenter un *m-slice*. L'adaptativité apparaît réellement dans le recours à des règles qui réévaluent l'intérêt de l'utilisateur pour un concept en fonction des pages qu'il a visitées. Ces règles (*Update Rule*) sont basées sur un paradigme événement – condition – action emprunté à [DeBr99] et décrites dans le modèle utilisateur. Globalement, lorsqu'un lien est suivi par l'utilisateur, les concepts associés voient leurs valeurs augmenter dans le modèle de cet utilisateur.

3.2.4 Conclusion sur Hera

Hera est caractérisée par une véritable étape de conception de la présentation qui vise à faire le lien entre des spécifications de niveau logique et l'implémentation. La modélisation de la présentation proposée dans Hera constitue également un support pour l'expression des capacités d'adaptation de l'hypermédia. Ces capacités sont de deux ordres :

- l'adaptabilité, en tant que processus statique permettant d'adapter la présentation préalablement à la navigation.
- et l'adaptativité, en tant que processus dynamique permettant que la présentation change pendant la navigation [Fras02]).

Adaptabilité et adaptativité conditionnent toutes deux l'apparence des *m-slices* et la visibilité des relations entre eux. Cette récente proposition de [Fras02] est une évolution de RMM qui va dans le sens de plus d'adaptation. Néanmoins, parce qu'elle est avant tout proposée pour réduire les inconvénients de RMM en termes de présentation et de navigation, l'approche ne porte, à ce jour, que sur ces aspects.

3.3. OOHDM : Object Oriented Hypermedia Design Methodology

3.3.1 Aperçu général de OOHDM

La méthode OOHDM (*Object Oriented Hypermedia Design Methodology*) [Schw96] [Schw98] s'inspire tout comme RMM du modèle HDM. Elle se distingue cependant des méthodes décrites précédemment par le recours à une démarche de conception *orientée objet*. Dans les premiers articles sur OOHDM [Schw96], l'approche objet citée comme référence est OMT [Rumb91]. Dans les suivants, les auteurs lui préfèrent UML [Booc99][Rumb99] notamment en raison de sa qualité de standard [Ross99] et parce que leurs propositions pour l'analyse des besoins [Güel00] s'en inspirent (cf. section 3.3.2). Pour les auteurs, l'approche objet est particulièrement utile dans le cas d'applications dont le contenu informationnel est sujet à des modifications ou lorsque des requêtes ou calculs complexes sont envisagés sur ce contenu. Ils proposent en effet d'exprimer les actions correspondantes via la composante comportementale des classes (*i.e.* en spécifiant des méthodes de classes).

A l'origine, la méthode proposait quatre étapes – modélisation conceptuelle, conception de la navigation, conception de l'interface abstraite et implémentation – chacune étant centrée sur un aspect de la conception. Un processus à la fois itératif et incrémental est préconisé et conduit à la

production d'un modèle, construit ou enrichi, à chaque étape. Les travaux dans lesquels OOHDMM est cité référencent de façon quasi systématique cette version en quatre étapes de la méthode.

Toutefois, dans [Güel00], les auteurs proposent une autre étape étendant le processus en amont des quatre autres (*cf.* Figure 3.2). Cette étape, consacrée à la collecte des besoins, constitue alors le point de départ de la conception. L'objectif de la contribution est de combler le vide existant – notamment dans HDM, RMM, et dans la version originale de OOHDMM – entre l'expression des besoins et l'activité de conception. La méthode est basée sur différentes spécifications permettant de capturer les besoins des utilisateurs (en termes d'information et d'action). L'activité de conception qui en découle est alors dirigée par les aspects relevant des interactions utilisateur/système et de la navigation. Le processus original de OOHDMM est modifié en conséquence : les étapes sont globalement identiques, mais la démarche associée et les spécifications produites sont nettement plus détaillées. Ceci est principalement constaté pour les étapes de modélisation conceptuelle et navigationnelle.

Nous présentons l'approche OOHDMM telle qu'elle est décrite dans [Guél00] et [Ross01]. Nous intégrons à plusieurs reprises la description d'éléments définis dans la première version de l'approche et nécessaires à la compréhension du processus de conception. Pour plus de lisibilité, nous employons *OOHDM-96* pour faire référence à la première version de l'approche, et simplement *OOHDM* lorsque nos propos concernent la version de [Güel00].

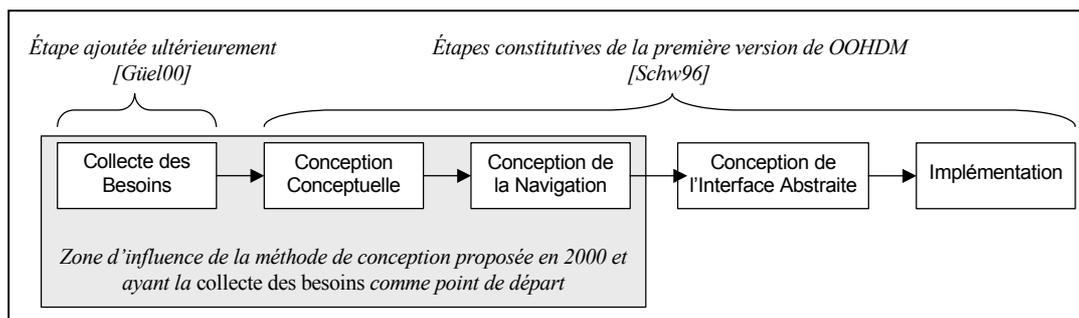


Figure 3.2 – Les étapes de OOHDMM dans sa version proposée en 2000

Aux quatre étapes de la proposition initiale est ajoutée en amont une étape de collecte des besoins qui influence sensiblement les étapes de modélisation conceptuelle et de la navigation

3.3.2 Collecte des Besoins

L'étape de collecte des besoins tient une place importante dans le processus de OOHDMM puisque les spécifications obtenues seront utilisées ensuite lors de la conception du schéma conceptuel et de la navigation. La collecte des besoins est réalisée selon une succession de cinq phases :

- Identification des rôles (au sens d'acteur dans UML) et des tâches qui doivent être supportées par l'application ;
- Spécification de scénarios (*i.e.* des descriptions narratives des tâches) par les utilisateurs eux-mêmes selon les principes proposés dans [Caro95].
- Spécification des cas d'utilisation [Jaco99], chacun regroupant les scénarios décrivant la même tâche.
- Spécification de Diagrammes d'Interaction Utilisateur (User Interaction Diagram – UID) [Vila00a][Vila00b] dont l'objectif est de capturer les interactions observables de

l'extérieur, entre un utilisateur et une application. Un UID est défini par cas d'utilisation identifié précédemment.

- Validation des Cas d'Utilisation et Diagrammes d'Interaction Utilisateur à travers un dialogue entre le concepteur et les utilisateurs impliqués.

À l'issue de cette étape, la modélisation conceptuelle est engagée à partir des spécifications obtenues jusqu'ici : scénarios, cas d'utilisation et diagrammes d'interaction utilisateur. Dans les prochaines sections, nous présentons la suite du processus, en faisant parfois référence aux éléments d'une conception basée sur OOHDm-96 et repris dans [Guël00].

3.3.3 *Modélisation conceptuelle*

Cette étape vise à définir le modèle conceptuel du domaine d'application selon un formalisme orienté objet à la UML. Les classes du schéma seront utilisées ultérieurement pour dériver les nœuds de l'hypermédia tandis que les relations donneront certains liens.

La démarche, qui s'inscrit dans le prolongement de la collecte des besoins, consiste à appliquer des règles de transformation aux UID afin d'obtenir un diagramme de classes qui représente le domaine d'application. Ces règles, détaillées dans [Vila00a][Vila00b], s'appuient principalement sur les techniques standard de normalisation de schéma définies dans [Elma94]. Globalement, le processus consiste, pour chaque UID, à trouver à partir des éléments qui le composent, les classes, les attributs de classes, et relations sémantiques entre classes.

3.3.4 *Conception de la navigation*

Cette étape est considérée comme primordiale dans les deux versions de OOHDm. La caractéristique essentielle de l'approche consiste à construire le Modèle de la Navigation comme une *vue* sur le modèle conceptuel obtenu à l'étape précédente. Il s'agit ici d'une différence majeure par rapport aux propositions antérieures : on considère ici que les objets accédés par navigation *ne sont pas* les objets du modèle conceptuel, mais des objets élaborés, via des mécanismes de vues, à partir d'un ou plusieurs objets conceptuels. Le modèle de la navigation dans l'hypermédia sera obtenu à l'issue d'une étape qui repose sur différentes phases décrites ci-dessous.

3.3.4.1 Modélisation de la navigation correspondant à une tâche

Pour chaque tâche, les cas d'utilisation, scénarios, UID et partie du modèle conceptuel correspondant sont analysés en vue de construire un *contexte navigationnel* [Schw96]. Ces contextes constituent un moyen d'organiser l'espace de navigation en définissant des ensembles constitués d'informations, de liens et d'éventuels autres contextes. Ils possèdent une structure de navigation interne (séquentielle, index, etc.), un point d'entrée et des index associés. Les contextes navigationnels les plus courants définissent des regroupements, en fonction de propriétés que doivent satisfaire les éléments, d'objets issus d'une même classe ou impliqués dans une même relation. Des notations graphiques propres aux contextes sont données dans [Schw98] et des lignes directrices sont proposées dans [Guël00] pour procéder à la création de tels contextes.

La description des contextes navigationnels est complétée par l'ajout de *cartes de spécification* qui font apparaître des informations qui précisent la représentation graphique ou des règles permettant de dériver l'information délivrée par le contexte.

3.3.4.2 Production du diagramme de contexte navigationnel de l'application

Cette phase consiste à produire un seul diagramme à partir des contextes navigationnels définis pour chaque tâche. L'union des différents contextes est menée à bien et d'éventuelles adaptations sont entreprises afin d'obtenir un diagramme cohérent qui reflète les besoins de l'ensemble des utilisateurs. Ce diagramme n'est toutefois pas suffisant pour refléter précisément la navigation dans l'hypermédia. L'objectif de la phase suivante est d'en affiner la conception.

3.3.4.3 Spécification du modèle de navigation de l'application

La spécification du modèle de navigation de l'application repose sur un ensemble de modèles parmi lesquels figurent le diagramme final de contexte navigationnel et les cartes de spécifications issus des étapes précédentes. D'autres spécifications complètent le modèle :

Un schéma des *classes navigationnelles* est établi. Les classes navigationnelles, aussi appelées *nœuds*, décrivent le contenu à travers lequel l'utilisateur va naviguer. Un nœud est défini comme une vue sur une ou plusieurs classes du modèle conceptuel. Un nœud possède des attributs issus de ces classes et des *liens* permettant le passage vers un autre nœud. Ces nœuds sont obtenus à partir des relations conceptuelles (et éventuellement, établissent de nouvelles relations).

Un nœud peut être impliqué dans différents contextes. Dans ce cas, il peut être nécessaire d'exprimer un certain nombre d'informations qui sont dépendantes du contexte (i.e. des informations particulières délivrées en fonction du contexte, ou des mécanismes d'accès propres au contexte). Pour cela, des classes spécifiques, appelées *InContext* [Schw96], sont définies pour chaque nœud et chaque contexte dans lequel le nœud apparaît. Une classe *InContext* a une structure similaire à celle du patron de conception « décorateur » de Gamma [Gamm95]. Lorsqu'un nœud est atteint dans un contexte donné, les informations portées par la classe *InContext* correspondante viennent s'ajouter aux informations du nœud de base. La classe *InContext* est donc un support pour *apporter de l'information supplémentaire* en fonction d'un contexte et n'est utilisée ni à des fins de masquage ni de redéfinition de l'information.

Une *table de correspondance conceptuelle-navigationnelle* est ensuite établie. Pour chaque nœud, une classe dite 'de base' est choisie dans le modèle conceptuel. Une comparaison est pratiquée entre les attributs et méthodes du nœud et ceux de la classe de base. Pour chaque attribut ou méthode du nœud qui n'a pas d'équivalent dans la classe de base, une règle de correspondance est spécifiée et décrit comment l'obtenir à partir du modèle conceptuel.

3.3.5 Conception de l'interface abstraite

Lorsque la structure navigationnelle est définie, les aspects relevant de l'interface sont abordés. OOHDM s'appuie sur l'approche ADV (*Abstract Data View* [Cowa95]) pour décrire l'interface utilisateur. Les ADV sont des objets interface dont certains attributs définissent les propriétés en termes d'organisation et d'apparence. D'autres attributs établissent la liste des événements que l'utilisateur peut déclencher et auxquels l'ADV doit répondre. Les ADV sont associés aux objets du modèle de navigation OOHDM tels que les nœuds, les index, etc. Les aspects dynamiques (e.g. le comportement à produire en réaction à un événement) sont décrits par des *ADV-charts*, un formalisme dérivé des *statecharts* [Carn94].

3.3.6 Implémentation

Lors de l'étape d'implémentation, les objets conceptuels, de navigation et d'interface décrits par les différents modèles de OOHDM sont traduits en fonction de l'environnement choisi pour

l'implémentation. Une discussion sur les difficultés et techniques préconisées pour établir les différentes correspondances est proposée dans [Schw98].

3.3.7 Conclusion sur OOHD

La méthode OOHD (version proposée en 2000 [Güel00]) est caractérisée par un processus en cinq étapes qui débute par une analyse des tâches que les utilisateurs peuvent effectuer dans l'application. OOHD introduit donc une dimension fonctionnelle absente dans les méthodes précédentes. Les besoins fonctionnels sont exprimés via des scénarios, cas d'utilisation et diagrammes d'interaction utilisateurs. Ces spécifications sont ensuite utilisées dans les étapes de modélisation conceptuelle et navigationnelle. Des schémas *objets* spécifiques sont définis à chaque étape : modèle conceptuel, diagramme de contextes navigationnels, diagramme de classes navigationnelles et de classes InContext (similaire aux *perspectives* dans HDM). La spécification de l'interface abstraite consiste ensuite à décrire les objets qui constituent l'interface de l'application à partir des éléments de navigation modélisés. Enfin, la dernière étape établit la correspondance entre les objets d'interface et les objets d'implémentation.

OOHD-Web [Schw99] est un environnement qui a été proposé pour supporter la première version de OOHD [Schw96]. Il offre des outils pour mener à bien les phases de conception de la navigation et des interfaces abstraites ainsi que la phase d'implémentation. Une librairie de scripts CGI est proposée pour stocker les spécifications des modèles (sous formes de tables dans une base de données relationnelle) et les exploiter lors de la construction de pages HTML. Aucune évolution visant à tenir compte de la version de la méthode décrite dans [Güel00] n'est proposée.

OOHD se distingue des approches précédentes par le recours à une approche de conception objet. Le degré de relative indépendance offert entre les niveaux conceptuel et navigationnel permet l'application de la méthode à un modèle conceptuel objet existant. Mais surtout, le fait de considérer les objets de navigation comme des vues sur le modèle conceptuel augmente la possibilité de spécifier différentes versions d'hypermédi

à partir d'un même modèle conceptuel. Ceci contribue à une prise en compte de l'adaptation du contenu, et dans une certaine mesure de la navigation et des fonctionnalités offertes. Toutefois, dans OOHD l'adaptation permet uniquement de choisir la version de l'hypermédi

à proposer à un utilisateur, en fonction du groupe d'acteurs auquel il appartient. Dans les *cartes de spécification*, les différents groupes (e.g. employés, dirigeants, etc.) auxquels s'adresse le contexte navigationnel sont mentionnés [Güel00]. Nous supposons que le choix de l'hypermédi

présenté à l'utilisateur s'appuie sur cette information. Néanmoins, rien n'est précisé dans la méthode sur la représentation des utilisateurs et des groupes, l'établissement des relations d'appartenance entre eux, et l'exploitation de telles connaissances.

3.4. WSDM: Web Site Design Method

3.4.1 Aperçu général de WSDM

La méthode de conception de site Web WSDM¹³ [DeTr98][Goed98] se distingue des propositions précédentes par l'accent mis sur la prise en compte des besoins du public cible de l'application¹⁴. Plutôt que prendre les données comme point de départ de la conception, WSDM s'intéresse en premier lieu aux utilisateurs : la proposition est décrite comme une approche

¹³ Prononcer « WiSDoM »

¹⁴ Comme nous l'avons vu (section 3.3), OOHD [Güel00] adopte également une approche qui part des besoins de utilisateurs, mais WSDM a été la première à le proposer.

centrée utilisateur par opposition aux approches guidées par les données (HDM, RMM, OOHDM-96). L'approche centrée utilisateur de WSDM est justifiée par le fait que le public d'un site est composé d'utilisateurs et de visiteurs aux besoins hétérogènes. L'attention portée à la diversité des attentes permet de développer des applications mieux ciblées et plus satisfaisantes pour leurs utilisateurs.

Les différentes étapes de la méthode de conception de WSDM sont décrites par la Figure 3.3. La première d'entre elles (appelée *Mission Statement*), consiste en une déclaration de la 'mission' que doit remplir le site en répondant aux questions suivantes : quel est le *but* du site, quel en est le *sujet* et quel est le *public* ciblé. Les réponses fournies sont décrites en langue naturel (cf. [DeTr01] pour un exemple). À partir de ces spécifications, la modélisation des utilisateurs est engagée en deux temps, par la *classification des publics* tout d'abord, puis, par la *caractérisation* de ceux-ci.

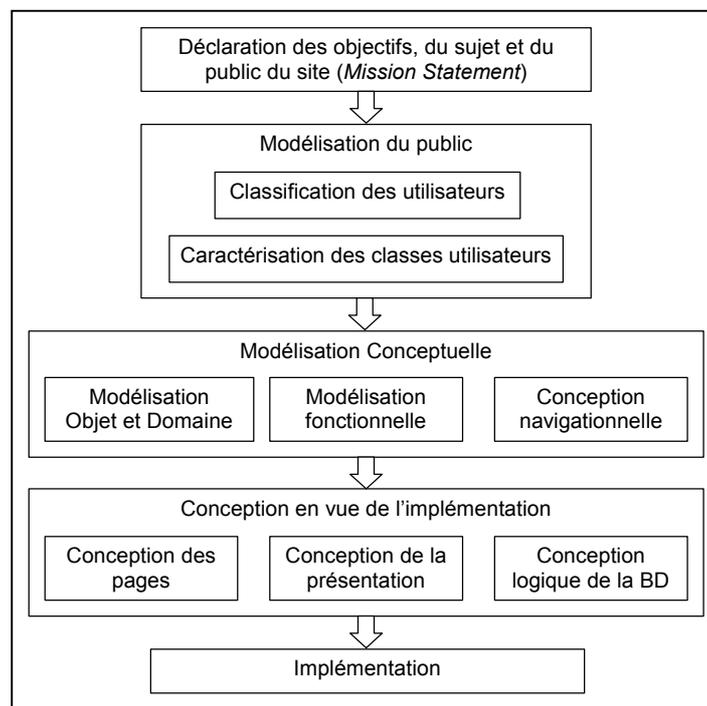


Figure 3.3 – Les différentes étapes de la méthode WSDM (D'après [Cast01]).

La modélisation conceptuelle amorcée ensuite se décline en trois phases : *modélisation objet et modélisation du domaine* (qui constituent une seule phase), *modélisation fonctionnelle*¹⁵ puis *conception navigationnelle*. L'étape suivante porte sur la *conception en vue de l'implémentation*. WSDM s'achève par la phase d'*implémentation* du site dans l'environnement choisi (HTML, XML, etc.). Nous décrivons dans ce qui suit, de façon un peu plus détaillée, les étapes de modélisation du public, de modélisation conceptuelle et de modélisation en vue de l'implémentation.

¹⁵ Mentionnés dans [Cast01], les concepts de la modélisation fonctionnelle et la démarche envisagée ne sont cependant pas décrits, à notre connaissance.

3.4.2 Modélisation du public

La modélisation du public comprend deux phases. La *classification des utilisateurs* (désignés par *audience* dans [Cast01]) est menée à partir de l'étude des déclarations de mission du site établies lors de la première étape. Des *classes utilisateurs* sont identifiées : elles regroupent les utilisateurs qui présentent des besoins informationnels et fonctionnels similaires (*user requirements* [Goed98], désignés par *UR* dans la suite). Les classes utilisateurs ne sont pas disjointes (un utilisateur peut appartenir à plusieurs classes utilisateurs). Dans [Cast01], une approche qui structure les classes d'utilisateurs en hiérarchie est proposée.

Les classes utilisateurs sont décrites plus finement dans un second temps, lors de la phase de *description des classes utilisateurs*. Une analyse est menée pour chaque classe utilisateur afin d'identifier différents sous-groupes d'utilisateurs, appelés *perspectives*, à l'intérieur de celle-ci. Les perspectives traduisent des sous-classes utilisateur, qui partagent les mêmes UR, mais qui ont néanmoins des caractéristiques et besoins d'utilisation (*usability requirements*) spécifiques. Il s'agit de critères tels que la langue utilisée, un niveau de compétence, des préférences pour la présentation, etc., qui justifient de délivrer différemment l'information. La seule représentation qui est donnée pour les descriptions des classes utilisateurs est informelle (description textuelle) [DeTr98].

3.4.3 Modélisation conceptuelle

3.4.3.1 Modélisation objet et modélisation du domaine

La *modélisation objet* vise à définir plus formellement les UR exprimés dans les classes utilisateurs et perspectives. Une notation de type OMT ou E-R est utilisée dans ce but. Un groupement d'objets conceptuels (*objects chunk*) est construit pour chaque UR : il montre les types d'objet et les relations entre eux impliqués par le UR. L'ensemble des *chunks* définis pour une classe utilisateur est regroupé en un modèle appelé *User Object Model* (UOM) qui donne donc la représentation du domaine d'application pour une classe utilisateur particulière. A partir de chaque UOM, un Modèle Objet Perspective (*Perspective Object Model* – POM) est défini lorsqu'il est nécessaire d'offrir des versions différentes de types d'objets conformément aux spécifications des perspectives. Dans WSDM, ces versions sont représentées par des classes appelées *perspectives variantes* qui redéfinissent un type d'objet.

Le modèle du domaine est obtenu lors d'une phase de modélisation qui correspond à une fusion de tous les UOM et qui nécessite de résoudre des inconsistances ou conflits éventuels.

3.4.3.2 Conception Navigationnelle

La seconde étape de la modélisation conceptuelle vise à produire le modèle de la navigation. Ce modèle est composé de pistes de navigation (*navigation tracks*) qui décrivent les possibilités de cheminement à travers l'information. Un groupement de navigation (*navigation chunk*) est déterminé pour chaque *User Requirement* défini à l'étape précédente. Les groupements de navigation décrits pour une même classe utilisateur sont regroupés pour former une piste de navigation. Le site Web est modélisé par un modèle correspondant qui intègre l'ensemble des pistes de navigation.

Des représentations graphiques sont proposées pour décrire les groupements et pistes de navigations. Des *composants information* véhiculent le contenu tels que décrit dans les perspectives, des *composants navigation* correspondent un groupement de liens d'accès (à des composants navigation intermédiaires ou des composants information), et des *composants externes* font référence à des éléments appartenant à un autre site. Un composant navigation particulier joue le rôle de point d'entrée de la piste de navigation.

3.4.4 *Modélisation en vue de l'implémentation*

La modélisation en vue de l'implémentation repose sur la définition des pages qui vont constituer le site en termes de structure et d'apparence. WSDM préconise de suivre des règles de conception permettant d'aboutir à un site cohérent, convivial et efficace dans la mise à disposition de l'information. Par exemple, la structuration des pages doit contenir un nombre d'informations ne mettant pas l'utilisateur en situation de surcharge cognitive, des informations contextuelles doivent également être fournies à l'utilisateur afin de faciliter son repérage dans le site. La modélisation en vue de l'implémentation comporte également une phase de conception logique de la base de données supportant l'information du domaine.

3.4.5 *Conclusion sur WSDM*

WSDM adopte une approche centrée utilisateur qui consiste à déterminer les besoins des utilisateurs puis à identifier des *classes utilisateurs* qui « représentent » ces besoins. Des variantes de classes utilisateurs sont décrites pour affiner la modélisation des différents publics. Il n'existe néanmoins pas de véritable modèle de l'utilisateur dans WSDM. Les informations qui caractérisent les différents utilisateurs sont reportées au niveau du modèle de données. Plusieurs modèles objets sont construits, offrant ainsi une représentation du domaine d'application en adéquation avec les caractéristiques de chacune des variantes de classes utilisateurs. Un modèle de navigation et de manipulation est ensuite généré à partir des spécifications établies au préalable. L'adaptation autorisée par WSDM revient, dans cette approche, à créer un hypermédia correspondant aux besoins des différentes catégories d'utilisateurs identifiées. Il s'agit donc d'un processus relevant d'une vision statique de l'adaptation.

3.5. Adaptive Web-based Information System - Method – AWIS-M

3.5.1 *Aperçu général de AWIS-M*

AWIS-M (*Adaptive Web-based Information System – Method*) [Gnah00][Gnah01] est une approche méthodologique pour l'ingénierie des SIW. L'approche est centrée sur l'utilisateur et orientée par les buts de celui-ci. Un objectif d'adaptation aux besoins et préférences individuels des utilisateurs est également annoncé.

AWIS-M repose sur un ensemble de modèles, appelés *modèles de produits*, dédiés à la modélisation des utilisateurs potentiels et de leurs buts, à la conception du contenu, de la navigation et de l'interface de l'application. Elle propose également des directives méthodologiques définissant la façon dont l'ingénieur d'application Web doit procéder pour créer et transformer les modèles de produits. La démarche comprend six étapes représentées dans la Figure 3.4. Nous abordons tout d'abord les modèles de produits avant de présenter les principes de chaque étape de la démarche.

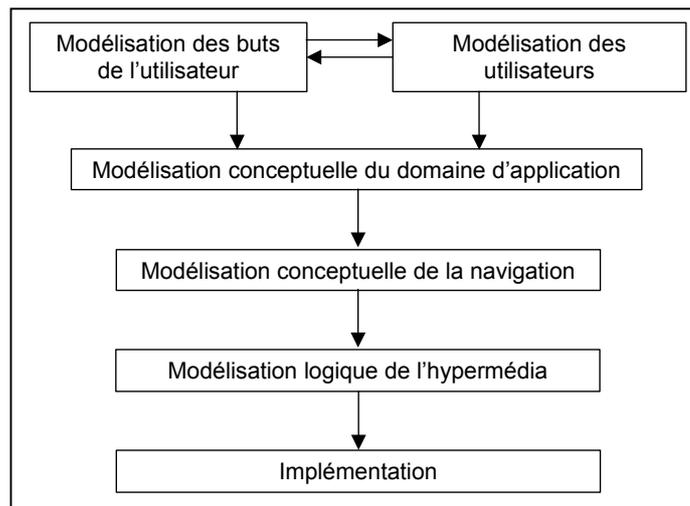


Figure 3.4 – Les étapes de la démarche dans AWIS-M

3.5.2 Les modèles de produits

Les modèles de produits sont organisés selon quatre dimensions : identification et description des utilisateurs potentiels et de leurs buts, gestion du contenu informationnel, définition de la structure navigationnelle et définition de l'interface utilisateur. Les modèles de produits sont par ailleurs décrits à deux niveaux d'abstraction : Gnaho distingue des modèles de niveau conceptuel et des modèles de niveau logique. Ces derniers offrent une représentation intermédiaire entre le niveau conceptuel et l'implémentation.

3.5.2.1 Les modèles de produits de niveau conceptuel

a – Le modèle des Objectifs Utilisateurs

Le premier modèle de niveau conceptuel est celui des *Objectifs Utilisateurs*. Ce modèle est destiné à la représentation et à la description des besoins, vus comme des objectifs de haut niveau, auxquels doit répondre le SIW. L'auteur distingue des *Objectifs Utilisateurs Informationnels* et des *Objectifs Utilisateurs Applicatifs*. Les premiers permettent à l'utilisateur d'accéder à l'information et de naviguer dans les informations. Les seconds correspondent à la réalisation par l'utilisateur de tâches complexes. Une approche formelle, issue de [Prat97], est utilisée pour décrire ces objectifs. Un Objectif Utilisateur est exprimé par une clause composée d'un *verbe* et d'un ensemble de *paramètres*, chacun étant associé à une fonction sémantique. Les paramètres qui entrent dans la description d'un objectif sont de quatre types, certains étant affinés en sous-types :

- la *Cible* qui désigne les entités affectées par l'objectif ; il peut s'agir d'un *Objet* ou d'un *Résultat* selon que la cible pré existe ou non à la réalisation de l'objectif.
- la *Façon* qui exprime la *Manière* et/ou les *Moyens* utilisés pour atteindre le but ;
- la *Direction* qui identifie la *Source* ou la *Destination* d'un objectif ;
- le *Bénéficiaire* qui correspond à la personne ou au groupe en faveur de qui l'objectif doit être atteint.

Les *Objectifs Utilisateurs* peuvent être décomposés en sous-objectifs et sont reliés à eux par des *liens d'affinement* [Gnah00]. Ces liens sont porteurs d'une sémantique, basée sur les

opérateurs « Et/Ou », qui conditionne la réalisation de l'objectif décomposé. La décomposition obtenue est appelée *Grappe de Réduction des Objectifs Utilisateurs*.

b – Le modèle des Utilisateurs

Le *modèle des Utilisateurs* est le second modèle décrit au niveau conceptuel. Son but est de décrire les attentes et les caractéristiques des différentes catégories d'utilisateurs susceptibles d'utiliser le SIW. Ce modèle repose sur les concepts de *Rôle Utilisateur*, de *Profil Utilisateur* et de *Catégorie d'Utilisateur*.

Un *Rôle Utilisateur* possède un *nom* et est défini par un sous-ensemble d'*Objectifs Utilisateurs*. On peut voir le rôle comme la définition des différentes fonctionnalités qu'un utilisateur est autorisé à mener à bien dans le SIW en fonction de son statut. Par exemple, dans le cas du SIW d'une bibliothèque, les utilisateurs jouent le rôle d'Abonné, de Visiteurs, de Responsable des emprunts, etc.

Un *Profil Utilisateur* est décrit comme une agrégation de *Paramètres de Profil* qui représentent des informations qui caractérisent un utilisateur. Des *Paramètres Élémentaires* sont décrits par un *nom* et une *caractéristique linguistique*, cette caractéristique étant définie par une *valeur linguistique* et un *modificateur linguistique* (adverbe optionnel permettant d'atténuer ou d'amplifier la valeur linguistique). Un exemple de Paramètre Élémentaire est « Niveau d'Anglais (très faible) » où *Niveau d'Anglais* est le nom, *faible* est la valeur linguistique et *très* le modificateur (amplificateur).

Les Paramètres Élémentaires peuvent être utilisés pour construire des *Paramètres Complexes* à l'aide d'opérateurs appelés *fonctions linguistiques d'agrégation*. Ces fonctions autorisent la conjonction (« Et ») de paramètres, leur disjonction (« Ou ») ou encore la spécification d'un compromis entre paramètres (« Comp »). Un exemple de compromis donné dans [Gnah00] est : « (Connaissance Internet (faible)) 'Comp' (Connaissance Internet (Bon)) ». Ceci exprime que l'on cherche un compromis entre les deux valeurs indiquées du paramètre.

Le concept de *Catégorie d'Utilisateurs* est défini comme l'association d'un *Rôle Utilisateur* et d'un *Profil Utilisateur*. Une telle association représente un sous-ensemble d'utilisateurs ayant les mêmes besoins en termes d'informations et de services et présentant des caractéristiques communes. Nous présentons dans la Figure 3.5 l'exemple donné par Gnaho pour illustrer une telle catégorisation [Gnah00].

À partir de cet exemple, nous discutons le choix d'une telle représentation. Gnaho souligne, à juste titre, que *Rôle Utilisateur* et *Profil Utilisateur* sont complémentaires mais orthogonaux : un *Rôle Utilisateur* peut être associé à plusieurs *Profils Utilisateurs*, et de façon similaire, des utilisateurs de profil identique peuvent jouer des rôles complètement différents. Une catégorie d'utilisateur étant une association entre ces deux concepts, le modèle nécessite que soient définies autant d'associations, donc de *Catégories d'Utilisateurs*, que de combinaisons *Rôle/Profil* possibles.

Ainsi, à partir de l'exemple précédent, quatre catégories devraient logiquement être décrites pour obtenir un modèle complet : aux deux mentionnées (*Abonné_Débutant* et *Visiteur_Expert*) s'ajoutent *Abonné_Expert* et *Visiteur_Débutant*. On voit donc que cette catégorisation des utilisateurs implique $m \times n$ spécifications de catégories (où m est le nombre de *Rôles Utilisateurs* et n le nombre de *Profils Utilisateurs* définis).

Nous signalons que l'auteur introduit, dans la définition des profils, des valeurs linguistiques pouvant être altérées par des modificateurs linguistiques d'une part, et des fonctions linguistiques d'agrégations pour composer les paramètres de profil d'autre part. Ces deux éléments permettent

d'apporter de nombreuses nuances dans la caractérisation des profils. Une telle approche offre potentiellement aux concepteurs les moyens de représenter finement les utilisateurs en tant qu'individu. Conformément au modèle proposé, cette possibilité peut conduire à un nombre très élevé de Profils Utilisateurs et, par suite, augmente proportionnellement le nombre de Catégories d'Utilisateurs à définir. En effet, toute chose étant égale par ailleurs, une différence de valeur, voire une différence du modificateur de cette valeur, pour un paramètre de Profil Utilisateur nécessite de définir un nouveau profil et donc une nouvelle catégorie.

```

< Catégorie_Utilisateurs ID = « Abonné_Débutant »>
  <Rôle ID = « Abonné» Objectifs = « Consulter_Info_Générale, Réserver_Ouvrage, ... » />
  <Profil ID = « Débutant » Description = « Connaissance_Internet (Novice) OU Pratique_du_Système (médiocre) »
/>
</Catégorie_Utilisateurs>

< Catégorie_Utilisateurs ID = « Visiteur_Expert »>
  <Rôle ID = « Visiteur» Objectifs = « Consulter_Info_Générale, S'abonner, ... » />
  <Profil ID = « Expert » Description = « Connaissance_Internet (très élevée) ET Pratique_du_Système (Bonne) » />
</Catégorie_Utilisateurs>

```

Figure 3.5 – Exemples de Représentation de Catégories d'Utilisateurs dans AWIS-M

Malgré la finesse introduite dans la description des profils, une approche basée sur les stéréotypes qui sous-tendent la catégorisation des utilisateurs est, en fait, proposée par Gnaho. Une conséquence directe de la définition de Catégories d'Utilisateur est que l'adaptation à l'utilisateur dans un SIW conçu avec AWIS-M se fait sur la base de l'appartenance de cet utilisateur à une catégorie prédéfinie. Il s'agit donc d'adapter à un groupe plus qu'à un utilisateur individuel.

c – Le modèle du Domaine d'Information Web

Le troisième modèle conceptuel décrit le *Domaine d'Information Web*. Ce modèle fait apparaître les concepts du domaine manipulés par le SIW ainsi que leurs relations. Il est exprimé par un modèle objet.

d – Le modèle conceptuel de Navigation

Gnaho propose un *modèle conceptuel de Navigation*, dont le concept central est celui d'*Unité Sémantique de Navigation (USN)*. Une USN est définie comme *un ensemble de procédés de navigation prédéfinis, traduisant l'opérationnalisation d'un Objectif Utilisateur*.

Une USN est un agrégat de sous-unités de navigation appelées *Méthodes de Navigation*. Chaque Méthode de Navigation est construite en fonction d'un Profil Utilisateur et décrit comment réaliser l'Objectif Utilisateur pour ce profil particulier. Elle est composée d'une *Structure de Navigation* et d'un *Modèle de Processus de Navigation*.

Une Structure de Navigation est un réseau de *Nœud de Navigation* et de *Liens de Navigation*¹⁶. Un Nœud de Navigation est défini par un ensemble d'*Éléments Atomiques de Navigation* qui véhiculent une information (issue du Modèle du Domaine ou non) ou sont liés à une action. L'auteur distingue les *Nœuds de Navigation Fixes* et les *Nœuds de Navigation Adaptables*. Dans les premiers, les *Éléments Atomiques de Navigation* sont invariables : le contenu du nœud est identique pour tous les utilisateurs. Dans les seconds, l'adaptation est

¹⁶ Les liens, assez classiques dans AWIS-M, ne sont pas détaillés. Voir [Gnah00] pour une description.

permise. Pour cela, une *Matrice d'Adaptation* est définie. Cette matrice exprime le *degré de compatibilité (DC)* de chaque Élément Atomique de Navigation (EAN) avec un paramètre de Profil Utilisateur. Ainsi à tout couple (EAN, paramètre) est associée une valeur DC telle que, si l'utilisateur possède pour ce paramètre de profil une valeur égale ou appartenant à DC (si DC est un intervalle), alors l'EAN est jugé intéressant pour cet utilisateur.

Le second élément de description d'une Méthode de Navigation est le *Modèle de Processus de Navigation*. Cette notion a pour but de guider l'utilisateur dans la navigation lui permettant d'atteindre son objectif. Nous rappelons qu'une Méthode de Navigation est définie pour un Profil Utilisateur et pour un Objectif d'Utilisateur donnés : un tel processus l'est également. Le Modèle de Processus de Navigation est décrit par des *Directives de Navigation*. La *Signature* d'une directive définit les conditions de son application, en faisant référence au Nœud de Navigation et à l'Objectif Utilisateur auxquels elle correspond. Le *Corps* de la directive contient les informations de guidage, en proposant, par exemple, différentes alternatives et des critères de choix. Sur la base de ces informations, l'utilisateur sélectionne l'option qui lui convient.

3.5.2.2 Les modèles de produits de niveau logique

Les modèles produits de niveau logique proposés par AWIS-M sont le *Modèle Logique des Utilisateurs*, le *Modèle de Présentation de l'Hyperespace* et le *Modèle Mathématique de l'Hyperespace*.

Nous ne présentons pas en détail le *Modèle de Présentation de l'Hyperespace*, celui-ci consistant essentiellement, par le biais de descriptions en XML et XSL :

- à définir la structure et les caractéristiques logiques des éléments décrits au niveau conceptuel par le Modèle conceptuel de Navigation et,
- à spécifier des règles de mises en formes des différents éléments.

Nous préférons nous concentrer sur les deux autres modèles, ceux-ci constituant à notre avis les contributions majeures de la méthode.

a – Le Modèle Logique des Utilisateurs

Le *Modèle Logique des Utilisateurs* offre une représentation logique des trois concepts décrits au niveau conceptuel. La notion de *L-Rôle* (L pour logique) fait référence à un ensemble de *L-USN* qui décrivent au niveau logique la notion d'Unité Sémantique de Navigation présentée au niveau conceptuel (et qui traduit un Objectif Utilisateur en termes de navigation). Globalement, cet ensemble de L-USN définit la partie de l'Hyperespace qui permet de répondre aux utilisateurs jouant ce rôle.

Le concept de *L-Profil* permet de quantifier les valeurs linguistiques utilisées pour caractériser les paramètres de Profil Utilisateur au niveau conceptuel. Ce concept repose sur la théorie des sous-ensembles flous et la logique floue [Zade75][Tong95]. Dans un premier temps, chaque valeur linguistique caractérisant un paramètre de profil conceptuel est associée à un ensemble flou défini sur un ensemble de valeurs définissant un référentiel au moyen d'une *fonction d'appartenance*. Il en résulte une représentation logique du paramètre telle que celle illustrée par la Figure 3.6 pour le paramètre « Age » (d'après [Gnah00]).

Les courbes représentent les fonctions d'appartenance définies pour chaque valeur linguistique caractérisant le paramètre de Profil « Age » au niveau conceptuel. Ces fonctions décrivent un ensemble flou de valeurs sur l'ensemble référentiel. Ainsi, pour un utilisateur âgé de 25 ans, son *degré d'appartenance* est de 0,4 pour les sous-ensembles flous « Non Jeune » et

« Très Jeune », de 0,75 pour le sous-ensemble flou « Jeune » et de 0,80 pour le sous-ensemble flou « Plus ou moins Jeune ».

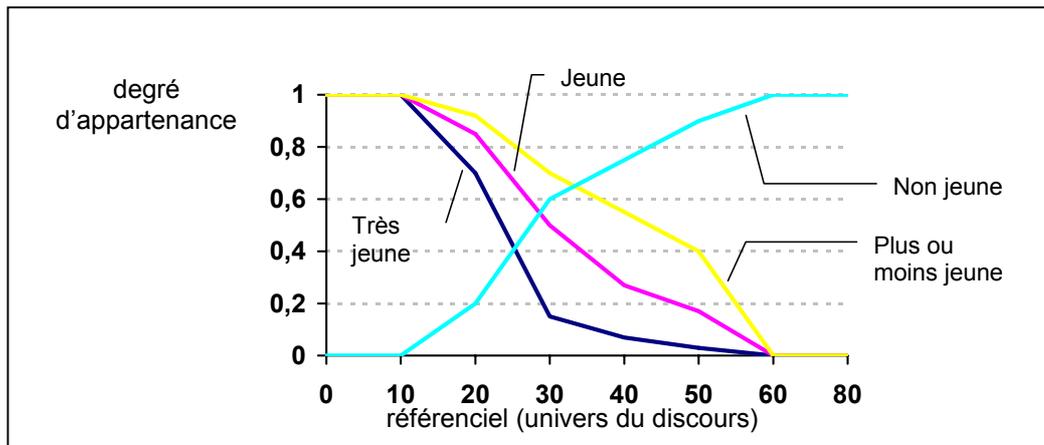


Figure 3.6 – Expression logique d'un paramètre de profil dans AWIS-M

Un L-Profil est défini comme une agrégation de paramètres. Il peut en effet prendre en compte plusieurs paramètres à l'image des profils conceptuels. Nous proposons dans la Figure 3.7 un exemple d'un même profil décrit au niveau conceptuel et au niveau logique. Nous expliquons ensuite les aspects principaux de la spécification logique¹⁷.

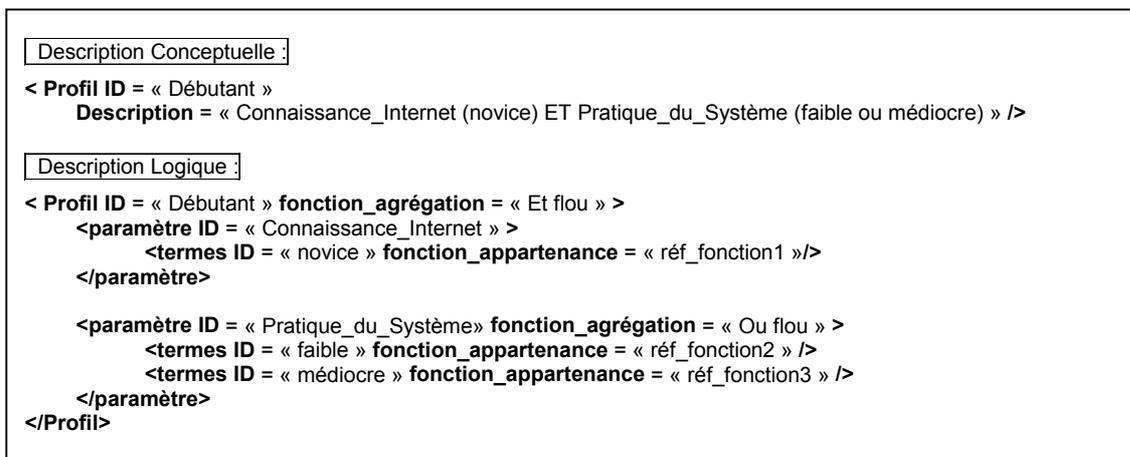


Figure 3.7 – Descriptions conceptuelle et logique d'un profil dans AWIS-M

Les *fonctions d'agrégation* expriment la façon dont il faut considérer les termes ou les paramètres dans la définition du profil. Lorsqu'il existe plusieurs termes dans un paramètre (cf. paramètre *Pratique_du_Système*), la *fonction d'agrégation de termes* indique qu'il faut retenir le

¹⁷ Nous proposons un exemple que nous avons créé dans la mesure où celui présenté par Gnaho ([Gnah00], p 142) nous paraît erroné.

cas le plus notable. Basée sur le « Ou flou »¹⁸, elle implique de retenir le maximum des degrés d'appartenance obtenus pour les termes par application de fonctions d'appartenance vues plus haut.

Une *fonction d'agrégation de paramètres*, déclarée au niveau du profil, établie une règle similaire entre les profils. Ici basée sur le « Et flou », elle spécifie qu'on retiendra le minimum entre les valeurs (i.e. les degrés d'appartenance) obtenues pour chaque paramètre.

Les L-Profiles sont donc des spécifications qui permettent de quantifier le Profil d'un utilisateur. Supposons que les fonctions d'appartenance utilisées dans cet exemple aient donné, pour l'évaluation d'un utilisateur particulier, 0,7 pour le terme *novice*, 0,5 pour *faible* et 0,8 pour *médiocre*. On obtient pour l'utilisateur en question :

- pour le paramètre *Connaissances Internet (novice)*, un degré d'appartenance de 0,7
- pour le paramètre *Pratique du Système (faible ou médiocre)*, un degré d'appartenance de 0,8 (maximum de 0,5 et de 0,8)
- pour le profil *Débutant*, un degré d'appartenance de 0,7 (minimum de 0,7 et de 0,8). C'est sur la base de cette valeur qu'il sera décidé si l'utilisateur appartient ou non à ce profil. Notons toutefois que dans [Gnah00], rien n'est précisé à propos de la décision d'associer un utilisateur à une catégorie plutôt qu'à une autre. On peut se poser la question du choix à faire en cas d'égalité des degrés d'appartenance à deux profils différents.

Nous terminons la présentation du modèle logique des utilisateurs par la définition des *L-Catégories d'Utilisateurs*. Celles-ci correspondent à une association d'un *L-Rôle* et d'un *L-Profil* de façon similaire à ce qui est décrit au niveau conceptuel. Nous notons que dans la description d'une L-Catégorie d'Utilisateur (en XML), un L-Profil est entièrement spécifié (i.e. de la même façon que ce que nous avons montré dans la Figure 3.7). Or, un même L-Profil pouvant être exploité par plusieurs L-Catégories, ce choix nous paraît contestable. Une approche basée sur des importations de spécifications indépendantes des L-Profiles permettrait de bénéficier des avantages des techniques de réutilisation (réduction du nombre de spécifications à écrire, réduction du risque d'erreurs, etc.).

b – Le Modèle Mathématique de l'Hyperespace

Le second modèle de niveau logique que nous abordons est le *modèle Mathématique de l'Hyperespace*. L'objectif de ce modèle est de fournir des mécanismes permettant de calculer l'Hyperespace à présenter à un utilisateur en fonction de la Catégorie d'Utilisateur à laquelle il appartient.

Ce modèle est basé sur des notions mathématiques telles que les *familles de parties* et les *clans engendrés par une famille de parties*. Il nous est difficile de présenter de façon concise les fondements mathématiques de ces notions ainsi que leur application à la gestion d'un Hyperespace. Nous renvoyons le lecteur qui souhaiterait une description de ces aspects à [Gnah00]. Nous présentons simplement les principes qui sous-tendent ce modèle. Notre objectif est surtout de mettre en évidence comment cette proposition traite l'adaptation à l'utilisateur.

Un Hyperespace est la construction définie par le modèle logique de présentation de l'Hyperespace. Cette construction présente différents niveaux de granularité. Au plus haut

¹⁸ Plusieurs opérateurs sont possibles pour définir une fonction d'agrégation (union, intersection, moyenne, etc., [Gnah00]). Tout dépend des ensembles flous considérés et de la sémantique que l'on souhaite donner à cette agrégation.

niveau, l'Hyperespace est composé des descriptions logiques des Unités Sémantiques de Navigation (L-USN). Chaque L-USN est elle-même constituée de Nœuds de Navigation Logiques (L-NN). Les L-NN correspondent au second niveau de granularité. Enfin, les L-NN sont construits à partir d'Éléments de Base Logique (L-EB) qui font référence au troisième niveau de granularité.

Cette structure est étroitement reliée aux concepts du Modèle Logique des Utilisateurs. C'est en spécifiant comment sont exploitées ces relations que le Modèle Mathématique autorise la construction dynamique d'un Hyperespace adapté à une catégorie d'utilisateurs. Ainsi, par la prise en compte du L-Rôle, le système sélectionne les L-USN qui correspondent aux objectifs de la Catégorie d'Utilisateurs. Chaque L-USN offre potentiellement plusieurs L-NN. L'exploitation du L-Profil permet ici au système de ne retenir que ceux qui sont pertinents. A l'étape suivante, le degré d'appartenance de l'utilisateur à un L-Profil permet d'affiner l'adaptation : à partir de la *matrice d'adaptation*, seuls les L-EB pour lesquels le *degré de compatibilité* est satisfait sont présentés (cf. page 46). Cette dernière étape est le seul cas où la proposition dépasse le cadre d'une adaptation à une catégorie d'utilisateur pour proposer une adaptation qui soit plus individuelle.

3.5.3 La démarche proposée

La démarche méthodologique proposée dans AWIS-M comporte trois phases au cours desquelles sont conduites différentes étapes.

La première phase est *conceptuelle*. Les étapes de cette phase consistent à décrire les quatre modèles conceptuels que nous avons présentés : modèles des Utilisateurs, des Objectifs Utilisateurs, du domaine et de la navigation.

La phase *logique* est alors engagée avec la transformation des précédents modèles en leurs équivalents logiques : modèle logique des utilisateurs, modèle de présentation de l'Hyperespace et modèle mathématique de l'Hyperespace. Notons que le modèle conceptuels des Objectifs Utilisateurs ne conduit pas à un modèle logique propre : l'expression logique des objectifs est traduite dans le modèle de présentation de l'Hyperespace via les Unités Sémantiques de Navigation logiques.

La troisième et dernière phase est celle de la *réalisation*. Elle correspond à l'implémentation du SIW qui se fait sur la base des spécifications de niveau logique.

L'approche AWIS-M a la particularité de proposer une démarche inspirée par l'ingénierie des processus. AWIS-M exploite le *méta-modèle de processus* développé dans le cadre du projet ESPRIT NATURE [Gros97]. Ce méta-modèle adopte une approche contextuelle pour guider l'ingénieur d'application. Il décrit différents concepts et relations entre eux que nous résumons. Une *situation* est définie par l'état actuel du produit (i.e. du modèle ou de la partie de modèle) en construction. L'*intention* est un objectif de développement (e.g. ce que l'on souhaite construire). L'association d'une situation et d'une intention décrit un *contexte*. A ce contexte, est associée une *action* qui mène à la modification adéquate du produit. Le méta-modèle permet de décrire une démarche en spécifiant l'ensemble des contextes qui guideront le comportement de l'ingénieur d'application. Les différents contextes sont reliés entre eux par des liens d'affinement et de composition ; une structure arborescente est utilisée pour représenter ces aspects. Les contextes entretiennent aussi des relations de précédence. Un *graphe de précédence* les décrit. Les liens du graphe permettent de spécifier la nature progressive et parfois ordonnée des différentes étapes de la démarche ; ils sont associés à des critères de choix qui conditionnent la progression dans le graphe (et donc la démarche).

La démarche AWIS-M est donc entièrement spécifiée au moyen d'instances du méta-modèle NATURE.

3.5.4 Conclusion sur AWIS-M

La proposition AWIS-M est caractérisée par la définition de modèles de produits, décrits aux niveaux conceptuel et logique, qui abordent la modélisation des utilisateurs, de leurs objectifs et de l'hypermédia (structure, navigation et présentation). L'ensemble de l'approche est guidée par les objectifs des utilisateurs : les spécifications en terme de navigation visent à guider l'utilisateur dans l'atteinte de ses objectifs.

Parmi les modèles de produits proposés, on retient l'effort particulier consacré à modéliser les catégories d'utilisateur. Celles-ci sont décrites comme des associations entre un rôle (i.e. un ensemble d'objectifs) et un profil. Dans AWIS-M, la notion de profil repose sur une approche originale exploitant la logique floue. L'appartenance des utilisateurs à une catégorie est évaluée par quantification floue (*fuzzy quantification* ou *fuzzyfication*) du profil associé à cette catégorie. On note cependant que la proposition n'indique pas clairement comment procéder à la classification des utilisateurs.

Le modèle mathématique de l'Hypermédia décrit comment un hypermédia est construit dynamiquement et de façon adaptée en fonction d'une catégorie d'utilisateur. Les principes de ce modèle reposent sur l'exploitation, formalisée mathématiquement, du rôle et du profil spécifiques à une catégorie d'utilisateur. La notion de degré d'appartenance d'un utilisateur à une catégorie permet, en outre, d'adapter dans une certaine mesure le contenu à cet utilisateur particulier.

AWIS-M se distingue par ailleurs par une formalisation relativement détaillée d'une démarche de développement contextuelle. Elle s'appuie pour cela sur l'instanciation du méta-modèle de processus de NATURE [Gros97].

L'architecture d'un outil associé à la proposition est décrite dans [Gnah00]. L'outil, baptisé AWIS-CASE, est notamment destiné aux ingénieurs d'applications Web. Dans sa version actuelle, il offre un module permettant la spécification graphique des schémas conceptuels de navigation (*AWIS Navigational Conceptual Schema Editor*). Un guide méthodologique électronique est également disponible (*AWIS-M Electronic Handbook*) afin d'aider l'ingénieur d'application à comprendre et utiliser la démarche proposée. Ce guide a été développé comme exemple applicatif de la proposition de Gnaho [Gnah00].

3.6. UML-based Web Engineering – UWE

3.6.1 Aperçu général d'UWE

UWE (*UML-based Web Engineering*)¹⁹ [Koch00a] est une approche pour le développement d'applications hypermédiées basées sur le Web. Cette approche couvre tout le cycle de développement des applications à travers cinq phases: la création (*inception* en anglais), l'élaboration, la construction, la transition et la maintenance. Les quatre premières phases reprennent et adaptent celles décrites dans le Processus Unifié (PU) [Jaco99] duquel s'inspire pour beaucoup le processus proposé dans UWE. La cinquième phase étend le PU par la prise en compte d'activités liées à la maintenance d'un système qui, après sa mise en service, nécessite

¹⁹ Dans [Koch00a], l'approche est désignée par l'acronyme UPHD pour Unified Process-based Hypermedia Systems Development. L'auteur, dans ses travaux suivants, emploie l'acronyme UWE que nous utiliserons également dans cette présentation.

divers ajustements portant sur le contenu, la présentation ou encore la structure de l'hypermédia. Dans le cadre de cet état de l'art, nous ne décrivons pas la totalité du processus, mais nous axons l'étude d'UWE sur les propositions relatives à l'analyse et à la conception de l'application. Nous mentionnons également que l'approche repose sur un méta-modèle, appelé *Munich Reference Model* [Koch00], qui formalise les caractéristiques et les principes de fonctionnement des applications hypermédias adaptatives. La description de ce méta-modèle n'est toutefois pas indispensable à la compréhension de la proposition décrite dans cet état de l'art. Nous renvoyons le lecteur intéressé par la description du méta-modèle et par celle du processus de développement complet à [Koch00].

UWE propose une notation et une méthode [Koch00]. La notation proposée offre des éléments de modélisation qui combinent les notations d'UML [UML99] et des extensions spécifiques décrites par un profil UML²⁰ [Baum99][Henn00]. La méthode décrit les étapes nécessaires à la construction de modèles représentant différents aspects de l'hypermédia. A travers un processus de modélisation itératif et incrémental, la méthode permet l'analyse des besoins, la modélisation conceptuelle, la conception de la navigation et celle de la présentation, ainsi que la modélisation des utilisateurs et des adaptations.

Les dimensions abordées donnent respectivement lieu à six modèles, représentés par des paquetages à leur nom dans la Figure 3.8. Les relations de dépendance entre les packages expriment la traçabilité existant entre les processus permettant d'aboutir à ces modèles.

Nous décrivons dans ce qui suit les différentes étapes constituant le processus d'analyse et de conception d'UWE. Pour chacune, nous présentons son objectif, les éléments de modélisation introduits et un résumé de la démarche préconisée pour produire le modèle attendu à l'issue de l'étape.

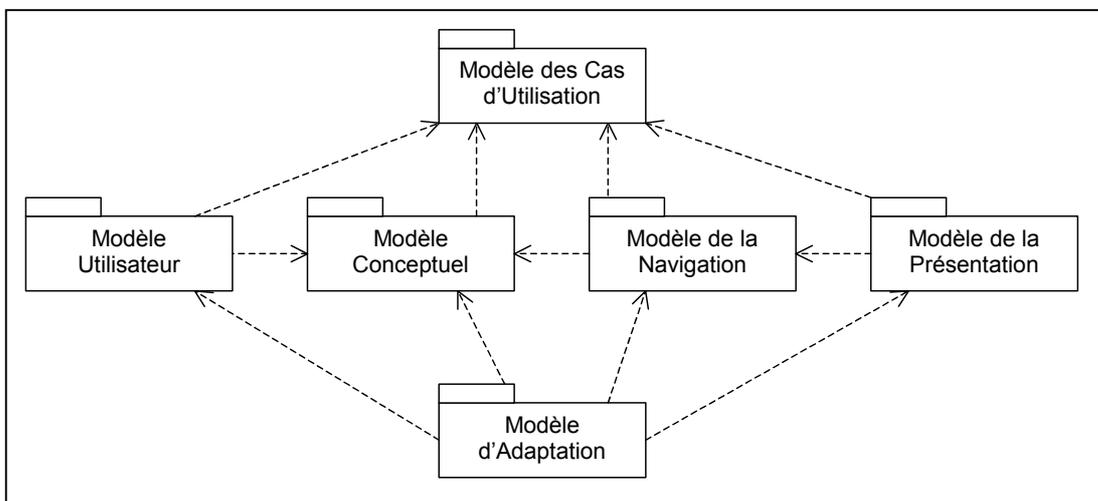


Figure 3.8 – Les modèles construits selon la méthode d'analyse et de conception de UWE (Traduit de [Koch01]).

3.6.2 Analyse des besoins

L'analyse des besoins dans UWE suit une approche basée sur les cas d'utilisation. Elle consiste à identifier les besoins fonctionnels des différents utilisateurs auxquels doit répondre l'application.

²⁰ Nous emploierons désormais l'expression « profil UWE » pour désigner ce profil UML.

Les éléments de modélisation utilisés à cette étape sont ceux proposés par UML. Les notions d'*acteurs*, de *cas d'utilisation*, de relations d'*héritage*, d'*utilisation* et d'*extension* entre eux, ainsi que les mécanismes de *paquetages* et de *vues* sont repris et conservent la sémantique décrite par UML. Les notations graphiques sont également celles d'UML.

Lors de cette étape, la démarche suit les propositions communes à plusieurs processus guidés par les cas d'utilisation tels que dans [Jaco99] et [Kruc00]. Il s'agit globalement d'identifier les acteurs et, pour chacun, les activités qu'il entreprend. Ces activités sont exprimées par des cas d'utilisation auxquels sont reliés les acteurs. Si nécessaire, des relations d'utilisation et d'extension sont définies entre certains cas d'utilisation, ainsi que des relations d'héritage entre acteurs et/ou entre cas d'utilisation. Chaque cas d'utilisation peut également bénéficier d'une description détaillée.

3.6.3 Modélisation Conceptuelle

Cette étape vise à établir le modèle conceptuel du domaine d'application en tenant compte des besoins identifiés, au préalable, par les cas d'utilisation. Un diagramme de classes UML est décrit. Il organise les concepts du domaine sur la base des éléments UML suivants : les classes, les associations et éventuellement les paquetages. Si ces deux derniers éléments sont exactement ceux d'UML, les classes du modèle conceptuel d'UWE sont, quant à elles, dotées d'un compartiment optionnel qui s'ajoute à celui des attributs et des opérations. Ce compartiment est destiné à recevoir des *variantes* qui constituent des informations supplémentaires utilisées pour adapter le contenu véhiculé par la classe en fonction du profil utilisateur.

La démarche préconisée est, à cette étape encore, similaire à ce qui est proposé par les techniques de modélisation orientées objet. Elle consiste à trouver les classes et à spécifier pour chacune ses attributs et opérations, relier les classes par des associations et définir d'éventuelles contraintes. Ces dernières sont exprimées en OCL (*Object Constraint Language*) et visent à augmenter la précision du modèle.

3.6.4 Modélisation de la Navigation

La modélisation de la navigation comprend en fait deux étapes, conduisant à la construction de deux modèles. D'une part, un modèle de l'*espace de navigation* permet de décrire *quels* objets l'application permet d'atteindre par navigation. D'autre part, le modèle de la *structure de navigation* exprime *comment* ces objets sont atteints (*i.e.* comment on y accède dans l'application). Nous étudions successivement ces deux modèles et les démarches associées.

3.6.4.1 Modélisation de l'espace de navigation

L'espace de navigation est modélisé grâce à deux éléments introduits par le profil UWE : les *classes navigation* et les *associations navigation*.

Une classe navigation représente une classe dont les instances sont accédées par l'utilisateur lorsqu'il navigue dans l'hypermédia. Elle correspond à une classe du modèle conceptuel dont elle reprend le nom. Les attributs d'une classe navigation sont ceux de la classe conceptuelle correspondante. Il est toutefois possible d'omettre certains attributs de la classe conceptuelle lorsqu'ils sont jugés non pertinents pour le modèle de navigation (*i.e.* s'il n'est pas utile de les rendre visibles dans l'hypermédia). Il est également possible de doter la classe navigation d'attributs qui n'appartiennent pas à la classe conceptuelle correspondante. Ces attributs supplémentaires sont obligatoirement des *attributs dérivés*²¹ à partir d'une classe du modèle

²¹ au sens donné par UML (le nom de ces attributs est précédé par un *slash (/)* dans la représentation graphique).

conceptuel qui n'apparaît pas dans le modèle de navigation. Ils sont calculés au moyen d'une expression OCL.

Une association navigation exprime une possibilité d'accès direct à une classe navigation *cible* à partir d'une classe navigation *source*. Elle est représentée graphiquement par une flèche orientée et stéréotypée par l'expression « *direct navigability* ». Le sens de la flèche indique la direction de l'association navigation, qui est dite uni-directionnelle. Si l'association navigation peut être parcourue dans les deux sens, les deux extrémités portent une flèche. On parle d'association navigation bi-directionnelle. Toute extrémité fléchée d'une association navigation est dotée d'un nom de rôle et d'une multiplicité qui expriment respectivement le rôle de la classe navigation cible (i.e. ce qu'on va trouver en suivant l'association navigation) et le nombre d'instances de cette classe navigation accessibles par ce biais.

Les associations du modèle conceptuel sont toutes transformées en associations navigation dès lors que les classes du modèle conceptuel qu'elles relient ont des classes navigation correspondantes dans le modèle de l'espace de navigation. Des associations navigations peuvent également être créées pour offrir un accès plus direct à certaines informations. Bien que cela ne soit pas mentionné dans [Koch01], l'ajout d'une association navigation qui n'est pas en correspondance directe avec une association du modèle conceptuel nécessite d'exprimer la sémantique associée, ainsi que la façon d'obtenir les instances de la classe navigation cible à afficher. Des contraintes en OCL permettent de spécifier des restrictions sur le modèle de l'espace de navigation (par exemple, le parcours d'une association navigation peut ne pas aboutir à la visualisation du même ensemble d'instances en fonction de qui est l'utilisateur, de contraintes temporelles, matérielles, etc.).

Le modèle de l'espace de navigation prend donc la forme d'un diagramme de classes UML particulier car faisant apparaître des éléments stéréotypés de deux types. Les classes navigation et associations navigation donnent une première représentation de l'hypermédia (à ce stade, les classes navigation sont le pendant des nœuds et les associations navigation, celui des liens de l'hypermédia). Les associations navigation sont vues, dans le modèle de l'espace de navigation d'UWE, comme une *possibilité de navigation* (*navigability* en anglais). La façon dont cette navigation est réellement supportée dans l'hypermédia est décrite par le modèle de la structure de navigation que nous décrivons à présent.

3.6.4.2 Modélisation de la structure de navigation

Cette étape vise à produire un modèle qui concrétise les possibilités de navigation exprimées dans le modèle de l'espace de navigation. Des éléments de modélisation spécifiques, appelés *primitives d'accès*, sont définis pour construire le modèle de la structure de navigation à partir du modèle précédent. Trois étapes sont proposées par la démarche. Dans un premier temps, le modèle de navigation est augmenté par l'introduction de trois primitives d'accès, que sont les index, visites guidées et requêtes. Les deux premières structures correspondent aux notions déjà utilisées dans des travaux tels que HDM, RMM, etc. La primitive d'accès introduite par UWE permet d'exprimer une requête en OCL. Ces primitives d'accès sont modélisées par des classes stéréotypées « *index* », « *guided tour* » et « *query* ».

Ces trois primitives d'accès sont utilisées pour augmenter le modèle de l'espace de navigation obtenu à l'étape précédente. La démarche spécifiée par UWE pour procéder à cette opération est résumée dans ce qui suit. Elle consiste principalement à suivre des règles de transformation du modèle initial. Afin d'en faciliter la compréhension, nous proposons une illustration graphique pour chaque transformation mentionnée (cf. Figure 3.9), les numéros mentionnés devant les paragraphes ci-dessous permettent de se référer à la représentation de

même numéro dans le tableau). Ces règles nous apparaissent imprécises sur certains aspects. Nous commentons donc ces règles et donnons notre interprétation sur les points soulevés.

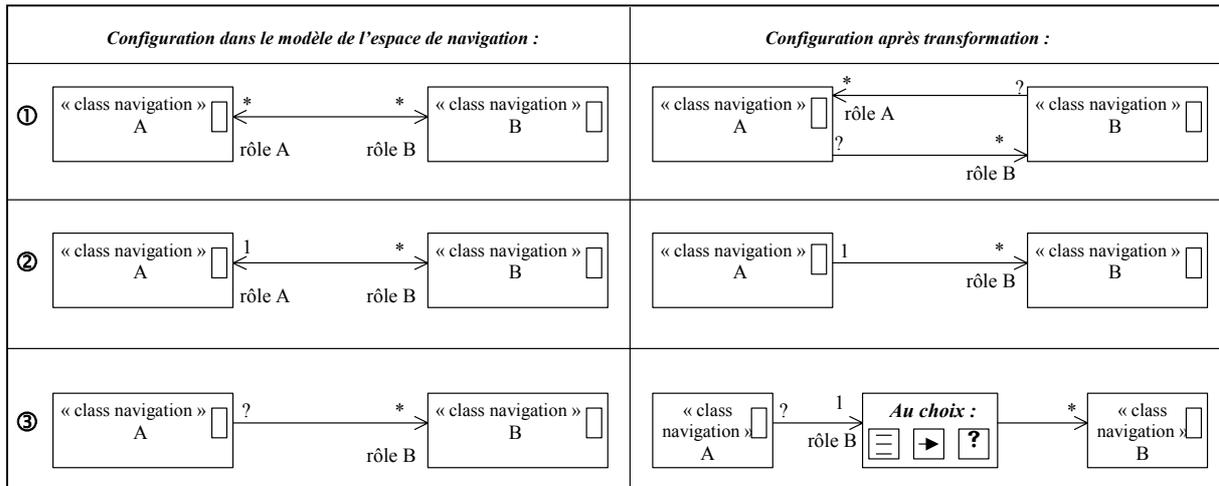


Figure 3.9 – Du modèle de l'espace de navigation au modèle de structure de la navigation selon UWE
 Il s'agit de l'interprétation (graphique) que nous donnons des règles de transformation permettant ce passage. Les multiplicités représentées par un '?' traduisent le fait que UWE ne précise rien à leur sujet. Pour la configuration 3, à droite, les symboles reproduits dans le cadre 'Au choix' sont les icônes associées aux classes index, visite guidée et requête (de gauche à droite)

① Règle : remplacer les associations navigation bi-directionnelles dont la multiplicité à chacune des deux extrémités est supérieure à 1 par deux associations navigation unidirectionnelles correspondantes.

Commentaire : La question qui se pose ici est de déterminer la multiplicité à associer à la classe navigation source de chacune des deux associations navigation unidirectionnelles créées. Nous avons en effet noté qu'une association navigation bi-directionnelle ne donne aucune information à ce propos. Nous proposons que l'affectation de cette multiplicité soit faite en revenant au modèle conceptuel et en observant la sémantique associée à l'association (conceptuelle) initiale qui lie les classes (conceptuelles) desquelles ont été dérivées les classes navigation en question.

② Règle remplacer les associations navigation bi-directionnelles dont la multiplicité à l'une des deux extrémités est supérieure à 1 par une association navigation unidirectionnelle orientée vers l'extrémité de multiplicité supérieure à 1. Le lien de navigation qui disparaît sera rétabli lors d'une phase ultérieure de la conception.

Commentaire : Bien que les auteurs indiquent le rétablissement ultérieur du lien d'association qui est supprimé, nous n'avons rien trouvé à ce propos dans leurs descriptions. Le processus appliqué sur ce point nous aurait notamment permis de lever le voile sur le traitement du cas d'un lien de multiplicité supérieure à la source. Cette question est de nouveau soulevée dans la règle suivante :

③ Règle : pour les associations du modèle de l'espace de navigation ayant une multiplicité supérieure à 1 du côté de l'extrémité qui porte la flèche, choisir une ou plusieurs primitive(s) d'accès pour concrétiser la navigation. Les primitives d'accès sont insérées entre la classe navigation source et la classe navigation cible et le nom de rôle migre au niveau de l'élément représentant la primitive.

A ce stade, le modèle de la structure de la navigation est constitué des classes navigation, associations navigation et primitives d'accès que nous avons décrites. Ce modèle subit une transformation supplémentaire qui vise à ajouter un quatrième type d'accès appelé *menu*. Un menu est une instance d'une classe stéréotypée « *menu* » qui est liée à une classe appelée *MenuItem* par une composition. Les Menus permettent un accès direct aux instances de classes de types hétérogènes (alors que les index offrent un accès à des instances d'un seul type de classe, les classes navigation).

La démarche proposée pour introduire des menus au modèle de la structure de la navigation obtenu à l'étape précédente consiste à associer un Menu à toute classe navigation source d'au moins une association navigation. L'association entre une telle classe navigation et un Menu est faite par le biais d'une composition. Les rôles précédemment portés par les associations navigation deviennent les items du menu et ce sont ces items qui font désormais le lien avec la cible (i.e. une classe navigation ou une primitive d'accès).

La dernière étape dans la modélisation de la structure de la navigation consiste à ajouter des propriétés aux différents éléments du modèle pour spécifier des contraintes en vue de l'adaptation. A titre d'exemple, la propriété *{direct guidance}* assignée à une visite guidée ou à un lien indique que le système devra décider quel est le meilleur nœud à présenter en fonction de l'utilisateur ; la propriété *{annotated}* affectée à un index ou un menu indique que des mécanismes d'annotation seront utilisés pour indiquer la pertinence de ces éléments pour un utilisateur donné ; les items de menus ou d'index peuvent également être caractérisés par une propriété *{removed}* afin de mentionner que le système peut les supprimer s'il estime qu'ils n'ont pas d'intérêt pour l'utilisateur.

La modélisation de la structure de la navigation est alors achevée. L'étape suivante consiste en la conception des aspects relatifs à la présentation dont nous résumons les principes.

3.6.5 Modélisation de la Présentation

Le modèle de la présentation est constitué d'un ensemble de vues qui montrent le contenu et la structure des nœuds de l'hypermédia et qui décrit la façon dont l'utilisateur interagit avec ces nœuds. Deux étapes sont proposées par UWE pour procéder à cette modélisation : la conception de *storyboards*, puis celle du flot de présentation.

La conception de *storyboards* repose sur trois types d'élément de modélisation décrits par le profil UWE²². Le premier élément est une classe stéréotypée « *UI View* » pour *User Interface View*. Une instance de cette classe (nous la désignerons par *VIU* pour *vue interface utilisateur*) est un conteneur qui permet de grouper des éléments qui ont comme caractéristique de devoir être tous présentés simultanément à l'utilisateur. Le second élément, une classe stéréotypée « *presentation class* », permet de diviser une VIU en différentes unités structurelles (chaque classe présentation est une unité structurelle liée à une VIU par une composition). Le troisième élément est une classe abstraite appelée *User Interface Element* (*EIU* pour nous dans la suite) qui est associée à une classe présentation par une composition. La classe *EIU* se spécialise en différentes classes stéréotypées telles que « *text* », « *image* », « *video* », « *audio* », « *form* » ou « *anchor* ». Ces deux dernières classes sont considérées comme des éléments pour l'interaction : la classe « *anchor* » est associée à un lien de navigation, la classe « *form* » permet la soumission d'information, le déclenchement d'événement etc. Chacun de ces éléments dispose d'une représentation graphique particulière que le lecteur intéressé pourra trouver dans [Koch01].

²² Voir [Koch00] pour les représentations graphiques associées.

Le modèle de storyboard est un diagramme de classes UML (utilisant les extensions du profil UWE). Son élaboration suit un ensemble d'étapes qui consistent globalement :

- à construire une classe présentation pour chaque nœud (classe navigation, index, menu, etc.) du modèle de structure de la navigation,
- si nécessaire, à ajouter des éléments d'interface ou des contraintes pour traduire des besoins fonctionnels non encore spécifiés lors des différentes étapes,
- à déterminer les classes présentation devant être affichées simultanément et à le spécifier au moyen d'une composition les liant à une classe VIU,
- à établir les liens des classes « *anchor* » vers les classes VIU adéquates.

Suite à la définition des storyboards, le modèle du flot de présentation est établi. Lors cette étape, le concepteur effectue des choix pour la présentation de l'information, en décidant notamment de recourir à du multi-fenêtrage ou à des frames, et en indiquant les modalités de remplacement du contenu dans ces espaces. Des éléments de modélisation sont, conformément à l'approche d'UWE, proposés sous la forme d'extension UML. Ces éléments permettent d'établir une correspondance avec les objets de présentation décrits dans les storyboards. Des scénarios, sous forme de diagrammes de séquence UML, complètent la spécification.

3.6.6 Modélisation des utilisateurs

Dans UWE, la modélisation des utilisateurs a pour but à la fois de déterminer les caractéristiques à retenir pour construire des profils utilisateur et d'établir les liens entre ces caractéristiques et leurs relations avec les éléments du modèle. Le modèle de l'utilisateur est représenté par un diagramme de classes. Une classe *Utilisateur* est reliée par des associations à d'autres classes (e.g. classe *Préférence*) permettant de caractériser l'utilisateur.

La construction du modèle utilisateur nécessite de trouver les caractéristiques pertinentes par rapport au domaine d'application en vue de l'adaptation. Par exemple, une classe *Type de Paiement* (qui indique le mode de paiement généralement choisi par l'utilisateur) est justifiée dans le cas d'une application de commerce électronique mais n'a pas forcément d'intérêt dans d'autres domaines. Pour chacune de ces caractéristiques, une classe est décrite dans le modèle utilisateur ; les associations avec les autres classes des modèles (utilisateur et du domaine) sont spécifiées, ainsi que d'éventuelles contraintes d'adaptation.

3.6.7 Modélisation de l'Adaptation

Le modèle d'adaptation de UWE est basé sur une notion de règle. Une règle est modélisée par une classe stéréotypée « *rule* » et liée par une composition à une classe « condition » et une classe « action ». Les conditions et les actions sont liées à des classes des modèles utilisateur ou du domaine. Des règles *d'adaptation* sont distinguées des règles *d'acquisition*. Les premières spécifient en fonction de quoi et comment doivent être adaptés contenu, navigation et/ou présentation. Les secondes décrivent comment acquérir des informations à propos de l'utilisateur et mettre à jour en conséquence son modèle. Dans ce but, une classe stéréotypée « *UserBehaviour* » est définie pour décrire les comportements observés par le système (tels que les actions de navigation ou de saisie de données). Les règles (conditions et actions) sont exprimées en langage naturel ou de façon plus formelle avec OCL. Par ailleurs, des diagrammes de collaboration sont construits pour mettre en évidence les enchaînements éventuels de règles.

La démarche associée à la création du modèle d'adaptation consiste à reprendre les spécifications insérées préalablement dans les différents modèles et à les exprimer sous forme de règles. Ainsi, pour les variantes de classes navigation, des règles sont créées pour définir les conditions selon lesquelles le contenu doit être adapté. De même, les propriétés ({direct

guidance}, {removed}, etc.) du modèle de la structure de la navigation sont traduites. Ou encore, des règles d'adaptation de la présentation sont décrites pour supporter les préférences des utilisateurs exprimées dans leur modèle. La démarche consiste également à définir les comportements des utilisateurs que l'on souhaite observer et à s'assurer que toute action supposée être détectée est bien utilisée dans la spécification d'au moins une condition de règle.

3.6.8 Conclusion sur UWE

UWE est un processus qui couvre tout le cycle de développement des applications hypermédias et basées sur le Web. Nous avons présenté la partie du processus dédié à l'analyse et à la conception des applications. Les caractéristiques majeures résident dans :

- le recours systématique aux notations et techniques d'UML à partir desquelles une extension est proposée sous la forme d'un profil spécifique aux dimensions navigation et présentation des applications hypermédias;
- un degré de précision certain dans la modélisation apporté par UML et ses extensions, par le recours à OCL pour l'expression de contraintes et par les directives guidant le processus de conception ;
- la définition d'un modèle utilisateur qui permet de les décrire individuellement ;
- la proposition d'un modèle d'adaptation basé sur des règles qui décrit comment adapter le contenu, la présentation et la navigation, comment détecter certains comportements de l'utilisateur et mettre à jour en conséquence le modèle utilisateur.

A ce jour, les auteurs souhaitent poursuivre leur travaux par l'extension de l'outil ArgoUML²³ afin d'y intégrer le profil UML qu'ils proposent. Un outil supportant le développement d'applications Web selon le processus UWE est envisagé. Les fonctionnalités de cet outil devraient permettre d'automatiser en partie les transformations entre les différents modèles ainsi que de procéder à de la génération de code (implémentation HTML ou JSP).

3.7. Web Modeling Language – WebML

3.7.1 Aperçu général de WebML

WebML (*Web Modeling Language*) [Ceri00a] est un langage de modélisation dédié à la conception de sites Web complexes. Ce langage est utilisé dans le cadre d'une approche de spécification par étapes successives. WebML s'inscrit dans la lignée des travaux menés précédemment par ses auteurs. Ces travaux sont d'une part, HDM-Lite [Frat98], un modèle qui adapte les propositions de HDM au contexte spécifique du Web, et d'autre part, l'évolution de HDM-Lite appelée W3I3 (Web-based Intelligent Information Infrastructure) [Ceri99] qui introduit les concepts de dérivation et de personnalisation. WebML reprenant en grande partie les concepts et approches de ces deux propositions nous faisons l'impasse sur leur présentation.

WebML propose un ensemble de représentations graphiques associées à une syntaxe XML pour la description d'un site. Plusieurs perspectives orthogonales sont abordées lors de cette description. Selon les articles sur WebML que nous avons étudiés [Ceri00a][Ceri00b] [Ceri01][Ceri02], le nombre de ces dimensions varie. Dans le cadre de cet état de l'art, nous faisons la synthèse de toutes les dimensions mentionnées par l'approche WebML. Bien qu'évoquées en tant qu'axes de modélisation à part entière, certaines d'entre elles cependant sont peu documentées et ne donnent pas lieu à un modèle spécifique.

²³ ArgoUML (2001). <http://www.tigris.org>

Nous présentons, dans un premier temps, les dimensions que nous estimons être à la base de l'approche en raison de leur présence constante dans la littérature sur WebML. Celles-ci aboutissent aux modèles *structurel*, de *dérivation*, de *composition*, et de *navigation*. Les deux derniers modèles constituent le modèle *hypertexte* qui définit la topologie de l'hypertexte. Des extensions apportées aux modèles de composition et de navigation permettent d'aborder la gestion des contenus qui constitue une quatrième dimension relativement bien décrite dans WebML. Nous abordons ensuite deux axes de conception que les auteurs mentionnent également sans toutefois bien les détailler : l'axe relatif à la *présentation* qui permet de définir l'apparence des pages du site, et celui relatif à la personnalisation dédié à l'adaptation du site à ses utilisateurs.

3.7.2 Le modèle structurel

Le *modèle structurel* permet de décrire le contenu géré par le site. Les éléments du modèle sont les *entités* et les *relations* qui dénotent des connexions sémantiques entre entités. Une entité a des propriétés appelées *attributs* qui sont typées et nommées. Les entités peuvent être organisées en hiérarchies (l'héritage multiple n'est pas autorisé). Les instances d'entités ont toutes un identifiant unique (OID) représenté par un attribut implicite de l'entité. Les relations sont binaires et ne possèdent pas d'attribut. Elles expriment le rôle d'une entité destination pour une entité source et peuvent être contraintes par une cardinalité. WebML impose que pour toute relation traduisant un rôle d'une entité A vers une entité B, la relation inverse soit spécifiée (i.e. une relation exprimant le rôle de B pour A et la cardinalité associée).

Dans les articles portant sur WebML, les représentations du modèle structurel s'appuient sur un paradigme Entité/Relation. Les auteurs indiquent toutefois qu'une approche objet peut également être utilisée, sous réserve de satisfaire aux contraintes du modèle structurel. Ces contraintes sont, selon nous, directement dictées par le fait que WebML propose une formalisation XML de ses concepts qui s'appuie sur l'utilisation de DTD (*Document Type Definition*).

3.7.3 Le modèle de dérivation

Le modèle de *dérivation* est obtenu à partir du modèle structurel. La dérivation pratiquée consiste à ajouter de l'information au schéma structurel afin d'en augmenter le pouvoir d'expression, et à définir différentes vues ou regroupements des données. La dérivation est exprimée au moyen de requêtes qui s'appliquent aux éléments du modèle structurel. Ces requêtes permettent de construire des concepts supplémentaires dont le contenu est défini par intention. Les dérivations sont principalement utilisées pour importer des attributs d'une entité dans une autre (lorsque les deux entités sont liées par une relation), définir des attributs calculés, créer des entités dérivées regroupant des objets aux propriétés communes, etc. Un langage de requêtes, baptisé *WebML-OQL*, est défini pour cela. Il est inspiré des expressions de chemins OQL.

3.7.4 Le modèle de composition

Le modèle de composition est utilisé pour déterminer les pages qui constituent le site Web et pour définir ce que chaque page contient en termes d'informations. Dans cette optique, le modèle offre une entité graphique symbolisant une page de l'hypermédia et différentes unités de description du contenu informationnel, appelées unités de contenu (*content units*) ou simplement unités (*units*). Ces unités offrent différentes possibilités pour intégrer du contenu dans les pages (e.g. présentation instance par instance, listes d'instances, sorte de visite guidée, etc.). Nous présentons en Annexe 1 les six unités de contenus définis dans WebML. Pour chacune, nous reproduisons sa représentation graphique, décrivons les principes de son utilisation et donnons sa syntaxe en XML.

3.7.5 *Le modèle de navigation*

Le *modèle de navigation* décrit la façon dont les unités et les pages du modèle de composition sont liées pour former un hypertexte. Ce modèle s'appuie sur la notion de lien déclinée en deux versions :

- les *liens contextuels* permettent d'établir, entre les unités, des connections qui reflètent la sémantique du schéma de l'application (i.e. du modèle structurel). Un lien contextuel porte une information (appelée *contexte*) d'une unité source vers une unité destination. Il est utilisé pour déterminer l'(ensemble d') objet(s) à montrer dans l'unité destination. Le contexte porté par un lien contextuel dépend du type de l'unité source : par exemple, pour une Data Unit, le contexte est l'identifiant de l'instance montrée dans cette unité au moment de l'activation du lien ; pour une Filter Unit, il s'agit des valeurs d'attributs saisies par l'utilisateur dans le formulaire associé. Graphiquement, les liens contextuels relient des unités, qu'elles soient ou non des composants d'une même page.
- les *liens non-contextuels* constituent un moyen de relier librement des pages, c'est-à-dire indépendamment des unités qu'elles contiennent et de la sémantique associée aux concepts véhiculés par les unités. Par exemple, une page quelconque du site est reliée à la page d'accueil de celui-ci par un lien non contextuel : il n'existe pas de contexte, au sens entendu précédemment, nécessaire à cette connexion. La représentation graphique des liens non-contextuels relie les frontières de deux pages.

3.7.6 *Extensions apportées aux modèles de composition et de navigation*

Des concepts pour la *saisie de données* et les *opérations* ont été ajoutés à WebML [Ceri00b] afin de supporter des modes d'utilisation autres que la consultation des informations. Ces concepts permettent de décrire au niveau conceptuel les actions telles que l'ajout, la modification, la suppression des données, ou encore l'invocation de différentes opérations (aussi bien propres à l'application qu'externes). Ces concepts consistent en des extensions apportées aux modèles de composition et de navigation. Le modèle de composition est enrichi de deux nouveaux types d'unités appelées Data Entry Unit et Operation Unit. Le modèle de navigation propose de nouveaux liens spécifiques à ces entités.

Une Data Entry Unit est utilisée pour collecter, dans des champs, des valeurs données en entrée qui seront ensuite utilisées comme paramètres d'une opération. La valeur d'un champ peut être une constante, extraite d'une base de données ou fournie par l'utilisateur. Une propriété (appelée '*activating*') est ajoutée aux liens du modèle de navigation afin d'indiquer que le lien en sortie d'une telle unité active une opération. Un lien d'activation possède également des propriétés permettant le transfert des valeurs de paramètres capturées dans la Data Entry Unit vers l'Operation Unit.

Une Operation Unit permet d'invoquer une opération externe à l'application ou une opération parmi celles prédéfinies dans WebML pour la création, la modification et la suppression de contenu, et la création et la suppression de relations. Deux types de lien peuvent être attachés en sortie d'une opération : appelés OK-Link et KO-link, ils indiquent vers quelle entité (ou page) aller respectivement en cas de succès et d'échec à l'exécution de l'opération.

3.7.7 *La définition de la présentation*

Dans [Ceri00a], les auteurs évoquent un *modèle de présentation* qui vise à exprimer l'apparence des pages du site qui ont été définies dans le modèle de composition. Ils envisagent de recourir à des feuilles de styles génériques (i.e. indépendantes du contenu des pages) et/ou de granularité plus fine qui s'appliquent à des concepts spécifiques (et qu'il est donc nécessaire de

rattacher aux modèles structurel et de composition). Toutefois, dans [Ceri01], il n'est plus fait référence à un véritable modèle conceptuel spécifique à la présentation, mais à des mécanismes d'application de feuilles de styles XSL à des fichiers XML (porteurs du contenu du site). Le « modèle » de présentation se résume alors à la spécification des feuilles de style à attacher aux fichiers de contenu.

3.7.8 La personnalisation

Un modèle structurel WebML contient deux entités prédéfinies appelées Group et User [Ceri00a]. Ces entités décrivent respectivement, un ensemble d'utilisateurs ayant des caractéristiques communes et, un utilisateur individuel. Ces entités possèdent les mêmes qualités que toute entité du modèle structurel (i.e. elles peuvent être reliées aux autres entités par des relations, organisées en hiérarchies, etc.). Les entités Group et User sont utilisées à des fins de gestion des *droits d'accès* et de *personnalisation*.

En matière de *contrôle d'accès*, WebML opère une distinction entre utilisateurs enregistrés et non-enregistrés. Ces derniers sont considérés comme membres d'un groupe appelé *everyone* alors que les utilisateurs connus (i.e. enregistrés) appartiennent à des groupes définis par l'administrateur du site en fonction des besoins. Il est défini une version du site, appelée *vue du site (site view)*, par groupe *et* par dispositif d'accès (i.e. une vue du site pour le groupe G dans l'optique d'une diffusion basée sur HTML, une autre pour le même groupe mais exploitant WML par exemple). Les vues du site sont des constructions définies à partir du même modèle structurel (éventuellement soumis à l'étape de dérivation). Elles sont composées d'un ensemble de pages (et de leur unités constituantes), d'unités de liens entre ces pages (les *direct units*) et d'unités d'opération.

En termes de *personnalisation*, chaque vue du site peut être adaptée au profil d'un utilisateur particulier. Dans WebML, une telle personnalisation peut être mise en œuvre de façon procédurale ou déclarative.

La personnalisation déclarative consiste à recourir à des concepts dérivés (entités, attributs, etc.) dont la définition dépend de données spécifiques à l'utilisateur. Le système récupère les informations relatives à chaque utilisateur et les utilise lors du calcul du contenu des unités correspondant aux concepts dérivés. Par exemple, le calcul du prix d'un article, en tenant compte du pourcentage de remise accordé à l'utilisateur (information stockée dans le profil de ce dernier), est réalisé de façon déclarative. La personnalisation procédurale repose sur une syntaxe XML proposée par WebML pour l'écriture de règles métiers permettant le calcul et le stockage d'information sur l'utilisateur. Une règle métier est représentée par un triplet événement-condition-action qui indique pour un événement détecté, lorsqu'une condition est remplie, quelle action entreprendre. Une telle règle pourrait consister, lors d'un achat effectué par l'utilisateur, sous réserve qu'il s'agisse de son $n^{\text{ième}}$ achat et que le montant total de ses commandes passées jusqu'ici dépasse une somme m , à augmenter le pourcentage de remise accordé à cet utilisateur (i.e. modifier l'information de son profil).

3.7.9 Processus de conception avec WebML

Après avoir présenté les différents modèles proposés par l'approche WebML, nous décrivons le processus de conception que préconisent les auteurs. Ce processus fait intervenir des acteurs aux compétences diverses : un *expert des données* pour la construction du modèle structurel, un *architecte de l'application Web* pour la conception des pages, unités et liens de navigation, un *architecte du style Web* pour les aspects relatifs à la présentation, et un *administrateur Web* pour la définition des utilisateurs, des groupes et de leur droits.

Une première étape du processus consiste à collecter les besoins de l'application en termes d'objectifs du site, de public, d'exemples de contenus attendus, de présentation, et d'éléments de personnalisation. Notons toutefois que WebML ne fait pas de réelle proposition pour l'analyse de ces besoins.

La seconde étape est dédiée à la conception des données. L'expert des données procède, de façon itérative, à la détermination du modèle structurel et aux dérivations nécessaires.

La conception de l'hypertexte est abordée à l'étape suivante, par une première approche globale. Le nombre de vues du site est évalué après avoir défini les groupes d'utilisateurs et leurs dispositifs d'accès. Pour chaque vue du site, l'architecte de l'application met en évidence les pages et liens requis, ainsi que les principales unités composantes de chaque page. La correspondance entre ces unités et les concepts du modèle structurel est définie. L'architecte procède par itérations pour définir la première vue du site et emploie une technique de conception par « copier-coller » pour les suivantes.

L'hypertexte est ensuite affiné par le biais d'une conception détaillée portant sur la définition de chaque page élaborée à l'étape précédente. A cette occasion, l'architecte de l'application introduit des éléments facilitant l'accès à l'information (index, filtre, etc.), définit de nouveaux chemins d'accès ou restructure l'hypertexte (ajout de pages supplémentaires, fractionnement ou fusion de page(s), etc.). L'administrateur Web de l'application intervient également pour intégrer les différents aspects (caractérisation des profils utilisateurs, dérivations guidées par les profils, règles métiers) requis en terme de personnalisation. Lors de cette étape, retravailler le modèle structurel et/ou sa dérivation pour y apporter des modifications peut apparaître nécessaire.

La dernière étape, relative à la dimension présentation, n'est abordée que lorsque l'ensemble des pages d'une vue du site atteint un certain degré de stabilité (i.e. lorsqu'il n'est plus nécessaire de retravailler les modèles structurel, de dérivation, de composition et de navigation).

3.7.10 Conclusion sur WebML

WebML est une notation dédiée à la spécification conceptuelle d'applications Web considérées selon plusieurs dimensions relatives au contenu et à sa gestion, à la composition des pages, aux caractéristiques de navigation et à la présentation de l'hypertexte. La personnalisation de l'application à ses (groupes d') utilisateurs est également abordée. La définition d'un modèle utilisateur, l'exploitation des informations qu'il recense, et la définition de règles, permettent de mettre en œuvre des capacités d'adaptabilité et d'adaptativité.

WebML est une approche orientée modèle qui s'appuie sur un ensemble de concepts disponibles sous forme de représentations graphiques associées à des spécifications basées sur XML. Les auteurs proposent une démarche de conception reposant sur leur langage.

Le langage WebML et la démarche associée sont implémentés dans un outil de conception commercial appelé *WebRatio Site Development Studio*²⁴.

3.8. Conclusion sur les méthodes présentées

Les différentes méthodes que nous avons présentées abordent la conception d'applications et de SI basés sur le Web sous différents angles.

Ainsi, RMM, Hera, OOHD (version de 1996) et WebML sont des *approches orientées par les données*. La conception de l'application est guidée par la définition du domaine. Le modèle

²⁴ disponible à l'adresse <http://www.webratio.com/>

des données produit subit des transformations visant à obtenir le modèle de l'hypermédia utilisé pour présenter l'information à l'utilisateur. Par opposition, WSDM, OOHDM (version de 2000), AWIS-M et UWE sont des *approches orientées utilisateurs*. Elles ont en commun une phase préalable d'analyse des besoins des utilisateurs qui pilote les phases suivantes de conception.

On peut également distinguer les propositions selon la technique qui sous-tend la modélisation. RMM repose sur le modèle Entité-Relation [Chen76]. Des propositions telles que OOHDM, AWIS-M et UWE adoptent une approche orientée objet. WSDM et WebML autorisent les deux approches pour la conception du domaine. La plupart des méthodes propose en outre des notations spécifiques pour modéliser certains aspects (e.g. les *m-slices* dans RMM et OOHDM, les unités de contenus dans WebML, etc.).

L'ensemble des méthodes envisage la conception des applications Web comme une activité qui vise à produire différents modèles. On retrouve de façon générale :

- un modèle de données qui permet de décrire les concepts du domaine manipulés par l'application. On note toutefois que pour RMM, le modèle de données n'est qu'un artefact immédiatement transformé en un modèle de l'hypermédia. Dans les autres approches, il s'agit d'un modèle à part entière, sur lequel reposent d'autres modèles (e.g. des vues sont définies sur le modèle conceptuel des données pour définir le modèle hypermédia dans OOHDM, des correspondances sont établies entre les classes du modèle de données et les classes navigation du modèle hypermédia dans UWE).
- un modèle généralement appelé modèle de navigation qui met en évidence la structure et les possibilités de navigation de l'hypermédia. Il s'agit globalement de définir les nœuds, les liens et les primitives d'accès (index, menu, visite guidée, etc.). A l'exception de WebML qui distingue un modèle de composition et un modèle de navigation, les méthodes représentent la structure et la navigation dans un même modèle.
- un modèle de présentation qui est destiné à spécifier l'apparence et la mise en forme des nœuds (pages) de l'hypermédia. Suggéré dans WSDM, un tel modèle n'est pas décrit dans la littérature à notre connaissance. Les autres approches prennent des options diverses pour cette modélisation. Hera propose des notations graphiques spécifiques. OOHDM utilise les ADV (Abstract Data View [Cowa95]). AWIS-M et WebML préconisent l'utilisation de feuilles de style XSL. UWE base son modèle de présentation sur des extensions de classes UML. On note que UWE ne fait pas de proposition concernant l'apparence en termes de couleurs, choix de police, etc., mais se concentre sur l'organisation spatiale et l'ordonnancement des éléments dans une page et des pages (*frames*) dans une fenêtre de visualisation.

Enfin, en plus de ces trois modèles communs aux différentes approches, on observe pour certaines propositions :

- des modèles pour l'analyse des besoins tels que les Diagrammes d'Interaction Utilisateurs dans OOHDM (version de 2000), le modèle des Objectifs Utilisateurs de AWIS-M, les cas d'utilisation repris dans UWE.
- des modèles des utilisateurs dans Hera, WSDM, AWIS-M, UWE et WebML
- des modèles d'adaptation dans Hera, AWIS-M, UWE et WebML.

Les aspects relatifs à ces derniers modèles sont plus finement abordés dans le chapitre suivant qui porte sur la prise en compte de l'adaptation dans les méthodes que nous venons de présenter.

4. SYNTHÈSE DE L'ÉTAT DE L'ART ET AXES D'ÉTUDE

Dans le chapitre 2, nous avons mis en évidence comment peut s'exprimer l'adaptation dans les SIW. Nous avons ainsi observé les différents types d'adaptation pouvant être mis en œuvre (adaptabilité et adaptativité), décrit les dimensions sur lesquelles peuvent porter les adaptations (contenu, navigation, présentation, fonctionnalité) et souligné l'importance d'une représentation de l'utilisateur pour le processus d'adaptation.

Dans le chapitre 3, les méthodes proposées dans la littérature pour concevoir des applications et SI basés sur le Web ont été présentées. Nous avons décrit pour chacune les modèles et démarches proposés par leurs auteurs.

L'objectif de ce chapitre est d'évaluer dans quelle mesure les sept méthodes étudiées (RMM, Hera, OOHDM, WSDM, AWIS-M, UWE et WebML) prennent en compte l'adaptation à l'utilisateur. Nous définissons pour cela un cadre d'évaluation afin d'estimer les possibilités d'adaptation supportées par les méthodes.

Ce cadre d'évaluation est organisé en quatre rubriques : modélisation de l'utilisateur, dimensions de l'adaptation, modélisation et mise en œuvre des adaptations, et capacités d'adaptation du SIW conçu. Chacune de ces rubriques fait l'objet d'une section (cf. sections 4.1 à 4.4) dans laquelle les méthodes sont évaluées en regard des différents critères retenus.

La section 4.5 conclut ce chapitre en mettant évidence les axes d'étude que nous retenons comme lignes directrices de ce travail à partir de certains manques constatés à l'issue de cette évaluation.

4.1. Modélisation des utilisateurs

4.1.1 Critères d'évaluation

Cette rubrique permet de définir si l'utilisateur fait l'objet d'une représentation dans les approches étudiées. Si tel est le cas, nous détaillons comment est représenté l'utilisateur par les méthodes. Les critères retenus sont présentés et décrits dans le Tableau 4.1.

Tableau 4.1 – Critères d'évaluation relatifs à la modélisation des utilisateurs

| NOM DU CRITÈRE | DESCRIPTION |
|--|---|
| Existence | Exprime s'il existe ou non un modèle décrivant spécifiquement les utilisateurs. |
| Granularité | Décrit le niveau auquel sont représentés les utilisateurs, i.e. groupe, individu, les deux. |
| Caractéristiques : | Précise les informations qui permettent de représenter un groupe ou un utilisateur. |
| Connaissances spécifiques au Domaine d'Application | Indique si les connaissances de l'utilisateur en la matière peuvent être représentées dans le modèle (i.e. niveau de compétence, d'expérience en informatique, etc.). |
| Connaissances Autres | |
| Conditions d'Exploitation | Indique si les conditions (matérielles ou logicielles) dans lesquelles l'utilisateur exploite l'application sont représentées. |
| Fonctionnalités | Indique si une représentation des fonctionnalités (objectif, buts) exploitées par l'utilisateur est donnée. |
| Préférences : | Indique s'il est possible pour l'utilisateur d'exprimer ses préférences et à quel propos : |
| Contenu | Représentation d'un concept, granularité de l'information souhaitée |
| Structure de l'hypermédia | Composition des nœuds en pages et gestion des liens |
| Présentation | Charte graphique personnalisée |
| Acquisition | Exprime comment sont collectées les informations à propos des utilisateurs |
| Droits sur le modèle Utilisateur | Exprime les droits de l'utilisateur à l'égard des informations le concernant (i.e. Aucun / Lecture / Écriture) |

4.1.2 Évaluation des méthodes

Le Tableau 4.2 donne une vision synthétique de l'évaluation des méthodes en regard de la modélisation de l'utilisateur. RMM et OOHDM ne proposent pas de modèle en la matière et ne sont donc pas évaluées sur les autres critères définis (cases vides dans le tableau). Les autres propositions utilisent un modèle des utilisateurs : il s'agit d'un modèle à part entière pour WSDM, Hera, AWIS-M, UWE, alors que WebML intègre la représentation des utilisateurs dans le modèle de données. Les représentations des utilisateurs diffèrent selon les approches comme le montre l'étude des autres critères.

WSDM propose un modèle qui permet de représenter des groupes d'utilisateurs (*classes d'utilisateurs*, affinées en *perspectives*) constitués sur la base de leurs besoins informationnels et fonctionnels communs. Les '*oui*' reportés en italique pour les critères de *connaissances spécifiques au domaine d'application* et *préférences de présentation* traduisent que l'exploitation de telles informations est suggérée par les auteurs mais décrite uniquement de façon informelle.

Une représentation des utilisateurs individuels existe dans Hera. Elle permet de caractériser les conditions matérielles et logicielles dans lesquelles ils exploitent le système. Des préférences en termes de présentation sont également représentées. Le modèle repose sur le vocabulaire CC/PP proposé par le W3C [Klyn01].

AWIS-M permet de représenter des groupes d'utilisateurs (appelés *Catégories d'Utilisateurs*) sur la base de leurs besoins fonctionnels et de leurs profils communs. Les profils portent sur des connaissances spécifiques ou non au domaine et sont à définir par le concepteur. Une évaluation de l'appartenance des individus à ces groupes est effectuée ; cette information est utilisée pour construire un hypermédia adapté, en partie, à l'individu. Néanmoins, il n'est pas précisé dans la proposition comment est géré, représenté et stocké ce type d'information. Nous faisons toutefois la supposition qu'un modèle individuel existe.

Tableau 4.2 – Évaluation des méthodes concernant la modélisation des utilisateurs.

| NOM DU CRITÈRE | | RMM | OOHDM | WSDM | HERA | AWIS-M | UWE | WEBML |
|----------------------------------|--|-----|-------|-------------|-----------------------|-----------------------|----------------|---|
| Existence | | non | non | oui | oui | oui | oui | En tant que partie du modèle conceptuel |
| Granularité | | | | groupe | individu | groupe (individu) | individu | groupe / individu |
| Caractéristiques | Connaissances spécifiques au Domaine d'Application | | | (oui) | non | oui, à définir | oui, à définir | oui, à définir |
| | Connaissances Autres | | | non | non | oui, à définir | oui, à définir | non |
| | Conditions d'Exploitation | | | non | oui | non | non | non |
| | Fonctionnalités | | | oui | non | oui | non | non |
| Préférences | Contenu | | | non | non | non | non | non |
| | Structure de l'hypermédia | | | non | non | non | non | non |
| | Présentation | | | (oui) | oui | non | oui, à définir | Non |
| Acquisition | | | | explicite | explicite / implicite | explicite / implicite | implicite | implicite |
| Droits sur le modèle Utilisateur | | | | non précisé | non précisé | non précisé | non précisé | non précisé |

UWE représente les utilisateurs en tant qu'individus. Les caractéristiques de l'utilisateur sont à définir par création des classes adéquates dans le modèle. Seule la représentation des connaissances est envisagée. Celle des fonctionnalités n'est pas considérée par le modèle : c'est dans la construction même de la structure de l'hypermédia que sont définies les modalités d'accès à l'information. Des possibilités de mener à bien des activités plus complexes (impliquant, par exemple, une saisie de données) ne sont pas décrites. Des préférences de l'utilisateur liées à la présentation peuvent également être exprimées par création de classes spécifiques.

WebML adopte une représentation des groupes et des individus en tant que classes du modèle structurel (i.e. modèle de données). Les groupes sont ici encore définis sur la base de fonctions communes à plusieurs utilisateurs, qui peuvent être plus complexes qu'une simple navigation. Les utilisateurs sont principalement caractérisés par des informations relatives au domaine qui sont exploitables dans des règles de gestion métier : l'application cible de WebML étant le commerce électronique, l'utilisateur est défini par des données telles que le taux de remise qui lui est accordé.

En ce qui concerne l'acquisition des informations constitutives d'un modèle utilisateur, les procédures ne sont pas clairement mentionnées dans les approches. À partir de notre étude des différentes propositions, nous déduisons les modalités probables d'acquisition de l'information qu'elles supposent. Pour WSDM, le concepteur construit explicitement les informations concernant les fonctionnalités de chaque groupe. Les exemples d'interface d'un système proposés dans AWIS-M [Gnah00] montrent que l'utilisateur remplit des formulaires dont les valeurs sont utilisées pour évaluer son appartenance à une catégorie dont les caractéristiques sont données par le concepteur. Il s'agirait donc ici d'un mode d'acquisition explicite de certaines

informations données par l'utilisateur ou le concepteur. De plus, un mode implicite (i.e. détermination par le système) est utilisé en ce qui concerne l'évaluation de l'appartenance d'un utilisateur à une catégorie. Dans Hera, une partie des informations de profil doit être donnée explicitement par le concepteur ou l'utilisateur. Dans Hera, UWE et WebML, un mode d'acquisition implicite de l'information existe, les trois approches proposant de détecter le comportement de l'utilisateur et d'utiliser cette information à des fins d'adaptation. Ceci ne signifie cependant pas qu'il s'agisse du seul mode d'acquisition possible pour UWE et WebML.

On note enfin qu'aucune proposition ne mentionne clairement quels sont les droits de l'utilisateur à l'égard du modèle le concernant. Notamment la lecture et la modification (ou rectification) des caractéristiques et préférences par l'utilisateur lui-même devraient être autorisées même si de telles informations sont automatiquement déterminées par le système. Ceci permet de limiter le sentiment que peut avoir l'utilisateur de subir le système ou d'insatisfaction si les adaptations ne lui conviennent pas.

4.2. Dimensions adaptées

4.2.1 Critères d'évaluation

Les critères de cette rubrique permettent d'identifier les dimensions qui font l'objet d'adaptation dans les approches présentées (cf. Tableau 4.3).

Tableau 4.3 – Critères d'évaluation relatifs aux dimensions adaptées

| NOM DU CRITÈRE | DESCRIPTION |
|-----------------|---|
| Contenu | Exprime s'il existe des adaptations possibles du contenu présenté à l'utilisateur. |
| Structure | Exprime s'il existe des adaptations possibles de la structure de l'hypermédia, i.e. des nœuds et de leur composition. |
| Navigation | Exprime s'il existe des adaptations possibles de la navigation dans l'hypermédia, i.e. des liens et structures d'accès. |
| Présentation | Exprime s'il existe des adaptations possibles de la présentation dans l'hypermédia, i.e. de la mise en page et de l'apparence des éléments. |
| Fonctionnalités | Exprime s'il existe des adaptations possibles concernant les fonctionnalités offertes par l'hypermédia, i.e. adaptation de l'espace fonctionnel et guidage. |

4.2.2 Évaluation des méthodes

Pour chacun des critères, nous indiquons dans le Tableau 4.4 les éléments de modélisation proposés ou les techniques employées à ces fins. Notons que nous mentionnons les aspects qui, selon nous, pourraient être utilisés à des fins d'adaptation même si, dans certaines propositions (RMM, OOHDM), l'adaptation n'apparaît pas clairement supportée (notamment en raison de l'absence d'un modèle utilisateur). Une case vide traduit l'absence de proposition en la matière.

Dans RMM, Hera et OOHDM, les adaptations en termes de contenu, structure et navigation dans l'hypermédia sont confondues. Cela tient au fait que le modèle de l'hypermédia est fortement couplé au modèle du domaine (inclusion du modèle de domaine dans le modèle d'hypermédia pour RMM et définition du modèle de l'hypermédia comme une vue sur le modèle de domaine pour OOHDM).

UWE et WebML proposent une représentation conceptuelle du domaine qui décrit déjà de possibles adaptations de contenu. UWE ajoute le compartiment '*variants*' à la description des classes UML. WebML adopte un modèle de dérivation défini sur le modèle de données. Il s'agit

aussi de vues mais, à la différence de OOHDM, ce modèle de dérivation est indépendant de toute représentation hypermédia. En terme de structure et de navigation, UWE intègre des propriétés (e.g. {removed}) dans le modèle de structure de navigation qui en contraignent les spécifications. WebML définit une vue du site spécifique à chaque groupe d'utilisateurs.

Tableau 4.4 – Évaluation des méthodes concernant les dimensions adaptées

| CRITÈRE | RMM/HERA | OOHDM | WSDM | AWIS-M | UWE | WEBML |
|-----------------|---|--|--|--|--|--|
| Contenu | (RMM) <i>m-slice</i> | <i>Navigational contexts, Navigational classes et classes Incontext</i> | <i>User Object Model et Perspective Object Model</i> | <i>Élément Atomique de Navigation, Nœud de Navigation Adaptable</i> | <i>Classes Variantes</i> | <i>Modèle de dérivation</i> |
| Structure | | | | | propriétés exprimées dans le modèle de structure de Navigation | une vue du site pour chaque groupe |
| Navigation | | | <i>Navigation track</i> | <i>Directive de Navigation</i> | | |
| Présentation | (Hera) <i>m-slice, adaptability et adaptativity conditions et update rules</i> | | | | dépend des spécifications du modèle utilisateur | une vue du site pour chaque groupe et dispositif d'accès |
| Fonctionnalités | | <i>Navigational contexts, Navigational classes et classes Incontext</i> définis en fonction des tâches | <i>User Object Model et Perspective Object Model</i> définis en fonction des besoins | <i>Unité Sémantique de Navigation</i> définies en fonction des objectifs | | Une vue du site pour chaque groupe |

WSDM décrit des modèles (*User Object Model et Perspective Object Model*) qui présentent la vision du modèle de données de chaque groupe d'utilisateur. La correspondance de ces descriptions avec les éléments de structure de l'hypermédia (i.e. nœuds) n'est pas clairement signifiée par l'approche. Les adaptations concernant la navigation sont en fait induites par les deux modèles ci-dessus : les pistes de navigation (*navigation tracks*) reflètent uniquement les possibilités de cheminement dans les éléments du modèle.

AWIS-M offre des concepts exprimant une certaine adaptation en terme de contenu via les Éléments Atomiques de Navigation (EAN). Néanmoins l'adaptation consiste uniquement à présenter ou non un EAN. Les Nœuds de Navigation Adaptables, composés des EAN retenus, peuvent être considérés comme définissant une structure adaptée. L'adaptation de la navigation prend la forme de choix offerts à l'utilisateur et exprimés dans les directives de navigation. On notera que les nœuds et les directives de navigation sont décrits pour chaque catégorie d'utilisateur (adaptation au niveau du groupe).

WebML prend le parti de définir une vue du site spécifique à chaque groupe comme souligné précédemment, mais également spécifique à chaque dispositif d'accès utilisé dans le groupe. Des feuilles de styles XLST adéquates sont définies, traitant ainsi une forme d'adaptation de la présentation. Hera adopte une approche plus fine en définissant des conditions de présentation sur les *m-slices* et en exploitant celles-ci dans différents processus d'adaptation qui tiennent compte des préférences des utilisateurs et des conditions d'exploitation.

Nous avons enfin mentionné que OOHDm, WSDM, AWIS-M et WebML font des propositions pour l'adaptation des fonctionnalités du SIW. Ces approches ont en commun une construction de différentes versions de l'hypermédia adaptées aux différents groupes d'utilisateurs. En conséquence, les pages auxquelles peuvent accéder les utilisateurs constituent un espace (restreint) qui répond à leurs besoins informationnels et fonctionnels. On note que AWIS-M introduit à travers les directives de navigation une certaine aide à la réalisation des tâches en proposant des alternatives et décrivant des critères de choix.

4.3. Modélisation et mise en œuvre de l'adaptation

4.3.1 Critères d'évaluation

A travers cette rubrique nous observons s'il existe une phase d'analyse des besoins en matière d'adaptation. Nous indiquons si les méthodes ont ou non recours à un modèle conceptuel de l'adaptation. La technique de mise en œuvre retenue pour procéder aux adaptations est précisée.

4.3.2 Évaluation des méthodes

Le Tableau 4.5 synthétise les résultats de cette analyse. Une case vide indique l'absence de proposition.

Dans WSDM, AWIS-M et UWE, une phase d'analyse vise à déterminer les besoins fonctionnels des utilisateurs. Nous l'indiquons comme relevant de l'analyse des besoins en matière d'adaptation dans la mesure où elle conduit à la détermination de différents groupes d'utilisateurs en fonction desquels différentes versions de l'hypermédia seront construites. Néanmoins, l'analyse des besoins fonctionnels ne suffit pas, selon nous, à déterminer réellement des attentes relatives aux capacités d'adaptation du système.

OOHDm et WSDM ne présentent pas de modèle conceptuel de l'adaptation. WebML ne propose pas de modèle spécifique : la personnalisation déclarative correspond à des spécifications introduites dans les classes groupes et utilisateurs qui permettent de dériver le modèle de dérivation. La personnalisation procédurale est exprimée par des règles événement – condition – action. Les auteurs ne précisent pas, à notre connaissance, à quelle structure est rattachée une telle règle. Nous faisons l'hypothèse qu'elles sont intégrées dans les descriptions des unités du modèle de composition.

Ces trois approches ont en commun une mise en œuvre de l'adaptation qui se traduit par le choix d'une version prédéfinie adaptée au groupe auquel appartient l'utilisateur.

Dans Hera, les éléments pour l'adaptabilité sont définis dans le modèle conceptuel de l'hypermédia sous forme de conditions sur les *m-slices*. L'adaptativité repose sur un modèle spécifique de règles. L'hypermédia délivré à l'utilisateur est construit en fonction de ces spécifications. D'une part, les éléments qui satisfont les conditions décrites dans le modèle conceptuel sont retenus pour composer les pages. D'autre part, les règles de types événement – condition – action sont traitées par le moteur d'adaptation emprunté à AHA [DeBr99].

Le modèle mathématique de l'hyperespace proposé par AWIS-M peut être vu comme un modèle d'adaptation. C'est sur la base des spécifications qu'il contient que l'hypermédia est construit dynamiquement selon une approche descendante : les Unités Sémantiques de Navigation (USN) sont sélectionnées en fonction de la catégorie à laquelle appartient l'utilisateur, puis les Nœuds de Navigation (NN) pertinents en fonction du profil de la catégorie d'utilisateur sont retenus, et enfin, les Éléments de Base (EB) sont choisis en fonction du degré d'appartenance de l'utilisateur à ce profil.

Tableau 4.5 – Évaluation des méthodes concernant la modélisation et la mise en œuvre de l'adaptation

| CRITÈRE | OOHDM | WSDM | WEBML |
|---|--|--|---|
| Analyse des besoins en matière d'adaptation | | Analyse des besoins fonctionnels – détermination de groupes (Mission Statements) | |
| Modèle conceptuel de l'adaptation | | | Pas de modèle particulier mais spécification de règles de personnalisation déclarative et procédurale |
| Technique de mise en œuvre de l'adaptation | Choix de la version correspondant au groupe parmi différentes versions prédéfinies | Choix de la version correspondant au groupe parmi différentes versions prédéfinies | Choix de l'hypermédia adapté à chaque groupe (site view) |

(Suite du Tableau 4.5)

| CRITÈRE | HERA | | AWIS-M | UWE |
|---|---|---|---|---|
| | Adaptabilité | Adaptativité | | |
| Analyse des besoins en matière d'adaptation | | | Analyse des besoins fonctionnels – détermination de groupes | Analyse des besoins fonctionnels – détermination de groupes (use cases) |
| Modèle conceptuel de l'adaptation | Conditions incluses dans le modèle conceptuel de l'hypermédia | Modèle spécifique de règles d'adaptation | Décrit par le modèle mathématique de l'hyperespace | Modèle d'adaptation défini sur la base des spécifications ajoutées aux autres modèles (classes Rule, Condition, Action, User Behaviour) |
| Technique de mise en œuvre de l'adaptation | Recherche des conditions satisfaites au moment de la présentation | Règles événement – condition – action traitées par un moteur d'adaptation | Construction dynamique d'un hypermédia adapté à chaque groupe par sélection des USN et des NN et par assemblage des EB en fonction de l'utilisateur | Implémentation d'un moteur d'adaptation |

Enfin, UWE propose un modèle spécifique de l'adaptation qui exprime sous forme de règles l'ensemble des spécifications éventuellement insérées dans les autres modèles. Néanmoins, l'implémentation d'un moteur d'adaptation ou de fonctionnalités capables de traiter ces règles reste à la charge du développeur.

4.4. Capacités d'adaptation du SIW conçu

4.4.1 Critères d'évaluation

Cette dernière rubrique a comme objectif d'évaluer les capacités d'adaptation que l'on peut attendre des SIW conçus avec les méthodes que nous avons étudiées. Nous distinguons les critères mentionnés dans le Tableau 4.6.

On note que les définitions retenues pour les différentes capacités d'adaptation combinent les définitions de l'adaptabilité et de l'adaptativité présentées au chapitre 2 (cf. section 2.3) et les simplifient dans une certaine mesure. Les deux approches sont rappelées par les axes de la Figure 4.1 ; L'axe vertical fait référence à la définition de [Step98a][Fras02][Kapp00] qui distingue adaptabilité et adaptativité en fonction du moment où est réalisée l'adaptation (i.e. à

l'initialisation ou en temps réel). L'axe horizontal reprend la vision de [Diet93][Oppe94][Kobs01a] basée sur le contrôle qu'a l'utilisateur dans le processus d'adaptation.

Tableau 4.6 – Critères d'évaluation relatifs aux capacités d'adaptation du SIW

| NOM DU CRITÈRE | DESCRIPTION |
|---------------------|--|
| Adaptation minimale | Cette capacité correspond aux systèmes qui permettent de choisir une version parmi plusieurs définies au préalable |
| Adaptabilité | Cette capacité est définie comme l'aptitude du système à réagir à des demandes explicites d'adaptation de la part de l'utilisateur |
| Adaptativité | Cette capacité est définie comme l'aptitude du système à observer le comportement de l'utilisateur et à procéder en conséquence à des adaptations. |

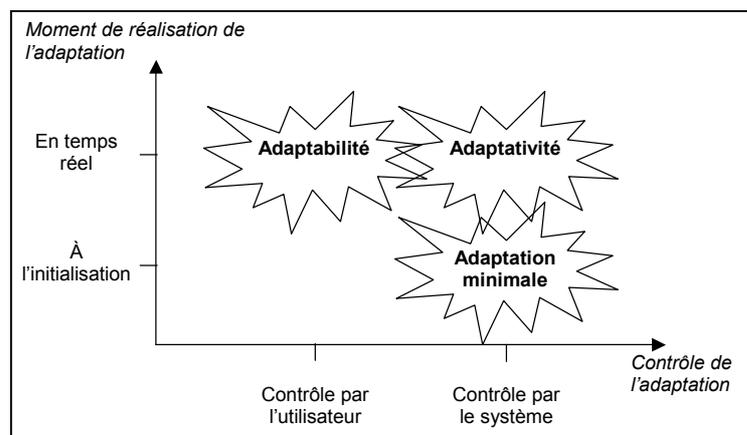


Figure 4.1 – Typologie retenue pour les capacités d'adaptation des SIW

La combinaison de ces deux approches apparaît à travers le croisement de ces deux dimensions. La simplification des définitions que nous retenons est liée au fait que nous considérons uniquement les deux cas extrêmes de processus d'adaptation tels que décrits par les auteurs proposant les notions de *contrôle de l'utilisateur* ou *du système* comme discriminantes de l'adaptabilité et de l'adaptativité.

4.4.2 Évaluation des méthodes

Cette évaluation repose sur les constats établis grâce aux précédentes rubriques. Elle est présentée dans le Tableau 4.7. Les conventions utilisées dans le tableau sont les suivantes : le caractère '+' indique que le SIW présente la capacité d'adaptation associée, le caractère '-' traduit que cette capacité n'est pas observable dans le système, l'association de ces deux caractères '+ / -' exprime que la présence de cette capacité est discutable. Nous commentons l'évaluation de ces capacités à la suite du tableau.

La capacité d'adaptation minimale est potentiellement observable dans tous les SIW conçus avec les méthodes présentées. Toutes les approches permettent, en effet, de modéliser des systèmes qui sont générés dans une version adaptée à l'utilisateur. Deux conditions méritent d'être posées pour une réelle observation de cette capacité. D'une part, il est évidemment nécessaire que l'implémentation aille dans ce sens *i.e.* que la programmation respecte les spécifications des modèles et les principes associés. D'autre part, la seconde condition porte sur l'obligatoire identification de l'utilisateur et du groupe auquel il appartient afin de pouvoir

choisir la version du SIW à présenter. RMM, OOHDM et WSDM proposent uniquement de choisir la version qui correspond à l'utilisateur et ne permettent pas d'aboutir à des systèmes présentant une autre capacité que l'adaptation minimale.

Tableau 4.7 – Évaluation des capacités d'adaptation des SIW conçus avec les méthodes étudiées.

| CRITÈRE | RMM | OOHDM | WSDM | HERA | WEBML | AWIS-M | UWE |
|---------------------|-----|-------|------|-------|-------|--------|-------|
| Adaptation minimale | + | + | + | + | + | + | + |
| Adaptabilité | – | – | – | + / – | + / – | + / – | + / – |
| Adaptativité | – | – | – | + / – | + / – | + / – | + |

En ce qui concerne les capacités d'adaptabilité du SIW (vues comme les possibilités offertes à l'utilisateur de contrôler le processus d'adaptation), l'évaluation est plus mitigée. Les raisons à cela sont principalement liées, d'une part à l'absence de description explicite des droits d'un utilisateur sur sa propre représentation dans le système, et, d'autre part, au peu de préférences de l'utilisateur prises en compte par les modèles (seules quelques préférences portant sur la présentation sont parfois envisagées). Les exemples d'interface déjà mentionnés issus de AWIS-M montrent qu'il est possible de collecter des informations fournies par l'utilisateur. Néanmoins, ces informations sont utilisées à des fins de classification de l'utilisateur dans une catégorie et ne lui permettent pas réellement d'initier de façon explicite des adaptations. D'une manière générale, Hera, WebML, AWIS-M et UWE présentent certaines caractéristiques pouvant laisser supposer que les SIW offrent des capacités d'adaptabilité. Nous pensons notamment au fait que ces méthodes gèrent une représentation individuelle des utilisateurs. Toutefois, aucune description ne positionne clairement ces approches en terme d'adaptabilité.

L'adaptativité est en partie prise en compte par ces quatre méthodes. AWIS-M permet la construction dynamique de l'hypermédia en fonction de *qui est* l'utilisateur mais pas en fonction de *ce qu'il fait*. En d'autres termes, la méthode n'établit pas de protocole de détection des comportements de l'utilisateur utilisée par le système pour réagir en conséquence (i.e. pour s'adapter). Hera, WebML et UWE proposent chacune des modèles permettant une détection plus ou moins fine des comportements des utilisateurs. Dans Hera, les comportements détectés correspondent uniquement aux pages visitées par l'utilisateur ; à partir de cette information le degré d'intérêt de l'utilisateur pour un concept est évalué et utilisé pour donner accès à de l'information supplémentaire. Dans WebML, les comportements détectés ont pour seul objectif d'être exploités dans les fonctions mettant en jeu une logique métier principalement de nature commerciale. UWE est la proposition qui offre le plus de possibilités en la matière, le concepteur étant libre de définir les comportements qu'il souhaite voir détectés par le système et de spécifier les actions à entreprendre suite à leur observation.

4.5. Conclusion et aperçu des principaux axes de notre proposition

De l'évaluation des méthodes, nous concluons que la prise en compte des capacités d'adaptation dans un processus de développement peut être améliorée et complétée. Nous dégageons trois axes principaux d'étude en rapport avec ces aspects. Nous les présentons dans ce qui suit, et donnons les éléments essentiels de notre approche qui permettent de pallier les manques constatés. Un quatrième point présente brièvement l'environnement que nous proposons pour concevoir des SIW tenant compte des trois axes d'étude retenus.

4.5.1 Prise en compte de l'adaptation et modélisation multidimensionnelle

La conception d'un SIW est généralement caractérisée par la description du système selon différentes perspectives (modèle du domaine, de la navigation, de la présentation, des fonctionnalités). La relative indépendance de leur conception permet de réduire la complexité de la tâche, mais également de gagner en flexibilité. Plusieurs structures de navigation peuvent être définies sur un même modèle de données, chacune pouvant, à son tour, faire l'objet de plusieurs types de présentation. Une démarche similaire guidant la conception des adaptations est selon nous totalement justifiée. Les adaptations ont la particularité de s'appliquer à chacune des dimensions citées : le contenu, la structure de l'hypermédia et la présentation sont des objets d'adaptation potentiels. Lors de la conception, les considérer chacune séparément du point de vue de l'adaptation facilite les spécifications mais a surtout l'avantage d'en accroître la flexibilité. Ceci est un atout certain dans l'optique de s'adapter à chaque utilisateur considéré individuellement. En conséquence, une modélisation des adaptations doit, selon nous, être engagée pour chacune de ces perspectives, comme cela est déjà suggéré dans [Koch00] ou encore dans [Cann01b]. Dans notre proposition, la conception d'un SIW repose sur des modèles qui trouvent leurs correspondants dans ceux proposés par les méthodes étudiées au chapitre 3. Ainsi nous décrivons :

- Le *Modèle du Domaine* pour représenter conceptuellement les objets du monde réel qui définissent le domaine d'application sur lequel est construit le SIW.
- Le *Modèle des Fonctionnalités* offrant une représentation conceptuelle des différentes tâches que peut réaliser un utilisateur grâce au SIW.
- Le *Modèle de l'Hypermédia* qui traite des aspects relatifs à la construction de l'hypermédia, vu comme la face visible du SIW pour l'utilisateur.
- Le *Modèle des Utilisateurs* permettant la représentation aussi bien des groupes que des individus. Ce modèle est le garant d'une adaptation selon plusieurs dimensions relatives, par exemple, au contenu ou à la présentation.

4.5.2 Adaptation minimale, adaptabilité et adaptativité du SIW

Les capacités attendues d'un SIW doivent selon nous s'exprimer de trois façons :

- la première relève de l'*adaptation minimale* : il est nécessaire qu'à la connexion de l'utilisateur, une version déjà en partie adaptée du SIW lui soit proposée. L'objectif de cette adaptation minimale est d'initier un sentiment positif chez l'utilisateur en lui montrant notamment ce qu'il peut attendre du système.
- la deuxième capacité, l'*adaptabilité*, permet à l'utilisateur d'être acteur du processus d'adaptation. Gardant un certain contrôle sur les adaptations réalisées par le système, l'utilisateur a moins l'impression de subir le système.
- enfin, l'*adaptativité* constitue la troisième capacité dont doit faire preuve un SIW. L'intérêt est d'éviter certaines manipulations à l'utilisateur auxquelles le fait de devoir procéder pourrait être perçu comme un signe d'inefficacité du système. Un autre avantage réside dans la possibilité que représente l'adaptativité d'anticiper voire de révéler des adaptations pertinentes pour l'utilisateur.

Notre proposition permet de définir des SIW dotés de ces trois types d'adaptation, notamment du point de vue de la mise en œuvre des Accès Progressifs qui constitue l'axe central de cette recherche.

4.5.3 Contribution de l'accès progressif aux capacités d'adaptation du SIW

Nous constatons un faible support offert par les méthodes pour gérer, avec un objectif d'adaptation à l'utilisateur, la masse d'informations à laquelle celui-ci peut être confronté. Les méthodes présentées reposent toutes sur un mode de diffusion de l'information exploitant les principes des hypermédias. De par la nature non-séquentielle de l'accès à l'information dans les hypermédias, l'utilisateur bénéficie d'une certaine flexibilité dans l'utilisation du SIW. Néanmoins, celle-ci peut être à l'origine d'une désorientation de l'utilisateur, amplifiée par une surcharge cognitive. Cette surcharge résulte de difficultés à appréhender des informations lorsqu'elles sont en trop grand nombre. Pour éviter les inconvénients liés à une quantité trop importante de données et de services mis à disposition par le SIW, une réflexion portant sur la façon de les organiser pour en faciliter l'utilisation est nécessaire. Ce travail vise à apporter une contribution dans ce sens, en mettant en œuvre une approche permettant à l'utilisateur d'agir sur la quantité d'informations délivrées par le système.

Le concept d'*Accès Progressif*, sa modélisation et sa mise en œuvre constituent la proposition majeure de ce travail. L'*Accès Progressif* répond au besoin de procéder à une organisation spécifique des informations permettant de contrer la désorientation et la surcharge cognitive. Le *Modèle d'Accès Progressif* (MAP) est un modèle qui sert de référence dans notre proposition : ses concepts et ses principes sont appliqués aux modèles présentés dans la section 4.5.1, dans le but de concevoir des SIW capables de supporter l'accès progressif :

- les informations supportées par le *Modèle du Domaine* peuvent faire l'objet de partitions qui autorisent un accès progressif aux données du domaine. Nous montrons comment un tel modèle est interprété dans les termes définis par le MAP.
- Le *Modèle des Fonctionnalités* décrit notamment comment l'accès progressif est intégré, dès la phase de conception, dans la spécification des fonctionnalités offertes par le SIW.
- Le *Modèle Hypermédia*, obtenu par dérivation des spécifications des différents modèles, permet de construire un espace d'informations et de fonctionnalités adaptées (en terme de contenu et de présentation) et accessibles progressivement.
- Le *Modèle des Utilisateurs*, à travers ses connexions avec les autres modèles garantit la conception d'un SIW adapté à ses utilisateurs selon les différentes dimensions considérées. En particulier, sa connexion avec le MAP assure un accès progressif *personnalisé*.

4.5.4 Un environnement de conception supportant l'approche

Notre proposition comporte également un environnement appelé KIWIS (Knowledge for Improving Web Information System). Développé à des fins de validation et d'opérationnalisation de notre approche, KIWIS est à la fois :

- un *environnement pour la conception* de SIW adaptables à leurs utilisateurs et capables de supporter l'accès progressif. En ce sens, KIWIS est destiné aux concepteurs de SIW.
- un *environnement pour l'utilisation* des systèmes qu'il permet de concevoir et de générer. Il constitue donc également une plate-forme pour les utilisateurs finaux des SIW.

L'architecture de cet environnement intègre différents modules assurant la gestion des modèles proposés, tant lors de la conception des SIW que lors de leur exploitation une fois la génération des SIW effectuée.

PROPOSITION

5. MODÈLE D'ACCÈS PROGRESSIF

Les SIW sont caractérisés par la dimension hypermédia qui constitue la « face visible » du système pour les utilisateurs. Comme nous l'avons vu dans l'état de l'art (*cf.* chapitre 3), plusieurs travaux intègrent ainsi une réflexion sur la conception de l'hypermédia dans le processus d'ingénierie du SIW. Ce mode de diffusion de l'information est intéressant en raison de la flexibilité qu'il introduit. Les possibilités d'accès non-séquentiel à l'information offrent à l'utilisateur la latitude de choisir son parcours dans l'hypermédia. Un inconvénient majeur des structures hypermédia réside cependant dans le risque avéré que l'utilisateur se perde dans le labyrinthe des pages (« *lost in hyperspace syndrome* » [Conk87]). La désorientation de l'utilisateur est essentiellement due au fait que le SIW se présente comme un espace très vaste et très riche en possibilités à la fois d'accès à l'information et de fonctionnalités.

La notion d'accès progressif, qui constitue le fil directeur de ce chapitre, repose sur l'hypothèse suivante : l'utilisateur n'a pas *tout le temps* besoin d'accéder à *toute* l'information disponible. A travers l'accès progressif, nous souhaitons permettre à l'utilisateur d'accéder, dans un premier temps, aux informations qui sont essentielles pour lui, puis, de façon graduelle, lui donner la possibilité, s'il le souhaite, d'accéder à d'autres informations. L'approche vise à réduire la surcharge cognitive parfois subie par les utilisateurs face à une masse informationnelle difficile à appréhender dans sa globalité.

L'objectif de notre proposition est donc de tendre vers de tels systèmes, capables de mettre en oeuvre des modalités d'accès progressif. L'atteinte de cet objectif nécessite de définir un cadre expliquant le sens donné à la notion d'accès progressif, identifiant et définissant les concepts qu'il met en jeu et fixant les règles de leur utilisation. Le Modèle d'Accès Progressif décrit dans ce chapitre constitue ce cadre.

Dans la section 5.1, nous présentons les différentes notions liées à l'accès progressif et en expliquons les principes. A l'issue de cette présentation, nous décrivons le *Modèle d'Accès Progressif* (MAP) au moyen de notations UML étendues.

Ce modèle est complété lors de la section 5.2 par la présentation d'un protocole d'exploitation du MAP. Celui-ci décrit comment est régi l'accès progressif et comment réorganiser les caractéristiques de cet accès en vue d'une meilleure adaptation aux utilisateurs.

Dans la section 5.3, nous montrons comment le MAP peut être utilisé pour mettre en œuvre un accès progressif sur un modèle de données exprimé en UML.

Dans la section qui conclut ce chapitre, nous expliquons en quoi le MAP constitue un cadre de référence pour la suite de la proposition. Nous exposons les principes d'exploitation de l'Accès Progressif lors de la conception d'un SIW (*cf.* section 5.4).

5.1. Description du Modèle d'Accès Progressif

Nous décrivons ici l'accès progressif à un niveau général, *i.e.* indépendamment de toute référence aux concepts spécifiques d'un SIW. Nous définissons tout d'abord les notions permettant de comprendre les principes d'une approche basée sur l'accès progressif. Nous présentons ensuite une description en UML du Modèle d'Accès Progressif qui met en œuvre les notions décrites.

5.1.1 Notions pour l'accès progressif

5.1.1.1 Entité Masquable et Représentations d'Entité Masquable

Une *Entité Masquable (EM)* est un ensemble d'au moins deux éléments (la cardinalité de l'ensemble EM est donnée par $|EM| \geq 2$) sur lequel on souhaite mettre en place un accès progressif. L'accès progressif sur une EM repose sur la définition de *Représentations d'Entité Masquable (REM)* associées à cette EM. En règle générale, les REM sont des sous-ensembles de l'EM ordonnés par la relation d'inclusion ensembliste. Ces REM sont dites *extensionnelles* lorsqu'elles agissent sur l'*extension* de l'EM. L'extension est ici définie comme l'ensemble des éléments de l'Entité Masquable. Dans le cas où l'EM correspond à un ensemble de données structurées de même type, on peut également lui associer un second type de REM, les REM *intensionnelles*, qui agissent sur la structure de l'EM. Les REM intensionnelles définissent des sous-ensembles de l'*intension* de l'EM, également ordonnés par la relation d'inclusion ensembliste. L'intension d'une EM structurée est ici définie comme l'ensemble des descriptions d'attributs (appelées variables ou champs dans d'autres contextes informatiques) qui composent la structure de l'EM.

Quelle que soit sa nature - extensionnelle ou structurelle -, la REM est associée à un *niveau de détail* ; on note REM_i la représentation de niveau de détail i avec $1 \leq i \leq max$, où max est le niveau maximum de détail défini pour l'EM.

Nous donnons ci-dessous, les règles de bonne formation des deux types de REM :

- REM extensionnelles

Soit une $EM = \{e_1, e_2, \dots, e_n\}$ telle que les e_i sont les éléments de EM,

toute REM_i extensionnelle définie pour EM est non vide ($REM_i \neq \emptyset$)

$REM_1 = \{e_k, \dots, e_m\}$ avec $\{e_k, \dots, e_m\} \subset E(EM)$ où E est la fonction qui associe à EM son extension (REM_1 ne peut pas être égale à l'extension de EM)

pour toutes REM_i et REM_j extensionnelles définies pour EM et telles que $j = i+1$,

$REM_j = REM_i \cup \{e_p, \dots, e_k\}$ avec $\{e_p, \dots, e_k\} \subseteq (E(EM) \setminus REM_i)$

- REM intensionnelles

Soit une EM dont l'intension $I(EM)$ est définie par $I(EM) = \{a_1: t_1, a_2: t_2, \dots, a_n: t_n\}$ où chaque $a_i \in \{a_1, \dots, a_n\}$ est un attribut (variable ou champ) de la structure commune à chaque élément de EM, et chaque $t_i \in \{t_1, \dots, t_n\}$, le type associé à l'attribut a_i

toute REM_i intensionnelle définie pour EM est non vide ($REM_i \neq \emptyset$)
 $REM_i = \{a_k: t_k, \dots, a_m: t_m\}$ avec $\{a_k: t_k, \dots, a_m: t_m\} \subset I(EM)$ (REM_i ne peut pas être égale à EM)
 pour toutes REM_i et REM_j intensionnelles définies pour EM et telles que $j = i+1$,
 $REM_j = REM_i \cup \{a_r: t_r, \dots, a_s: t_s\}$ avec $\{a_r: t_r, \dots, a_s: t_s\} \subseteq (I(EM) \setminus REM_i)$

Pour une REM, les éléments de l'extension d'une EM présents à un niveau de détail i sont également présents au niveau supérieur de détail $i+1$. De même, pour une REM intensionnelle, les éléments structurels (*i.e.* les descriptions des attributs) d'une EM présents à un niveau de détail i sont également présents au niveau supérieur de détail $i+1$. De plus, au niveau de détail $i+1$, au moins un élément supplémentaire (de l'extension ou de l'intension) de EM participe à REM_{i+1} , celle-ci étant souhaitée plus complète que REM_i . Cet (ou ces) élément(s) ajouté(s) dans REM_{i+1} par rapport à REM_i , est (sont) choisi(s) parmi les éléments de EM n'appartenant pas à REM_i .

La Figure 5.1 illustre les concepts que nous venons de décrire dans le cas de REM extensionnelles, le principe étant similaire dans le cas de REM intensionnelles. L'Entité Masquable présentée à gauche de la figure est constituée de divers éléments. Une conséquence des règles de formation des REM est que leurs éléments constituent des sous-ensembles de l'extension de EM. A partir de cette EM, trois Représentations d'Entité Masquable, correspondant à autant de niveaux de détail, sont définies. Au centre, nous symbolisons les ensembles d'éléments ajoutés par chaque REM. On note que l'élément symbolisé par le losange reste toujours masqué : même au plus haut niveau de détail, il ne participe pas à la représentation de l'entité.

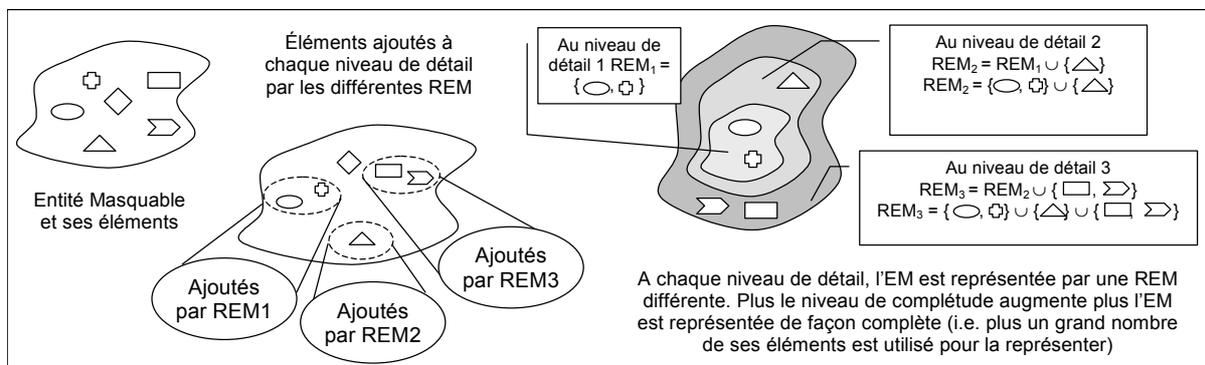


Figure 5.1 – Entité Masquable et Représentations d'Entité Masquable.

Les REM peuvent donc être considérées comme des masques sur une EM : elles offrent une visibilité sur les éléments à un certain niveau de détail, mais également, elles cachent une partie des éléments. Dans ce sens, les REM peuvent être utilisées pour assurer une certaine confidentialité de l'information. Cette ambivalence pourra être mise à profit lors de la mise en œuvre de ces REM dans un SIW.

Les définitions données plus haut sont très proches et la sémantique des Représentations d'Entité Masquable est préservée puisque, dans les deux cas, elles correspondent toutes deux à des sous-ensembles ordonnés par l'inclusion ensembliste. Aussi, dans la suite de ce chapitre, nous ne distinguons les REM extensionnelles des REM intensionnelles que lorsque ceci est nécessaire.

5.1.1.2 Stratification

Nous appelons *stratification* le processus (et son résultat) qui vise à définir une organisation particulière d'un ensemble d'éléments, appelé Entité Masquable, que ce soit en Représentations d'Entité Masquable extensionnelles (on parlera alors de *stratification en extension*) ou en Représentations d'Entité Masquable intensionnelles (on parlera alors de *stratification en intension*). Plus généralement, une stratification correspond à une organisation d'une EM en REM, extensionnelles et intensionnelles confondues ; elle combine donc les deux types de stratification mentionnés. Cette stratification est telle que les Représentations d'Entités Masquables :

- sont ordonnées,
- sont liées par une relation ensembliste d'inclusion stricte,
- sont associées à des mécanismes spécifiques pour un accès progressif,
- permettent de donner une représentation de l'EM à différents niveaux de détail.

5.1.1.3 Définitions de fonctions pour l'accès progressif

Nous donnons ici des définitions fonctionnelles des notions associées à l'accès progressif : *extension*, *intension*, *REM* et *stratification*. Ces définitions visent à formaliser ces notions et à préciser dans la suite comment les stratifications peuvent être utilisées.

Nous considérons les alphabets de symboles suivants :

- \mathcal{EM} est l'ensemble des symboles Entités Masquables
- \mathcal{E} est l'ensemble des éléments d'Entité Masquable
- \mathcal{A} est l'ensemble des noms d'attributs d'Entité Masquable
- \mathcal{t} est l'ensemble des types d'attributs d'Entité Masquable
- \mathcal{D} est l'ensemble des descriptions d'attributs d'Entité Masquable

➤ Extension :

Soit E la fonction d'extension d'une Entité Masquable. E est définie par :

$$E : \mathcal{EM} \rightarrow 2^{\mathcal{E}}$$

$$E(EM) = \{e_1, e_2, \dots, e_m\} \text{ avec } EM \in \mathcal{EM}, \text{ et } e_i \in \mathcal{E}, \forall i \in [1, m]$$

Soit E^{-1} la fonction inverse de l'extension d'une Entité Masquable. E^{-1} est définie par :

$$E^{-1} : 2^{\mathcal{E}} \rightarrow \mathcal{EM}$$

$$E^{-1}(\{e_1, e_2, \dots, e_m\}) = EM \text{ avec } EM \in \mathcal{EM}, \text{ et } e_i \in \mathcal{E}, \forall i \in [1, m]$$

➤ Intension :

Soit I la fonction d'intension d'une Entité Masquable structurée. I est définie par :

$$I : \mathcal{EM} \rightarrow 2^{\mathcal{D}}$$

$$I(EM) = \{a_1: t_1, a_2: t_2, \dots, a_n: t_n\}$$

$$\text{avec } EM \in \mathcal{EM}, a_i \in \mathcal{A}, t_i \in \mathcal{t}, \text{ et } a_i : t_i \in \mathcal{D}, \forall i \in [1, n]$$

et le type de EM est le type-record [Card91] défini par $\langle a_1: t_1 a_2: t_2, \dots, a_n: t_n \rangle$

Soit I^{-1} la fonction inverse de l'intension d'une Entité Masquable structurée. I^{-1} est définie par :

$$\Gamma^1 : 2^{\mathcal{D}} \rightarrow \mathcal{EM}$$

$$\Gamma^1 (\{a_1: t_1, a_2: t_2, \dots, a_n: t_n\}) = EM$$

avec $EM \in \mathcal{EM}$, $a_i \in \mathcal{A}$, $t_i \in \mathcal{t}$, et $a_i : t_i \in \mathcal{D}$, $\forall i \in [1, n]$

et le type de EM est le type-record [Card91] défini par $\langle a_1: t_1 a_2: t_2, \dots, a_n: t_n \rangle$

➤ REM extensionnelle :

Soit REM_{EXT} la fonction de REM extensionnelle d'une Entité Masquable. REM_{EXT} est définie par :

$$REM_{EXT}: 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}}$$

$$REM_{EXT}(E(EM)) = \{e_i, \dots, e_j\} \text{ avec } EM \in \mathcal{EM}, \text{ et } \{e_i, \dots, e_j\} \subseteq E(EM)$$

➤ REM intensionnelle :

Soit REM_{INT} la fonction de REM intensionnelle d'une Entité Masquable. REM_{INT} est définie par :

$$REM_{INT}: 2^{\mathcal{D}} \rightarrow 2^{\mathcal{D}}$$

$$REM_{INT}(I(EM)) = \{a_k: t_k, \dots, a_l: t_l\} \text{ avec } EM \in \mathcal{EM}, \text{ et } \{a_k: t_k, \dots, a_l: t_l\} \subseteq I(EM)$$

➤ Stratification en extension :

Soit $STRAT_{EXT}$ la fonction de stratification en extension d'une Entité Masquable. $STRAT_{EXT}$ est définie par :

$$STRAT_{EXT}: 2^{\mathcal{E}} \rightarrow 2^{2^{\mathcal{E}}}$$

$$STRAT_{EXT}(E(EM)) = \{REM_{EXT}(E(EM))_1, REM_{EXT}(E(EM))_2, \dots, REM_{EXT}(E(EM))_p\}$$

avec $EM \in \mathcal{EM}$, $REM_{EXT}(E(EM))_1 \neq E(EM)$, $\forall i \in [1, p]$, $p \geq 2$, p est le niveau de détail maximum pour cette EM stratifiée en extension, $REM_{EXT}(E(EM))_i \neq \emptyset$,

$$\forall i \in [1, p-1], REM_{EXT}(E(EM))_i \subset REM_{EXT}(E(EM))_{i+1}$$

➤ Stratification en intension :

Soit $STRAT_{INT}$ la fonction de stratification en intension d'une Entité Masquable. $STRAT_{INT}$ est définie par :

$$STRAT_{INT}: 2^{\mathcal{D}} \rightarrow 2^{2^{\mathcal{D}}}$$

$$STRAT_{INT}(I(EM)) = \{REM_{INT}(I(EM))_1, REM_{INT}(I(EM))_2, \dots, REM_{INT}(I(EM))_q\}$$

avec $EM \in \mathcal{EM}$, $REM_{INT}(I(EM))_1 \neq I(EM)$, $\forall j \in [1, q]$, $q \geq 2$, q est le niveau de détail maximum pour cette EM stratifiée en intension, $REM_{INT}(I(EM))_j \neq \emptyset$,

$$\forall j \in [1, q-1], REM_{INT}(I(EM))_j \subset REM_{INT}(I(EM))_{j+1}$$

Deux fonctions d'accès progressif permettent de passer d'un niveau de détail à l'autre. Étant donnée une REM (intensionnelle ou extensionnelle), le *masquage* donne accès à la REM (intensionnelle ou extensionnelle) de niveau de détail immédiatement inférieur lorsqu'elle existe. A l'inverse, le *dévoilement* permet d'atteindre la représentation de niveau de détail immédiatement supérieur, lorsqu'elle existe.

La Figure 5.2 illustre les principes des fonctions de masquage et de dévoilement en reprenant l'exemple schématique introduit précédemment par la Figure 5.1.

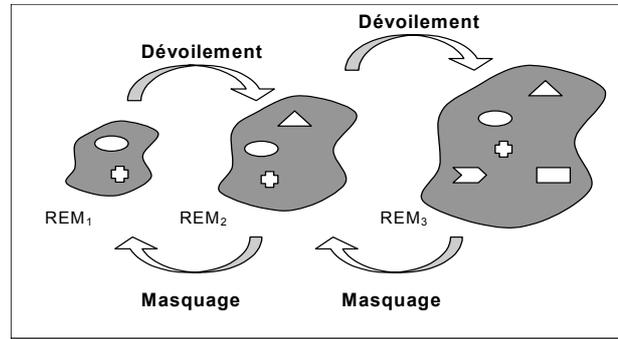


Figure 5.2 – Mécanismes de masquage et de dévoilement appliqués aux REM

Pour des REM extensionnelles ou intensionnelles, les fonctions de masquage et de dévoilement sont respectivement définies comme décrit ci-dessous. Nous supposons que les stratifications en extension $STRAT_{EXT}$ et en intension $STRAT_{INT}$ ont été définies pour l'Entité Masquable $EM \in \mathcal{EM}$ et qu'elles sont telles que :

$$REM_{EXT}(E(EM))_i \in STRAT_{EXT}(E(EM)) \quad \forall i \in [1, p] \text{ et } p = |STRAT_{EXT}(E(EM))| \text{ et}$$

$$REM_{INT}(E(EM))_j \in STRAT_{INT}(E(EM)) \quad \forall j \in [1, q] \text{ et } q = |STRAT_{INT}(E(EM))|$$

➤ Masquage extensionnel :

Soit M_{EXT} la fonction de masquage d'une Représentation d'Entité Masquable extensionnelle. M_{EXT} est définie par :

$$M_{EXT} : 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}}$$

$$M_{EXT}(REM_{EXT}(E(EM))_i) = REM_{EXT}(E(EM))_{i-1}, \text{ avec } 2 \leq i \leq p$$

➤ Masquage intensionnel :

Soit M_{INT} la fonction de masquage d'une Représentation d'Entité Masquable intensionnelle. M_{INT} est définie par :

$$M_{INT} : 2^{\mathcal{D}} \rightarrow 2^{\mathcal{D}}$$

$$M_{INT}(REM_{INT}(I(EM))_j) = REM_{INT}(I(EM))_{j-1}, \text{ avec } 2 \leq j \leq q$$

➤ Dévoilement extensionnel :

Soit D_{EXT} la fonction de dévoilement d'une Représentation d'Entité Masquable extensionnelle. D_{EXT} est définie par :

$$D_{EXT} : 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}}$$

$$D_{EXT}(REM_{EXT}(E(EM))_i) = REM_{EXT}(E(EM))_{i+1}, \text{ avec } 1 \leq i \leq p-1$$

➤ Dévoilement intensionnel :

Soit D_{INT} la fonction de dévoilement d'une Représentation d'Entité Masquable intensionnelle. D_{INT} est définie par :

$$D_{INT} : 2^{\mathcal{D}} \rightarrow 2^{\mathcal{D}}$$

$$D_{INT} (REM_{INT}(I(EM))_j) = REM_{INT}(I(EM))_{j+1}, \text{ avec } 2 \leq j \leq q-1$$

Dans la section suivante, nous décrivons le modèle qui sert de référence à la mise en œuvre de l'accès progressif dans une démarche de conception de SIW.

5.1.2 Modélisation UML du MAP

Nous proposons une description en UML du Modèle d'Accès Progressif (MAP). Les concepts introduits précédemment sont ici décrits en utilisant une notation étendue du langage UML qui constitue un *profil UML*. Les extensions proposées sont principalement basées sur des *stéréotypes* qui permettent d'étendre le vocabulaire d'UML et d'associer une nouvelle signification à un élément du modèle. Ils sont identifiés textuellement par une expression entre guillemets. Nous leur associons également des icônes spécifiques. Des *contraintes* permettent par ailleurs d'étendre la sémantique d'UML et de définir les *règles* qui doivent être vérifiées pour que le modèle soit bien formé. Dans le modèle proposé, les règles sont représentées par des notes stéréotypées « règle MAP » et exprimées en OCL.

La Figure 5.3 présente le profil UML que nous proposons pour la stratification en vue de l'accès progressif. On cherche donc ici à exprimer les règles de bonne formation des REM en utilisant le formalisme UML. Le MAP repose ainsi sur quatre classes stéréotypées et liées par des associations. La classe « Entité Masquable » (EM) est constituée d'au moins deux instances de la classe « Élément d'Entité Masquable » (EEM). Cette multiplicité est requise pour qu'une stratification de l'EM puisse être définie. Un même EEM peut être utilisé pour constituer plusieurs EM.

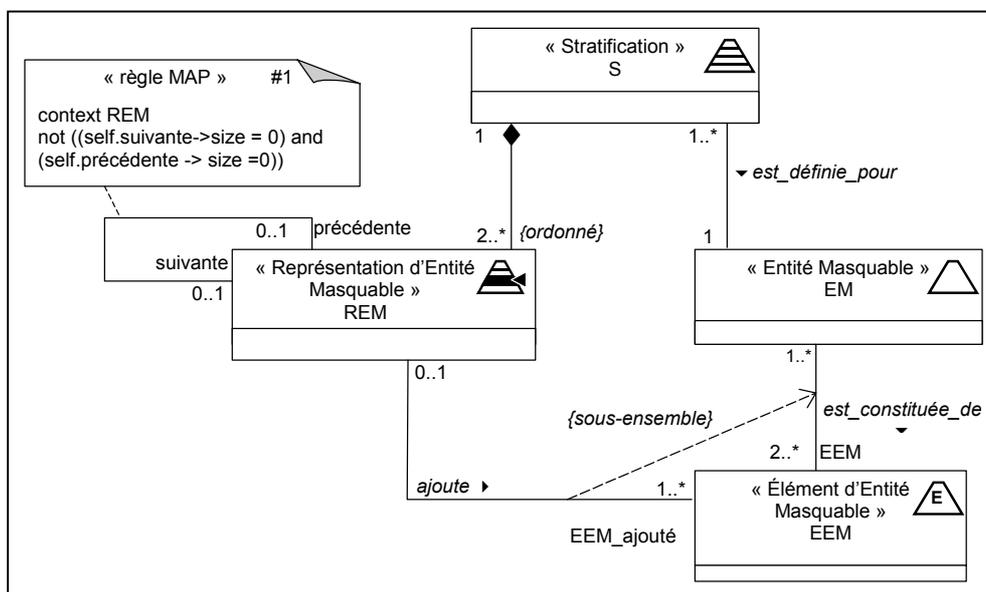


Figure 5.3 – Profil UML proposé pour la stratification en vue de l'accès progressif

Pour une EM, plusieurs stratifications peuvent être définies (classe « Stratification » S). En revanche, une seule stratification S est définie pour une et une seule EM. Une stratification S est liée par une agrégation à au moins deux REM (classe « Représentation d'Entité Masquable »),

celles-ci étant ordonnées (contrainte *{ordonné}*). La multiplicité et la contrainte d'ordre garantissent que la stratification respecte le sens que nous lui donnons.

Une REM est liée par une association à au moins un EEM, celui ou ceux qu'elle ajoute à sa REM précédente, si elle existe, pour représenter plus complètement encore l'EM. Un EEM est ajouté dans 0 ou 1 REM. La multiplicité de valeur 0 permet d'assurer une certaine confidentialité en laissant au besoin au moins un EEM toujours masqué. La multiplicité de valeur 1 permet de représenter, à travers l'association, la REM dans laquelle un EEM apparaît pour la première fois (*i.e.* la REM dans laquelle il est ajouté). Une relation de dépendance contrainte par *{sous-ensemble}* existe entre les associations *ajoute* et *est_constituée_de* (flèche en pointillés). Elle garantit que tous les EEM ajoutés par une REM sont forcément issus de l'ensemble des EEM constitutifs de l'EM, ces REM et EM étant associées à la même stratification S.

Une REM est liée à une autre REM par une association permettant de spécifier que l'une précède ou suit l'autre. La multiplicité maximum de l'association (portée à 1) garantit que pour toute REM il existe au plus une REM précédente et au plus une REM suivante. La contrainte #1 exprime le fait qu'une REM a forcément soit une REM précédente, soit une REM suivante, soit les deux. Une REM n'ayant qu'une REM suivante, et pas de REM précédente, est la représentation de EM correspondant au plus bas niveau de détail. Une REM n'ayant qu'une REM précédente, et pas de REM suivante, est la représentation de EM correspondant au plus haut niveau de détail.

Dans la section suivante, nous présentons le protocole d'exploitation du MAP qui complète la description donnée précédemment.

5.2. Protocole d'exploitation du MAP

Ce protocole définit les opérations que nous offrons à un concepteur qui souhaite mettre en œuvre une approche par accès progressif, ainsi que les opérations qui définissent le comportement à l'utilisation des stratifications. Il repose essentiellement sur les opérations des classes qui constituent le MAP, et parmi lesquelles nous distinguons :

- les opérations de définition d'une stratification, qui permettent au concepteur de créer et supprimer des stratifications.
- les opérations qui définissent les règles d'accès progressif, c'est-à-dire qui régissent l'accès aux informations stratifiées. On retrouve notamment ici les mécanismes de masquage et dévoilement des REM.
- les opérations de réorganisation d'une stratification dont l'intérêt est de mettre en œuvre diverses modifications d'une stratification existante.

La présentation de ces opérations s'appuie en partie sur une extension des diagrammes de cas d'utilisation. Celle-ci met en évidence les différents acteurs qui exploitent les opérations, ainsi que les éventuelles relations entre ces opérations. Nous désignons par *Cas d'Exploitation du MAP* (CEMAP), les cas d'utilisation particuliers qui décrivent comment est exploité ce modèle. Afin de les distinguer des cas d'utilisation traditionnels, nous étendons ces derniers au moyen du stéréotype « *CEMAP* » et proposons une icône équivalente (*cf.* Figure 5.4).

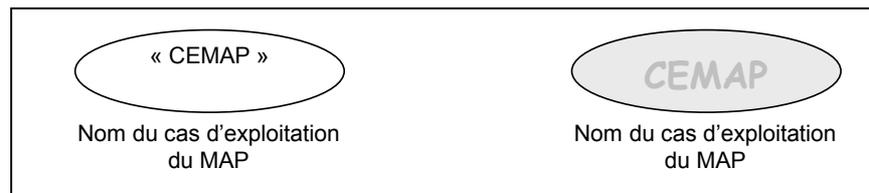


Figure 5.4 – Notations graphiques des cas d'utilisation pour le MAP
 À gauche, un cas d'utilisation étendu par le stéréotype « accès progressif », à droite l'icône proposée pour une représentation graphique équivalente.

Nous décrivons à présent plus précisément les différentes opérations représentées par les cas d'utilisation. Nous présentons pour certaines un diagramme de collaboration précisant les échanges de messages entre les classes impliquées.

5.2.1 Opérations de définition d'une stratification

Nous présentons successivement les opérations de création et de suppression d'une stratification. Ces opérations portent sur la classe « Stratification ». Elles impliquent certaines opérations d'autres classes du MAP.

5.2.1.1 Création d'une stratification

Une opération de définition d'une stratification appelée « créer une stratification » répond au CEMAP de création d'une stratification. Celle-ci implique de sélectionner une Entité Masquable (EM) et de créer les représentations correspondantes (REM). Nous supposons ici que l'EM existe et que son ensemble d'éléments constitutifs (EEM) est défini. Nous abordons ultérieurement les opérations de création d'une EM et de son ensemble d'EEM, celles-ci étant fortement liées à la nature même de l'EM (cf. section 5.3).

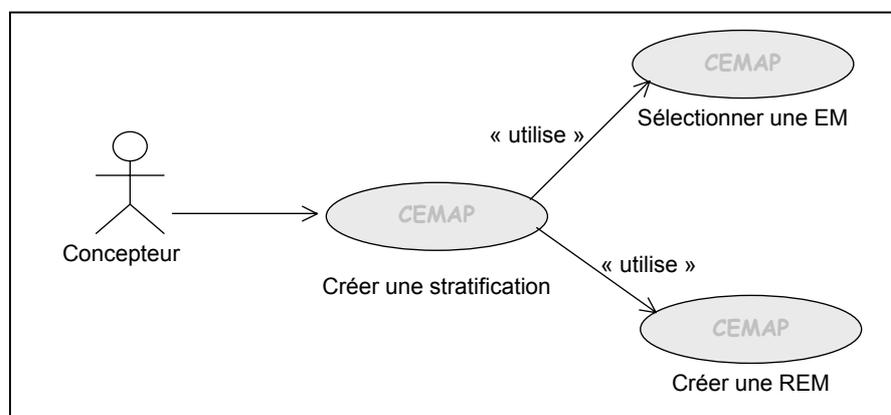


Figure 5.5 – Diagramme de Cas d'Exploitation du MAP : Création d'une Stratification

La création d'une stratification est engagée par un acteur que nous appelons *concepteur*. Nous soulignons que nous restons dans cette présentation à un niveau général (i.e. indépendant du contexte particulier de la conception d'un SIW donné). Il s'agit ici de mettre en évidence les principes majeurs d'exploitation du MAP. La Figure 5.6 présente le diagramme de séquence qui montre de façon plus détaillée les principes de la création d'une stratification par un concepteur.

Le concepteur demande la création d'une stratification *s* pour l'entité masquable *em* (passée en paramètre). Une fois créé, l'objet *s* établit un lien avec l'objet *em*. L'objet *em* établit la liste des éléments qui le constitue et la transmet à la stratification *s*.

Le concepteur procède ensuite à la définition des REM. L'accolade indique dans la figure la séquence de messages qui va être répétée pour chaque définition de REM. Cette demande de création de REM déclenche en retour l'envoi par l'objet *s* de la liste des EEM disponibles pour constituer la REM. Le concepteur choisit ceux qu'il souhaite ajouter dans la REM en cours de définition. L'objet *s* ordonne la création d'un objet *rem*, en transmettant le sous-ensemble d'EEM sélectionné par le concepteur. Une fois créé, l'objet *rem* établit la liaison avec les EEM indiqués. L'objet *s* procède à la mise à jour de l'ensemble des EEM disponibles : il supprime de l'ensemble ceux déjà liés à une REM. L'itération suivante de définition d'une REM peut être engagée.

Le processus de définition d'une REM se termine soit lorsque l'ensemble des EEM disponibles est vide, soit par arrêt explicite du concepteur avant ce terme : dans ce cas, les EEM non sélectionnés resteront toujours masqués dans la stratification. L'ordonnancement des REM est alors défini. Par défaut, il est établi en fonction de l'ordre dans lequel les REM ont été définies. C'est sur cette base que chaque REM établit un lien avec sa REM précédente et sa REM suivante. Nous verrons dans la section 5.2.3 que d'autres opérations permettent de modifier les liens de précedence ainsi établis.

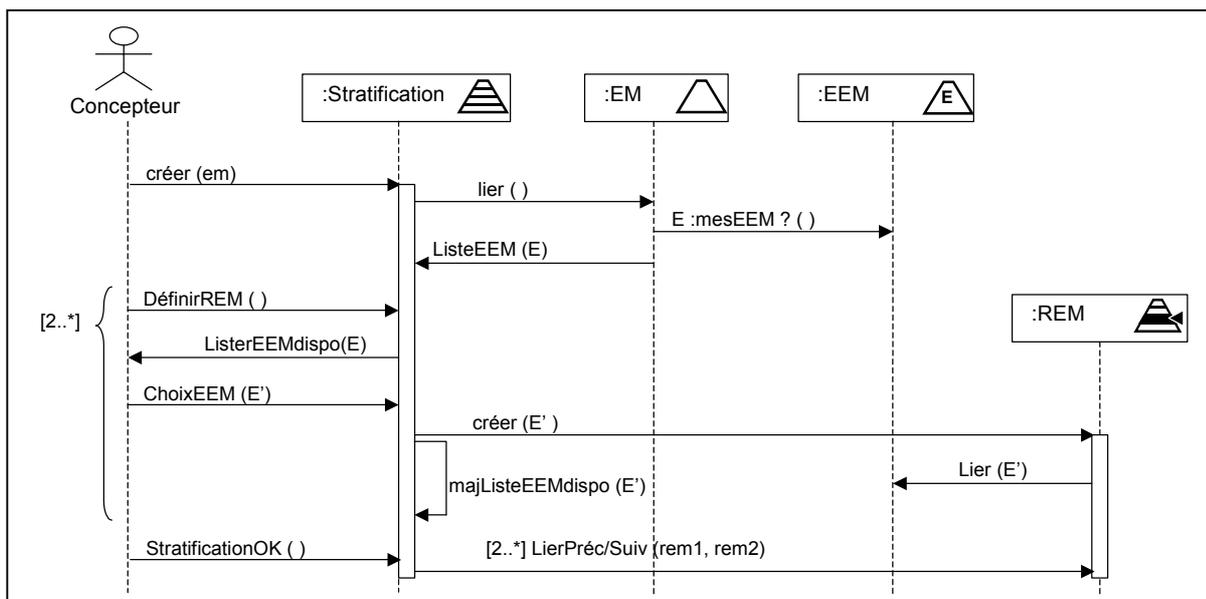


Figure 5.6 – Diagramme de séquence simplifié de la création d'une stratification

5.2.1.2 Suppression d'une stratification

L'autre opération présentée dans cette catégorie est celle de suppression d'une stratification. La suppression d'une stratification consiste à détruire l'objet *s* correspondant et tous les objets *rem* définis dans le cadre de celle-ci. Les liaisons dans lesquelles étaient impliqués ces objets sont en conséquence détruites (*i.e.* la liaison entre les objets *s* et *em* et celles unissant les objets *rem* et leurs objets *eem* respectifs). Notons que l'objet *em* et tous les objets *eem* ne sont pas affectés outre mesure par cette suppression.

5.2.2 Opérations d'accès progressif aux stratifications

Les opérations présentées dans cette section définissent les modalités d'accès progressif basées sur les spécifications du MAP. Elles sont présentées à travers le diagramme de cas d'exploitation du MAP de la Figure 5.7. Ce diagramme présente trois actions déclenchées par un utilisateur final :

- la première présente un cas d'utilisation, désigné par « *Utiliser une fonctionnalité du SIW* », qui nous permet d'introduire comment le MAP est utilisé dans le traitement d'une fonctionnalité selon la logique de l'accès progressif ;
- les deux autres actions concernent une demande explicite de masquage ou de dévoilement d'une REM par l'utilisateur. Elles montrent comment il exploite ainsi directement des opérations d'accès progressif.

Le cas d'utilisation « *Utiliser une fonctionnalité du SIW* » exprime volontairement un besoin de haut niveau. Bien qu'abstrait, ce cas est suffisant pour nous permettre de mettre en évidence les mécanismes d'accès progressif symbolisés par les CEMAP du diagramme. Nous considérons ici que, lorsque l'utilisateur fait appel à une fonctionnalité du SIW, celle-ci implique des données vues comme des entités masquables (des exemples sont donnés dans la section 5.3). Au lieu de considérer les données elles-mêmes, leurs représentations à différents niveaux de détail vont être utilisées. Par exemple, dans le cas d'une fonctionnalité de consultation d'information, l'information n'est pas livrée d'un seul tenant, mais présentée à un certain niveau de détail (i.e. une partie de l'information seulement est visible) à partir duquel l'utilisateur pourra exploiter les opérations de dévoilement et de masquage.

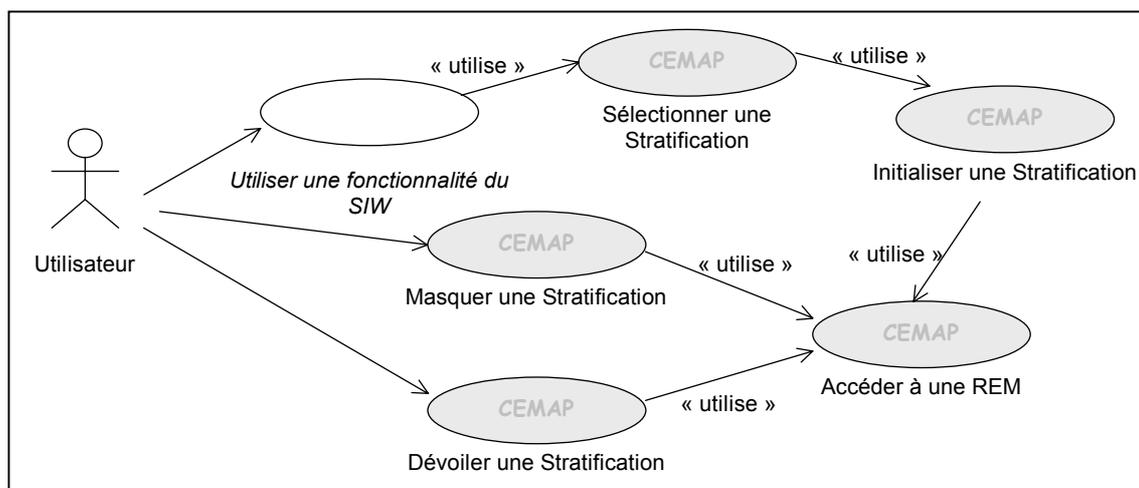


Figure 5.7 – Diagramme de Cas d'Exploitation du MAP : Opérations pour l'accès progressif

5.2.2.1 Sélection d'une stratification

La mise en œuvre d'un tel mécanisme débute par le recours à l'opération « *Sélectionner une Stratification* ». Cette opération consiste à trouver la stratification s concernant l'entité masquable em .

Lorsque la stratification s est identifiée, elle est ensuite initialisée. L'initialisation consiste à préparer la stratification en vue de l'accès progressif. Pour cela, nous introduisons deux attributs de la classe « *Stratification* » (cf. Figure 5.8):

- l'attribut *REMcourante* indique la représentation utilisée à un instant t. Par convention, lors de la sélection d'une stratification, la REM accédée est celle de plus bas niveau de détail. Soit, en OCL, la représentation *rem* telle que *rem.précédente -> size = 0* (i.e. il n'existe pas de REM précédente définie pour la REM *rem*).
- l'attribut *ensEEMvisible* correspond à l'ensemble des éléments d'entité masquable visibles à ce même instant. A l'initialisation, il est défini par le recours à l'opération « accéder à une REM » appelée pour *REMcourante*. Cette opération (désignée par *Accéder* pour simplifier) est définie par :

```

rem :: Accéder ( ):Set(EEM)
result=self.EEM_ajouté
-- le résultat retourné est l'ensemble constitué des objets de type EEM remplissant le rôle
EEM_ajouté dans l'association ajoute qui implique l'objet rem.
    
```

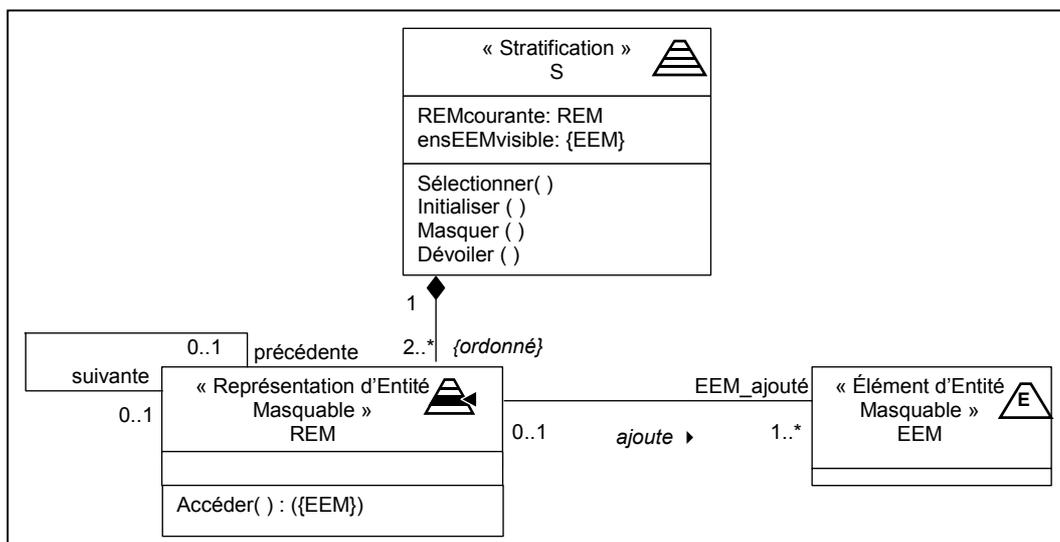


Figure 5.8 – Extrait du MAP: introduction des attributs et opérations des classes Stratification et REM

L'information délivrée à l'utilisateur est alors représentée à son plus bas niveau de détail. Il a alors la possibilité de procéder, dans un premier temps, au dévoilement de la stratification, puis ensuite d'utiliser également le masquage.

5.2.2.2 Masquage et dévoilement d'une stratification

Comme nous l'avons modélisé, une REM référence les EEM qu'elle ajoute au niveau de détail qui lui correspond. L'opération *Accéder()* retourne l'ensemble d'EEM correspondant. Les EEM visibles pour une REM sont ceux qu'elle ajoute *plus* ceux ajoutés par les REM qui la précèdent dans la stratification. Les opérations de masquage et dévoilement sont définies dans ce but. De plus, elles respectent le fait qu'il ne doit pas être possible de masquer la stratification lorsqu'on est déjà au plus bas niveau de détail et inversement, de la dévoiler au niveau maximal de détail. Les opérations *Masquer()* et *Dévoiler()* sont donc respectivement définies par :

```

s :: Dévoiler ()
pre : REMCourante.suivante -> size = 1
-- prérequis : la représentation courante n'est pas celle de plus haut niveau de détail
--(dans le cas contraire, la pré-condition n'étant pas vérifiée, l'opération n'est pas engagée).
    
```

```

REMCourante = REMCourante.suivante
-- la représentation courante devient la suivante
ensEEMvisible = ensEEMvisible ->union (REMCourante ::Accéder( ))
-- on ajoute à l'ensemble des EEM visibles ceux liés à la représentation courante

s :: Masquer ( )
pre : REMCourante.précédente -> size = 1
-- prérequis : la représentation courante n'est pas celle de plus bas niveau de détail
-- (dans le cas contraire, la pré-condition n'étant pas vérifiée, l'opération n'est pas engagée).
ensEEMvisible = ensEEMvisible -> différence (REMCourante ::Accéder( ))
-- on retire de l'ensemble des EEM visibles ceux liés à la représentation courante REMCourante
= REMCourante.précédente
-- la représentation courante devient la précédente
    
```

A ce stade, nous avons mis en évidence les opérations permettant de définir une stratification adaptée aux utilisateurs, ainsi que les opérations impliquées par la réalisation de l'accès progressif. Dans la section suivante, des opérations pour la réorganisation d'une stratification sont présentées.

5.2.3 Opérations de réorganisation d'une stratification

Les REM définies pour stratifier une EM peuvent être modifiées ou supprimées pour réorganiser la stratification. Les opérations de réorganisation d'une stratification permettent de la réadapter lorsque celle-ci ne répond plus exactement aux besoins en matière d'accès progressif. Nous proposons plusieurs opérations allant dans ce sens :

- Importation et suppression d'un EEM pour une REM donnée,
- Ajout et suppression d'une REM pour une stratification donnée,
- Fusion fractionnement et déplacement de REM dans une stratification donnée.

Leur présentation s'appuie sur un exemple de stratification donné par la Figure 5.9. L'entité masquable EM est stratifiée en quatre représentations REM1, REM2, REM3 et REM4, chacune ajoutant au moins un EEM de l'entité masquable. L'EEM g est totalement masqué.

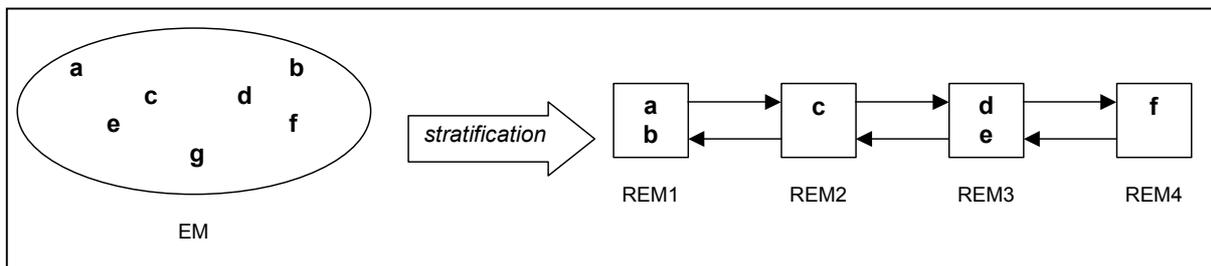


Figure 5.9 – Exemple de Stratification d'une EM en REM

5.2.3.1 Importation d'un EEM dans une REM

La première opération possible est désignée par « importer un EEM ». Elle consiste soit à ajouter un élément de EM non utilisé dans la stratification (par exemple l'EEM g), soit à déplacer un EEM d'une REM vers une autre. L'import d'un EEM non encore utilisé consiste simplement à lier cet EEM à la REM dans laquelle on souhaite le voir apparaître. Dans le cas

d'un déplacement d'EEM, on distingue deux cas, selon que l'EEM à déplacer est ou non l'unique élément référencé par la REM à laquelle il était affecté jusqu'ici. Si tel est le cas (par exemple, lorsqu'il s'agit d'importer les EEM *c* ou *f*), l'opération de fusion, présentée plus loin, est concernée. Si l'EEM à importer n'est pas unique dans sa REM d'origine (par exemple, importer *a*, *b*, *d* ou *e*), alors l'opération consiste à supprimer le lien entre l'EEM et sa REM d'origine et à établir une nouvelle liaison entre l'EEM et la REM pour laquelle est réalisé l'import.

5.2.3.2 Suppression d'un EEM d'une REM

La suppression d'un EEM de la REM à laquelle il est associé peut consister à le déplacer vers une autre REM. Comme nous l'avons vu précédemment, la suppression correspond donc à un import d'EEM ou à une fusion de REM. Cependant, la suppression d'un EEM peut signifier qu'on souhaite masquer toujours cet élément. Dans ce cas, le lien qui associait l'EEM à une REM est supprimé et non remplacé par un autre. Lorsque cet EEM est le seul élément d'une REM, l'opération de suppression de l'EEM revient à supprimer la REM comme nous le verrons dans la suite.

5.2.3.3 Ajout d'une REM

Une stratification peut être réorganisée en définissant une nouvelle REM et en :

- lui associant des EEM non utilisés dans la stratification (par exemple l'EEM *g*),
- important des EEM d'autres REM comme vu précédemment,
- fractionnant une REM existante, ce que nous présentons ultérieurement.

En ce qui concerne les deux premiers cas, il est nécessaire de placer la nouvelle REM dans la stratification en définissant les liens de précedence et succession adéquats. Ceci est fait par le recours à une opération d'insertion d'une REM dans la stratification. Cette opération consiste à choisir une REM donnée, à indiquer si l'insertion doit se faire avant ou après celle-ci et réorganiser les liens.

Supposons que l'on souhaite ajouter une REM à la stratification donnée en exemple. Cette REM permet d'intégrer l'EEM *g* et doit être insérée après la REM1 (cf. Figure 5.10).

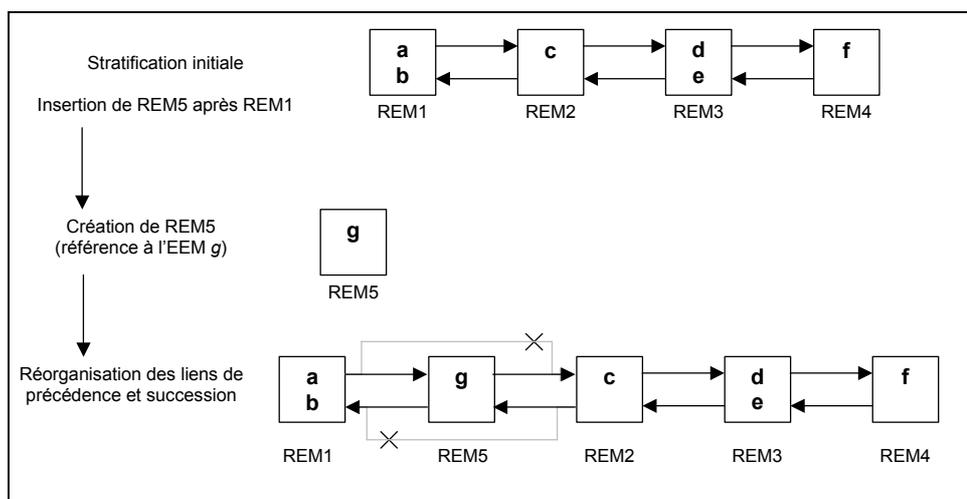


Figure 5.10 – Réorganisation d'une stratification par ajout d'une nouvelle REM

Le lien de précedence de REM2 vers REM1 est annulé et remplacé par un lien de précedence de REM2 vers REM5, et un autre de REM5 vers REM1. De façon symétrique, on substitue au lien de succession de REM1 vers REM2 deux liens de même nature de REM1 vers REM5, et de REM5 vers REM2. Lorsqu'une REM doit être ajoutée en début ou en fin de stratification, la réorganisation des liens implique de créer les liens de précedence et succession permettant de rattacher la nouvelle REM à l'ancienne REM de début ou de fin de stratification.

Les mécanismes de mise à jour des liens dans une stratification sont tout à fait similaires à ceux employés dans le cas des *listes doublement chaînées*. Une stratification peut en effet être représentée par une telle structure de données. Les autres cas de réorganisation d'une stratification présentés dans la suite font également appel à de tels mécanismes.

5.2.3.4 Suppression d'une REM

Une stratification peut être réorganisée par suppression d'une REM. La suppression d'une REM consiste à détruire les liens qui unissaient les EEM à cette REM et à réorganiser les liens de précedence et succession. Cette réorganisation correspond au processus inverse de celui décrit dans le cas de l'ajout d'une REM.

Lorsqu'une REM est supprimée, soit les EEM qu'elle référençait sont déplacés vers d'autres REM, soit ils ne sont plus associés à aucune REM. Le premier cas revient à procéder à des imports d'EEM, ou à une fusion de REM lorsque tous les EEM sont déplacés vers la même REM. L'opération de fusion est présentée ci-dessous.

5.2.3.5 Fusion de REM

L'opération de fusion consiste à réunir deux ou plusieurs REM pour n'en faire qu'une. L'opération ne peut toutefois impliquer que $n-1$ REM sur les n qui constituent la stratification afin de garantir qu'au minimum deux représentations à des niveaux de détail différents existent pour l'EM. La fusion peut être vue comme un import de *tous* les EEM d'une REM dans une autre et couvre ainsi le cas de l'import de l'élément unique d'une REM (e.g. les EEM c et f). Procéder à une fusion a, comme l'ajout ou la suppression de REM, un impact sur le nombre de REM et sur leur ordonnancement (i.e. sur les relations de précedence et de succession entre REM de la stratification). Nous définissons le cadre suivant pour l'opération de fusion :

- la fusion implique une REM *dominante* et une ou plusieurs REM *subordonnées*. Tous les EEM des REM subordonnées vont être affectés à la REM dominante. C'est à partir de la position de la REM dominante dans la stratification initiale qu'est déterminée la place de la REM résultant de la fusion.
- Pour chaque REM subordonnée, un processus identique s'applique : la REM dominante établit un lien avec chaque EEM de la REM subordonnée. L'opération de suppression de la REM subordonnée est engagée : les deux REM qui encadraient la REM subordonnée dans la stratification se lient directement : la REM précédente de la REM subordonnée se lie à la REM suivante, et à la REM suivante se lie à la précédente. La REM subordonnée n'est plus reliée dans la stratification et peut être détruite.

A partir de la stratification donnée précédemment, supposons que l'on souhaite fusionner *REM1* et *REM3*, avec *REM1* comme REM dominante. Les différentes étapes du processus sont illustrées dans la Figure 5.11 ainsi que la stratification qui résulte de la fusion.

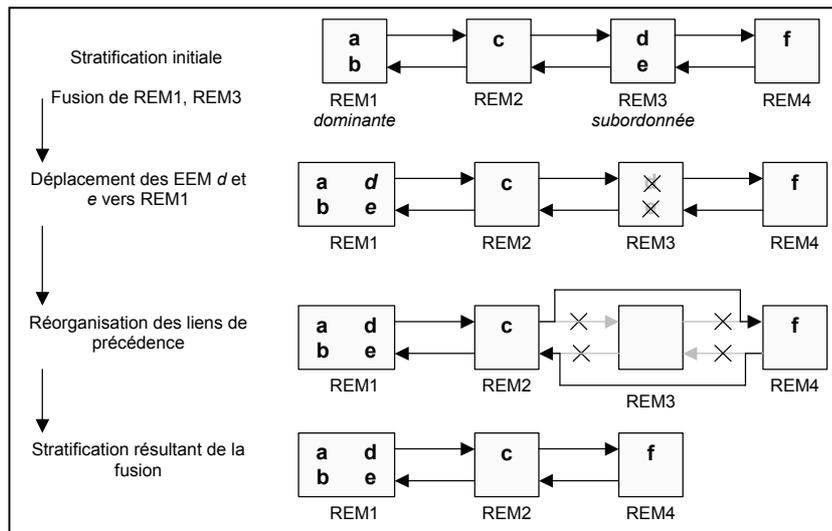


Figure 5.11 – Exemple de réorganisation d'une stratification par fusion de REM

5.2.3.6 Fractionnement d'une REM

En vue de la réorganisation d'une stratification, l'opération inverse de la fusion peut être utilisée. L'opération de fractionnement consiste en effet, à partir d'une REM donnée, également appelée REM dominante, à créer une ou plusieurs autres REM. Dans ce cas, les éléments de la REM dominante sont répartis dans d'autres REM créées à cet effet. L'opération de fractionnement est basée sur l'opération de définition d'une REM présentée dans la section 5.2.1. Cependant, le processus en jeu lors d'un fractionnement diffère dans la mesure où l'ensemble des EEM disponibles pour la création de REM supplémentaires correspond ici à l'ensemble des EEM associés à la REM dominante²⁵. Pour qu'un fractionnement puisse être engagé, cet ensemble doit être constitué d'au moins deux EEM. Au moins un EEM doit rester référencé par la REM dominante.

Suite à la création des REM par fractionnement, celles-ci sont ordonnées par définition des liens de précedence et succession adéquats, la REM dominante étant la première. Le lien de succession qui liait la REM dominante au reste de la stratification originale est reporté sur la dernière REM issue du fractionnement. De façon symétrique, le lien de précedence de la REM qui suivait la REM dominante avant le fractionnement pointe sur la dernière REM issue du fractionnement. La Figure 5.12 illustre une réorganisation de stratification par fractionnement d'une REM. Les liens de la REM dominante avec la REM qui la précède dans la stratification reste inchangés lors d'un fractionnement.

²⁵ Dans le processus de définition d'une REM pour créer une stratification, cet ensemble est constitué des EEM associés à l'EM à stratifier.

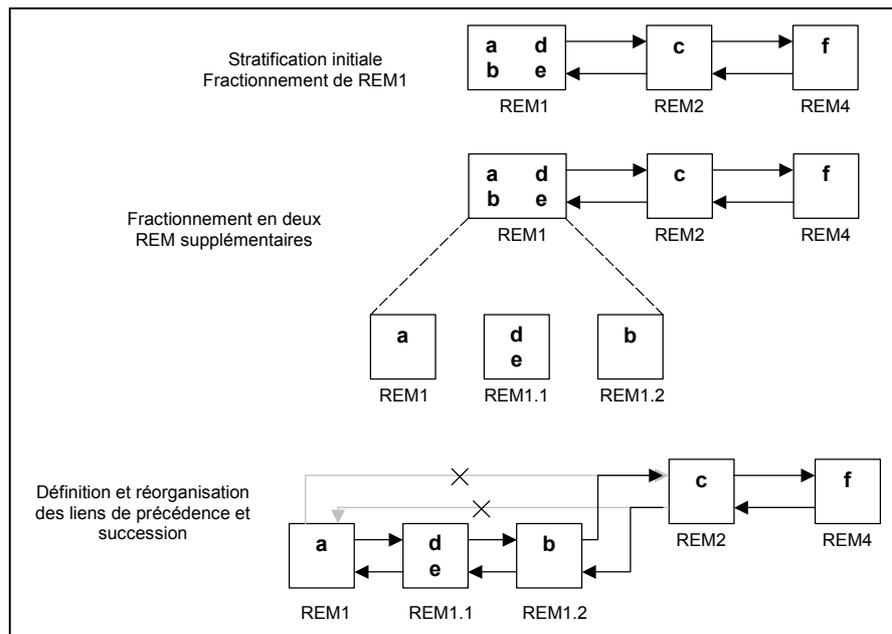


Figure 5.12 – Exemple de réorganisation d’une stratification par fractionnement d’une REM

5.2.3.7 Déplacement de REM

L’opération de déplacement de REM consiste à modifier l’ordre des REM dans la stratification sans pour autant changer la structure interne des REM (i.e. les EEM qu’elles ajoutent). Le processus de déplacement d’une REM repose sur le même mécanisme que celui d’insertion d’une REM présenté à l’occasion de l’ajout d’une REM.

Nous avons présenté le MAP à travers un diagramme de classes UML étendues au moyen de stéréotypes spécifiques à l’Accès Progressif. Le protocole d’exploitation du MAP décrit ensuite a mis en évidence les différentes opérations liées à l’utilisation du modèle. Dans la section suivante, nous montrons comment le MAP et ses principes peuvent s’appliquer à un modèle de données exprimé au moyen d’un diagramme de classes UML.

5.3. Application du MAP à un modèle de données

L’objectif de cette section est de montrer comment nous exploitons les concepts et principes du MAP en vue de proposer un Accès Progressif au contenu d’un modèle de données. Nous choisissons un modèle exprimé au moyen d’un formalisme objet. Plus précisément, nous considérons un modèle de données représenté par un diagramme de classes UML. Cependant, il convient de souligner que la prise en compte de l’approche utilisant le MAP est envisageable dans un contexte de représentation de connaissances bien plus large, qui peut s’appuyer sur un modèle structuré ou non, par exemple sur un modèle objet, relationnel, basé sur XML, etc. Pour illustrer l’applicabilité du MAP à un modèle de données particulier, dans [Vill02b], nous présentons une application du MAP à un modèle de données décrit à l’aide du formalisme objet du Système de Représentation de Connaissances AROM²⁶ [Page00] dont les entités de représentation présentent quelques variantes par rapport à celles présentes dans un diagramme de classes UML. Cette application est à percevoir comme l’une des instantiations possibles du MAP générique que nous avons présenté.

²⁶ AROM : Associer des Relations et des Objets pour Modéliser

Le modèle de données décrit par la Figure 5.13 est utilisé comme exemple dans cette section. Ce modèle correspond à un domaine d'application portant sur la *gestion documentaire d'inondations* (nous désignerons dans la suite ce modèle par GDI). Il traduit le fait que des Documents concernant des Inondations sont analysés par des Spécialistes. Un spécialiste peut être un Géographe ou un Historien. L'analyse d'un document par un spécialiste donne lieu à un Rapport. Les attributs des classes sont simplement nommés pour plus de simplicité (les types, domaines, valeurs initiales, etc. sont omis). Nous ne donnons pas de description des opérations des classes, celles-ci n'étant pas prises en compte dans l'approche basée sur l'accès progressif.

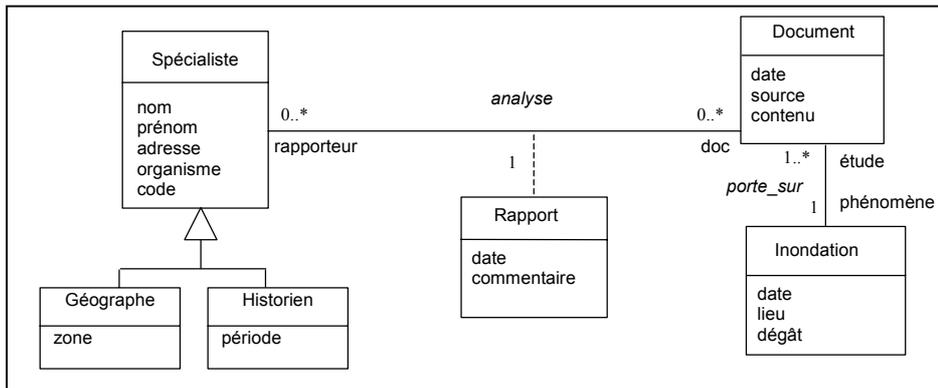


Figure 5.13 – Exemple de modèle de données : Gestion Documentaire d’Inondations

Nous donnons également quelques exemples d’instances des classes et classes-associations correspondant au modèle GDI. Ces exemples sont présentés dans les tableaux de la Figure 5.14.

| Spécialiste | | | | | |
|-------------|-----------|--------|--|-----------|-------|
| | nom | prénom | adresse | organisme | code |
| OS1 | JEAN | Loïc | 1 Avenue Desolivers 38100 Grenoble | CFRR | 02.71 |
| OS2 | MARIGNIER | Hugo | 56 Chemin Gallina 74000 Annecy | AFTR | 10.98 |
| OS3 | COPS | Flora | 74 Allée Vittet 38100 Grenoble | GUCN | 06.74 |
| OS4 | PETIT | Arthur | 12 Rue S. et C. Albert 38100 Grenoble | CPCC | 07.02 |

| Géographe | | | | | | |
|-----------|-------|----------|-------------------------------------|-----------|-------|------|
| | nom | prénom | adresse | organisme | code | Zone |
| OG1 | BRUNO | Sylviane | 19 Rue de la Vallée 74000 Annecy | AIB | 10.69 | 31J |
| OG2 | HECK | Loïs | 76 Route Dufresne 74000 Annecy | ENPML | 03.95 | 14B |

| Historien | | | | | | |
|-----------|---------|----------|--------------------------------------|-----------|-------|---------|
| | nom | prénom | adresse | organisme | code | période |
| OH1 | FOREST | Brigitte | 56 Rue Divonne 38100 Grenoble | SRACA | 11.52 | 12-16 |
| OH2 | ROLANDE | Jean-Luc | 28 Chemin Guillot 69000 Lyon | CMM | 08.43 | 17-20 |
| OH3 | MARTEL | Hervé | 3 Rue Hilaire 38000 Grenoble | T&M | 11.61 | 18-21 |
| OH4 | GENSIN | Jérôme | 19 Chemin Chantant 38000 Grenoble | SINK | 08.65 | 13-15 |

| Document | | | |
|----------|------------|-----------------------|---------|
| | date | source | contenu |
| OD1 | 28/07/1901 | Archives Mairie 74000 | |
| OD2 | 01/02/1771 | Société ActhYs | ... |

| Inondation | | | |
|------------|------------|---------|-------|
| | date | lieu | dégât |
| OI1 | 25/07/1901 | Bornand | |
| OI2 | 29/01/1771 | Isère | ... |
| OI3 | 14/10/1894 | Drac | ... |

| Rapport | | | | |
|---------|------------|-----|------------|-------------|
| | rapporteur | doc | date | commentaire |
| IC1 | OS2 | OD1 | 16/09/2002 | ... |
| IC2 | OS3 | OD1 | 12/10/2002 | ... |
| IC3 | OS3 | OD2 | 17/10/2001 | ... |
| IC4 | OG1 | OD1 | 22/07/2001 | ... |
| IC5 | OH1 | OD2 | 14/03/2002 | ... |

| Porte_sur | | |
|-----------|-------|-----------|
| | étude | phénomène |
| TP1 | OD1 | OI1 |
| TP2 | OD2 | OI2 |

Figure 5.14 – Exemples d’instances et de tuples définis pour le modèle de données ‘GDI’

5.3.1 Descriptions en intension et en extension du modèle de données

Dans cette section, nous décrivons un modèle de données représenté par un diagramme de classes UML et par les différents éléments de modélisation (appelés *constituants*) qu'il contient : les classes, associations et classes-associations. Nous cherchons ici à identifier dans ce modèle de données, les différentes entités de représentation qui peuvent constituer des entités masquables. Nous considérons qu'il est possible de mettre en œuvre un accès progressif pour le modèle de données lui-même, et pour chacun de ses constituants : classes, associations ou classes-associations. Pour chacune des Entités Masquables potentielles offertes par un modèle de données UML, nous donnons son *extension*. Cette description permettra de déterminer la nature des Représentations d'Entité Masquable extensionnelles associées à chaque Entité Masquable considérée. Lorsque l'Entité Masquable le permet, nous donnons l'*intension* de cette Entité Masquable. Cette description permettra de déterminer la nature des Représentations d'Entité Masquable intensionnelles associées.

- Soit un 'Modèle de Données', désigné par MD , traduisant la sémantique du domaine d'application.

Le Modèle de Données est ici vu comme l'ensemble des classes, associations et classes-associations qui le composent. Les constituants d'un Modèle de Données n'ont pas tous la même structure, aussi nous ne considérons pas ici l'intension du Modèle de Données. Ceci signifie qu'en tant qu'ensemble, il pourra être associé au Modèle de Données des Représentations d'Entité Masquable extensionnelles, mais pas de Représentations d'Entité Masquable intensionnelles.

La description en extension de MD , notée $E(MD)$, est donnée par un ensemble dont les éléments sont les extensions de toutes les classes C_i , les extensions de toutes les associations A_j et les extensions de toutes les classes-associations CA_k décrites dans le diagramme de classes associé à MD . D'où :

$$E(MD) = \{E(C_1), \dots, E(C_a), E(A_1), \dots, E(A_b), E(CA_1), \dots, E(CA_c)\}$$

avec $E(C_i)$ est l'extension de la classe C_i pour $1 \leq i \leq a$,
 $E(A_j)$ est l'extension de l'association A_j pour $1 \leq j \leq b$,
 $E(CA_k)$ est l'extension de la classe-association CA_k pour $1 \leq k \leq c$.

Dans l'exemple du modèle de données GDI (cf. Figure 5.13), $E(GDI)$ est l'ensemble des extensions des classes Spécialiste, Géographe, Historien, Document, et Inondation, des associations analyse et porte_sur, et de la classe-association Rapport, c'est-à-dire à l'ensemble des ensembles d'instances des classes, des associations et de la classe-association mentionnées (cf. Figure 5.14).

- Soit une 'Classe', désignée par C , qui décrit un concept du domaine dans le Modèle de données,
 - l'intension de C , notée $I(C)$, est un ensemble dont les éléments sont les descriptions²⁷ des n attributs att_i , avec $1 \leq i \leq n$, de la classe C . D'où :

$$I(C) = \{att_1 : t_1, \dots, att_n : t_n\}$$

avec $att_i : t_i$ ($1 \leq i \leq n$) description de l'attribut att_i de C et t_i type associé à att_i et le type de la classe C est donné par le type-record [Card91] $\langle att_1 : t_1, \dots, att_n : t_n \rangle$

²⁷ Nous limitons ici volontairement la description de l'attribut à son type, mais celle-ci peut comporter plus de spécifications (domaine, valeur initiale, etc.).

- l'extension de C , notée $E(C)$, est un ensemble dont les éléments sont les objets (instances) O_i de la classe C . Ainsi :

$$E(C) = \{ O_1, \dots, O_h \}$$

avec O_i ($1 \leq i \leq h$) est une instance de la classe C construite conformément à l'intension de C . Si le type de la classe C est le type-record [Card91] $\langle att_1 : t_1, \dots, att_n : t_n \rangle$, alors O_i , instance de C , peut être vue comme la record-value $\langle att_1 = v_1, \dots, att_n = v_n \rangle$, avec v_i la valeur de type t_i associé à att_i

Dans l'exemple du modèle de données GDI, l'intension de la classe Spécialiste du modèle GDI est constituée des descriptions des attributs nom, prénom, adresse, organisme et code (cf. Figure 5.13). L'extension de la classe Spécialiste est donnée par l'ensemble de ses instances, i.e. les objets OS1, OS2, OS3, OS4 (cf. Figure 5.14).

- Soit une 'Association', désignée par A , qui décrit une relation entre classes du domaine dans le Modèle de données ,

- l'intension de A , notée $I(A)$, est un ensemble dont les éléments sont les descriptions des rôles r_i d'une association A d'arité m . On définit :

$$I(A) = \{ r_1 : t_1, \dots, r_m : t_m \}$$

avec $r_i : t_i$ ($1 \leq i \leq m$) est la description du rôle r_i de A et t_i le type record associé à la classe C_i liée à l'association A par le rôle r_i .

- l'extension de A , notée $E(A)$, est un ensemble dont les éléments sont les instances d'association (que nous rebaptisons *tuples*) TU_j de l'association. Ainsi :

$$E(A) = \{ TU_1, \dots, TU_p \}$$

avec TU_j ($1 \leq j \leq p$) est un tuple de A construit conformément à l'intension de A :

Si le type de l'association est le type-record [Card91] $\langle r_1 : t_1, \dots, r_m : t_m \rangle$, alors TU_i , tuple de A , peut être vu comme la record-value $\langle r_1 = v_1, \dots, r_m = v_m \rangle$, avec v_i ($1 \leq i \leq m$) instance de la classe C_i liée à l'association A par le rôle r_i

Dans l'exemple du modèle de données GDI, l'association porte_sur du modèle GDI a comme intension un ensemble constitué des descriptions des rôles étude (qui est de type Document) et phénomène (qui est de type Inondation) (cf. Figure 5.13). L'extension de cette association est donnée par les tuples instanciant ces rôles, i.e. TP1 et TP2 (cf. Figure 5.14).

- Soit une 'Classe-Association', désignée par CA , qui, dans le Modèle de données, décrit une relation entre des classes du domaine et possède des attributs,

- l'intension de CA , notée $I(CA)$, est donnée par un ensemble dont les éléments sont les descriptions des rôles r_j et des attributs att_k de la classe-association CA . Ainsi:

$$I(CA) = \{ r_1 : t_1, \dots, r_s : t_s, att_1 : ta_1, \dots, att_z : ta_z \}$$

avec $r_j : t_j$ ($1 \leq j \leq s$) est la description du rôle r_j de CA et t_j le type de la classe liée à la classe-association CA par le rôle r_j et $att_k : t_k$ ($1 \leq k \leq z$) est la description de l'attribut att_k de CA et ta_k est le type associé à att_k

- l'extension de CA , notée $E(CA)$, est un ensemble dont les éléments sont les instances (que nous rebaptisons instances complexes) IC_i de la classe d'association CA . Ainsi :

$$E(CA) = \{IC_1, \dots, IC_r\}$$

avec IC_l ($1 \leq l \leq r$) est une instance complexe de CA construite conformément à l'intension de CA .

Si le type de la classe-association est le type-record [Card91] $\langle r_1 : t_1, \dots, r_s : t_s, att_1 : ta_1, \dots, att_z : ta_z \rangle$, alors IC_l , instance complexe de CA , peut être vue comme la record-value $\langle r_1 = v_1, \dots, r_s = v_s, att_1 = w_1, \dots, att_z = w_z \rangle$, avec v_j ($1 \leq j \leq s$) instance de la classe C_j liée à la classe-association CA par le rôle r_j et w_k ($1 \leq k \leq z$) la valeur de type ta_k associé à l'attribut att_k .

Dans l'exemple du modèle de données GDI, la classe association Rapport a comme intension un ensemble dont les éléments sont les descriptions des rôles rapporteur (qui est de type Spécialiste) et doc (qui est de type Document), et les descriptions des attributs date et commentaire (cf. Figure 5.13). On note que, dans notre approche, une classe-association attachée à une association assimile les informations portées par l'association : c'est le cas pour la classe-association Rapport qui intègre les informations sur les rôles de l'association analyse. L'extension de cette classe-association est donnée par les instances complexes conformes à cette définition, i.e. IC1 à IC5 (cf. Figure 5.14).

Nous avons exposé la façon dont sont décrits en intension et en extension un modèle de données, décrit par un diagramme de classes UML, et ses constituants. Cette double définition nous permet d'envisager la spécification de deux types d'accès progressif. Cette question est abordée dans la section suivante où nous établissons des correspondances entre le modèle de données (et ses constituants) et les concepts du MAP.

5.3.2 Stratifications applicables à un modèle de données

5.3.2.1 Stratifications en intension *versus* en extension

Lors de la présentation du MAP et de ses concepts (cf. section 5.1.2), une Entité Masquable (EM) a été présentée comme un ensemble d'éléments qu'il est possible de stratifier en différentes représentations (REM). Nous avons alors souligné que la vision ensembliste retenue pour une EM pouvait conduire à considérer deux types de REM, extensionnelle ou intensionnelle, selon que les éléments de EM possèdent la même structure ou non. En tant qu'ensembles, le modèle de données et ses constituants (i.e. classe, association ou encore classe-association) peuvent être considérés comme des EM.

Ces EM peuvent donc faire l'objet de stratifications en intension (à l'exception du modèle de données) et en extension. La sémantique associée à chacun des deux types de stratification diffère : la stratification en intension donne lieu à des représentations à différents niveaux de détail *de la structure* de l'EM. La stratification en extension offre des représentations plus ou moins complètes *de la population* (l'ensemble des *individus* ou éléments) de l'EM. Ces deux types de stratification portent donc sur des dimensions différentes et complémentaires du modèle de données: structure et population. Bien qu'elles partagent la même finalité : *permettre un accès progressif au contenu véhiculé par le Modèle de données*, stratification en intension et stratification en extension expriment des possibilités différentes, s'appliquant à des cas différents :

- Lorsque l'extension est stratifiée, la population (c'est-à-dire le contenu lui-même) est répartie en différents groupes (sous-ensembles d'individus ou d'éléments) ordonnés par inclusion ensembliste. L'accès progressif se traduit par le fait de délivrer le contenu en 'montrant' plus ou moins de groupes, et donc plus ou moins d'individus, à chaque niveau de détail. L'intérêt d'une approche par stratification en extension apparaît surtout pour les

applications dont le contenu informationnel est dense, en termes d'instances et de tuples, lesquels présentent des degrés d'intérêt divers pour l'utilisateur en fonction de leur valeur.

- Lorsque l'intension est stratifiée, la structure décrivant le contenu de l'entité masquable est répartie en différentes *sous-structures* (sous-ensembles de descriptions d'attributs de l'entité masquable) ordonnés par inclusion ensembliste. L'approche par stratification en intension se révèle utile pour les applications dont le modèle de données comporte de nombreux concepts, ceux-ci étant, de plus, très structurés et tels que tous leurs éléments de structuration (*i.e.* tous les attributs) ne sont pas d'égal intérêt pour l'utilisateur.

5.3.2.2 Combinaison de stratifications

Les deux types de stratifications présentés ont deux objectifs distincts : la stratification en extension agit sur l'extension en masquant à certains niveaux (dans certaines REM) des éléments de l'EM, l'autre agit sur la stratification en intension en masquant à certains niveaux (dans certaines REM) des attributs de la structure de l'EM.

Cependant, il est possible de combiner ces deux stratifications. En effet, en considérant, comme il a été défini dans la section 5.1.1.3, les REM et les stratifications comme des fonctions, il est possible de composer ces fonctions. Ces compositions de fonctions, quelle que soit leur complexité, peuvent se ramener à deux cas de base :

1. On applique une stratification en intension $STRAT_{INT}$ sur un élément d'une stratification en extension d'une entité masquable : $REM_{INT}(E(EM))$. On opère donc ici :

$$STRAT_{INT}(I(E^{-1}(REM_{EXT}(E(EM)))))) = \{ \\ REM_{INT}(I(E^{-1}(REM_{EXT}(E(EM))))))_1, \\ REM_{INT}(I(E^{-1}(REM_{EXT}(E(EM))))))_2, \dots, \\ REM_{INT}(I(E^{-1}(REM_{EXT}(E(EM))))))_q \}$$

L'idée est de commencer par masquer certains individus de la population de l'EM (en la stratifiant en extension), puis à partir d'une des REM extensionnelles, c'est-à-dire à un certain niveau de masquage de la population, masquer l'intension de l'EM. A un niveau donné de cette stratification en intension, on décrit par les attributs d'une REM, les individus rendus accessibles par la REM en extension choisie.

2. On applique une stratification en extension $STRAT_{EXT}$ sur un élément d'une stratification en intension d'une entité masquable : $REM_{INT}(E(EM))$. On opère donc ici :

$$STRAT_{EXT}(E(I^{-1}(REM_{INT}(I(EM)))))) = \{ \\ REM_{EXT}(E(I^{-1}(REM_{INT}(I(EM))))))_1, \\ REM_{EXT}(E(I^{-1}(REM_{INT}(I(EM))))))_2, \dots, \\ REM_{EXT}(E(I^{-1}(REM_{INT}(I(EM))))))_q \}$$

L'idée est de commencer par masquer certains attributs de la structure de l'EM (en la stratifiant en intension), puis à partir d'une des REM intensionnelles, c'est-à-dire à un certain niveau de masquage de la structure, masquer l'extension de l'EM. A un niveau donné de cette stratification en extension, on accède alors aux individus d'une REM, décrits par les attributs rendus accessibles par la REM en intension choisie.

Par itération des exemples précédents, une stratification en intension (respectivement en extension) peut être appliquée à plusieurs REM d'une stratification en extension (respectivement

en intension). Par suite, l'application d'une stratification en intension à *toutes* les REM d'une stratification en extension permet de bénéficier des mêmes mécanismes de masquage et de dévoilement sur la structure à tous les niveaux de détail de l'extension. De façon similaire, l'application d'une stratification en extension à toutes les REM d'une stratification en intension permet, à tous les niveaux de détail donnant accès progressivement à la structure, de bénéficier des mêmes mécanismes de masquage et de dévoilement sur la population.

Dans ce qui suit, nous traitons séparément stratification en intension et stratification en extension afin de simplifier la présentation. Nous présentons les principes des deux approches, en supposant que le choix exclusif de l'une ou de l'autre a été fait pour mettre en œuvre l'accès progressif. La stratification en intension d'une EM est traitée dans la section 5.3.3 alors que sa stratification en extension est abordée dans la section 5.3.4.

5.3.3 Stratification en intension d'une EM

Nous déclinons ci-dessous les différents cas que nous considérons dans le but de proposer un accès progressif basé sur la stratification en intension des EM. Pour chaque EM potentielle, nous mettons en évidence comment ses REM sont construites, et ce, en respect de la sémantique du MAP d'une part, et sur la base des descriptions intensionnelles décrites lors de la section 5.3.1 d'autre part.

5.3.3.1 Description

Soit une classe C d'un modèle de données 'MD' décrit par un diagramme de classes UML comportant n attributs att_i , avec $1 \leq i \leq n$. La classe C est une EM dont l'intension est à stratifier. Nous appelons $I(C)$ l'intension de l'Entité Masquable 'Classe C ' considérée et, $STRAT_{INT}(I(C))$, l'ensemble des REM intensionnelles qui définissent la stratification de l'intension de C . Chaque REM intensionnelle $REM_{INT}(I(C))$ de $STRAT_{INT}(I(C))$ est constituée de descriptions d'attributs choisis parmi l'ensemble définissant l'intension de C , i.e. $I(C) = \{att_1 : t_1, \dots, att_n : t_n\}$. Ainsi, chaque $REM_{INT}(I(C))$ donne accès à un plus ou moins grand nombre d'attributs de la classe. La Figure 5.15a donne une interprétation de cette définition dans les termes du MAP.

De façon similaire, considérons une association A d'un modèle de données 'MD' décrit par un diagramme de classes UML comportant m rôles r_i , avec $1 \leq i \leq m$. L'association A est une EM dont l'intension est à stratifier. Nous appelons $I(A)$ l'intension de l'Entité Masquable 'Association A ' considérée et, $STRAT_{INT}(I(A))$, l'ensemble des REM intensionnelles qui définissent la stratification de l'intension de A . Chaque REM intensionnelle $REM_{INT}(I(A))$ de $STRAT_{INT}(I(A))$, est constituée de descriptions de rôles choisis parmi l'ensemble définissant l'intension de A , i.e. $I(A) = \{r_1 : t_1, \dots, r_m : t_m\}$. Ainsi, chaque $REM_{INT}(I(A))$ donne accès à un plus ou moins grand nombre de rôles de l'association. Dans les termes du MAP, cette définition est interprétée comme illustré par la Figure 5.15b.

Enfin, considérons une classe-association CA d'un modèle de données 'MD' décrit par un diagramme de classes UML comportant s rôles r_i et z attributs att_j , avec $1 \leq i \leq s$ et $1 \leq j \leq z$. La classe-association CA est une EM dont l'intension est à stratifier. Nous appelons $I(CA)$ l'intension de l'Entité Masquable 'Classe d'Association CA ' considérée et, $STRAT_{INT}(I(CA))$, l'ensemble des REM intensionnelles qui définissent la stratification de l'intension de CA . Chaque REM intensionnelle $REM_{INT}(I(CA))$ de $STRAT_{INT}(I(CA))$, est constituée de descriptions de rôles et de descriptions d'attributs choisis parmi l'ensemble définissant l'intension de CA , i.e. $I(CA) = \{r_1 : t_1, \dots, r_s : t_s, att_1 : ta_1, \dots, att_z : ta_z\}$. Ainsi, chaque $REM_{INT}(I(CA))$ donne accès à un plus ou moins grand nombre de rôles et d'attributs de la classe-association. Dans les termes du MAP, cette définition est interprétée en combinant les deux cas décrits précédemment et illustrés par la Figure 5.15.

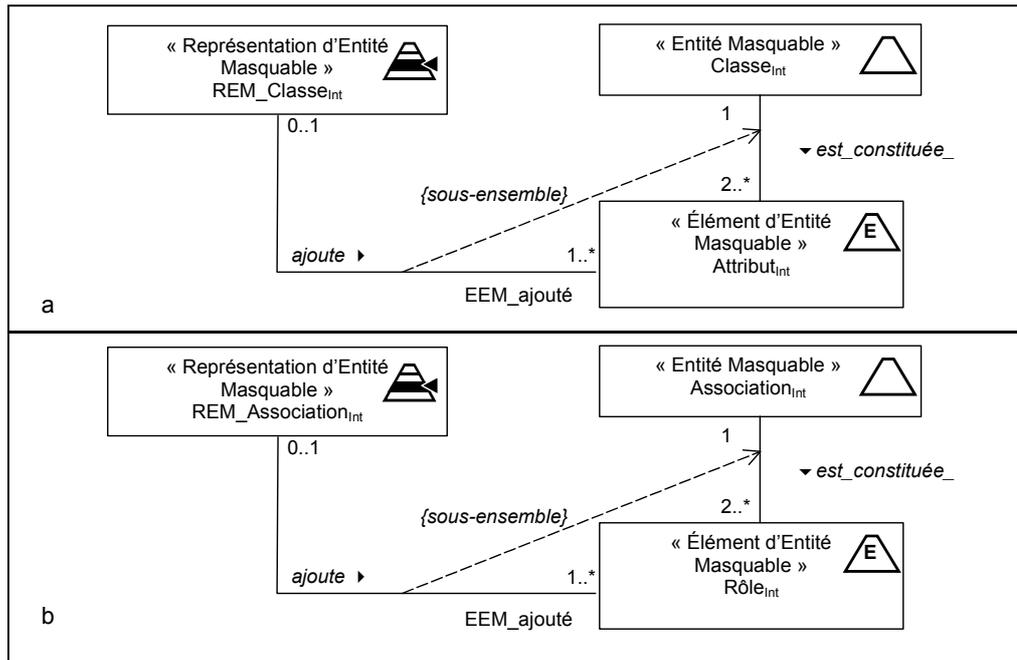


Figure 5.15 – EM potentielles et REM correspondantes pour stratifier en intension une Classe et une Association du Modèle de Données

En haut (a), cas d'une Classe considérée comme une Entité Masquable – En bas (b), cas d'une Association considérée comme une Entité Masquable. Le cas de la Classe-Association, non illustré, est une combinaison des cas de la Classe et de l'Association.

La Figure 5.15 illustre de quel(s) type(s) d'éléments (EEM) sont constituées les EM 'Classe' et 'Association'. La figure décrit également, par voie de conséquence, à quel type d'éléments font référence les Représentations d'Entités Masquables (REM) intensionnelles correspondant à de telles EM.

Dans la section suivante, nous illustrons nos propos en montrant, à partir du modèle de données GDI, des exemples de stratifications en intension pouvant être définies.

5.3.3.2 Exemples de stratification en intension

a – Stratification en intension d'une classe

La Figure 5.16 propose deux stratifications différentes établies pour la Classe Historien du Modèle de données GDI. La classe Historien hérite de sa super-classe Spécialiste les attributs nom, prénom, adresse, organiste et code, et définit l'attribut période. Si aucune stratification en intension n'est donnée pour une classe, alors la stratification en intension est celle de sa plus proche super-classe en remontant les liens de spécialisation/généralisation, si une telle super-classe existe. On ajoute alors les attributs définis par la classe à la REM de plus bas niveau de détail défini pour la super-classe. Si une classe ne possède pas de super-classe et qu'elle n'est pas explicitement stratifiée, alors aucun accès progressif n'est permis pour cette classe.

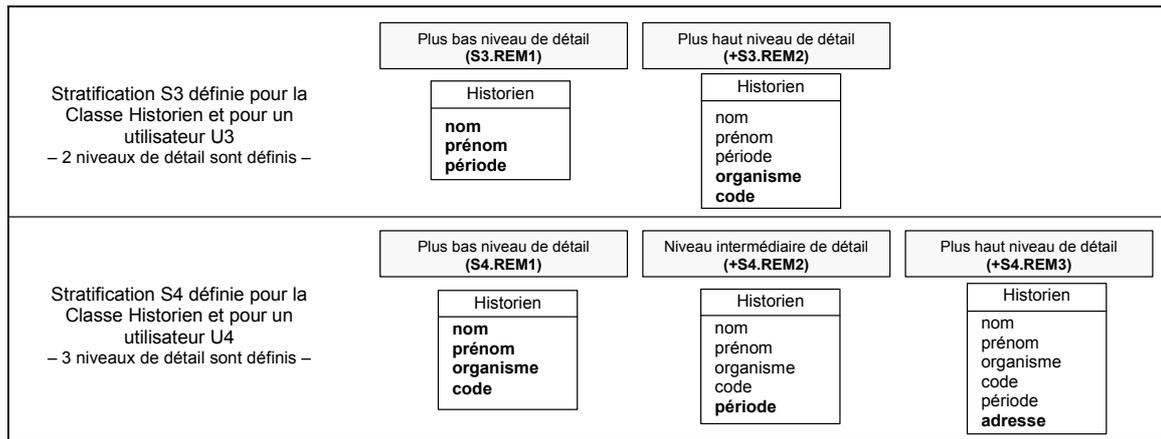


Figure 5.16 – Exemples de stratification d’une Classe considérée comme une Entité Masquable. Les stratifications S3 (en haut) et S4 (en bas) de la Classe Historien, proposées pour deux utilisateurs différents, diffèrent par le nombre de niveaux de détail offerts et par les éléments ajoutés par les REM (en gras).

L’interprétation de la stratification S3 en terme de contenu visible doit être comprise comme illustré par la Figure 5.17. Pour une stratification en intension de la classe, toutes les instances de celle-ci sont visibles, et ce, à tous les niveaux de détail. L’accès progressif s’exprime, lors du passage à un niveau de détail supérieur, par la présentation des mêmes instances mais en montrant les valeurs prises par ces instances pour les attributs ajoutés par la REM correspondante

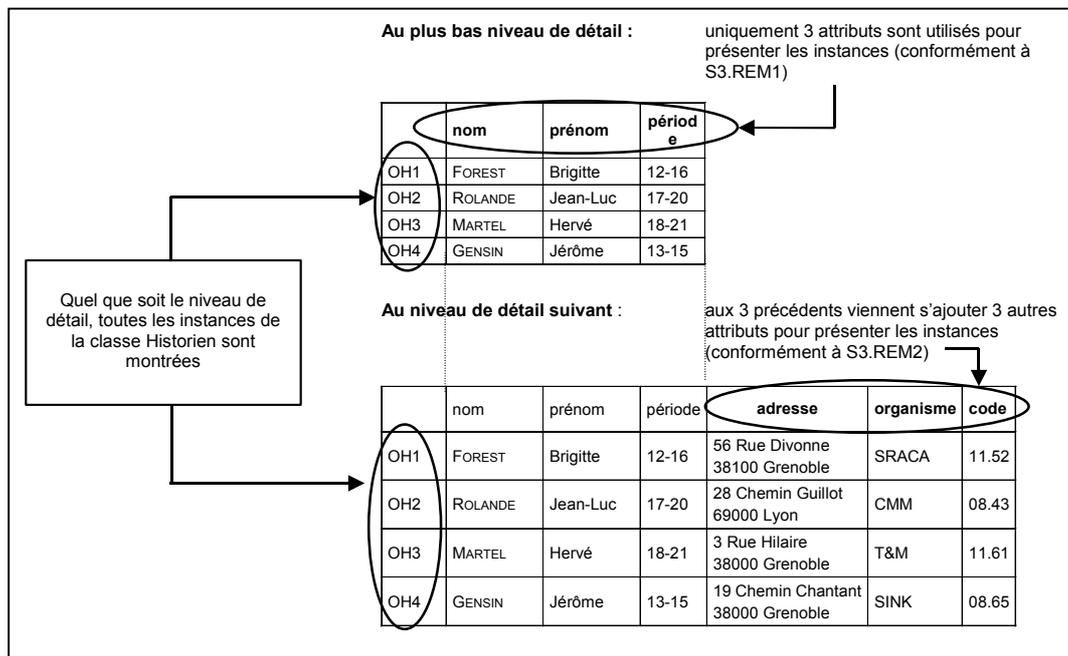


Figure 5.17 – Visibilité des instances d’une classe stratifiée en intension

b – Stratification en intension d’une Association et d’une Classe-Association

La stratification en intension d’une Association consiste à masquer certains rôles à un niveau de détail, et à les ajouter aux autres niveaux. Dans le cas d’une association binaire, il existe

exactement deux niveaux de détail, et donc deux REM définies. Dès que plus de deux classes sont impliquées par l'association considérée comme une EM, alors une stratification reposant sur plus de deux REM est possible.

Une stratification en intension définie pour une Classe-Association combine les possibilités d'une classe et d'une association (i.e. les REM ajoutent rôles et/ou attributs de la classe association).

La visibilité des tuples et instances complexes d'associations et classes associations stratifiées en intension est définie selon des principes similaires à ceux décrits ci-dessus pour une classe.

5.3.4 Stratification en extension d'une EM

5.3.4.1 Description

La stratification en extension d'une EM suit un schéma de description similaire à celui que nous avons présenté pour une stratification en intension : elle répond aux spécifications du MAP mais utilise les définitions en *extension* que nous avons données pour un modèle de données et ses constituants. La Figure 5.18, qui présente les EM et leurs REM respectives pour une stratification en extension, est donc relativement proche de celle illustrant la stratification en intension. On ajoute toutefois ici le cas du Modèle de Données considéré comme une EM potentielle pour une stratification en extension.

Soit $E(MD) = \{E(C_1), \dots, E(C_n), E(A_1), \dots, E(A_m), E(CA_1), \dots, E(CA_p)\}$, l'extension de l'Entité Masquable 'Modèle de données' et $STRAT_{EXT}(E(MD))$, l'ensemble des REM extensionnelles qui définissent la stratification en extension de cette EM. Celles-ci sont construites en choisissant des éléments parmi l'ensemble définissant l'extension de MD (cf. Figure 5.18a). Ainsi, le contenu informationnel du modèle de données est représenté à différents niveaux de détail : chaque REM donne accès à un plus ou moins grand nombre de représentants de la population complète (i.e. instances de classes, tuples d'associations et instances complexes de classes-associations).

La stratification en extension d'une classe C définit un ensemble de REM extensionnelles, $REM_{EXT}(E(C))$, constituées d'éléments choisis parmi l'extension de C, i.e. $E(C) = \{O_1, \dots, O_h\}$. Chaque REM donne accès à un plus ou moins grand nombre d'instances O_i de la classe C, avec $i \in [1, h]$ (cf. Figure 5.18b).

Soit $E(A) = \{TU_1, \dots, TU_q\}$, une l'extension d'une EM Association A que l'on souhaite stratifier. Les $REM_{EXT}(E(A))$ qui composent la stratification en extension de A, $STRAT_{EXT}(E(A))$, sont des ensembles d'éléments choisis parmi celui donnant l'extension de A. Un plus ou moins grand nombre de tuples TU_j de l'association A sont donc accessibles en fonction des REM, avec $j \in [1, q]$ (cf. Figure 5.18c).

Stratifier en extension une EM Classe-Association CA, suppose la définition de REM extensionnelles, $REM_{EXT}(E(CA))$ qui utilisent les éléments de l'extension de CA, $E(CA) = \{IC_1, \dots, IC_r\}$. Ainsi, chaque REM donne accès à un plus ou moins grand nombre d'instances complexes IC_k de la classe association CA, , avec $k \in [1, r]$ (cf. Figure 5.18d).

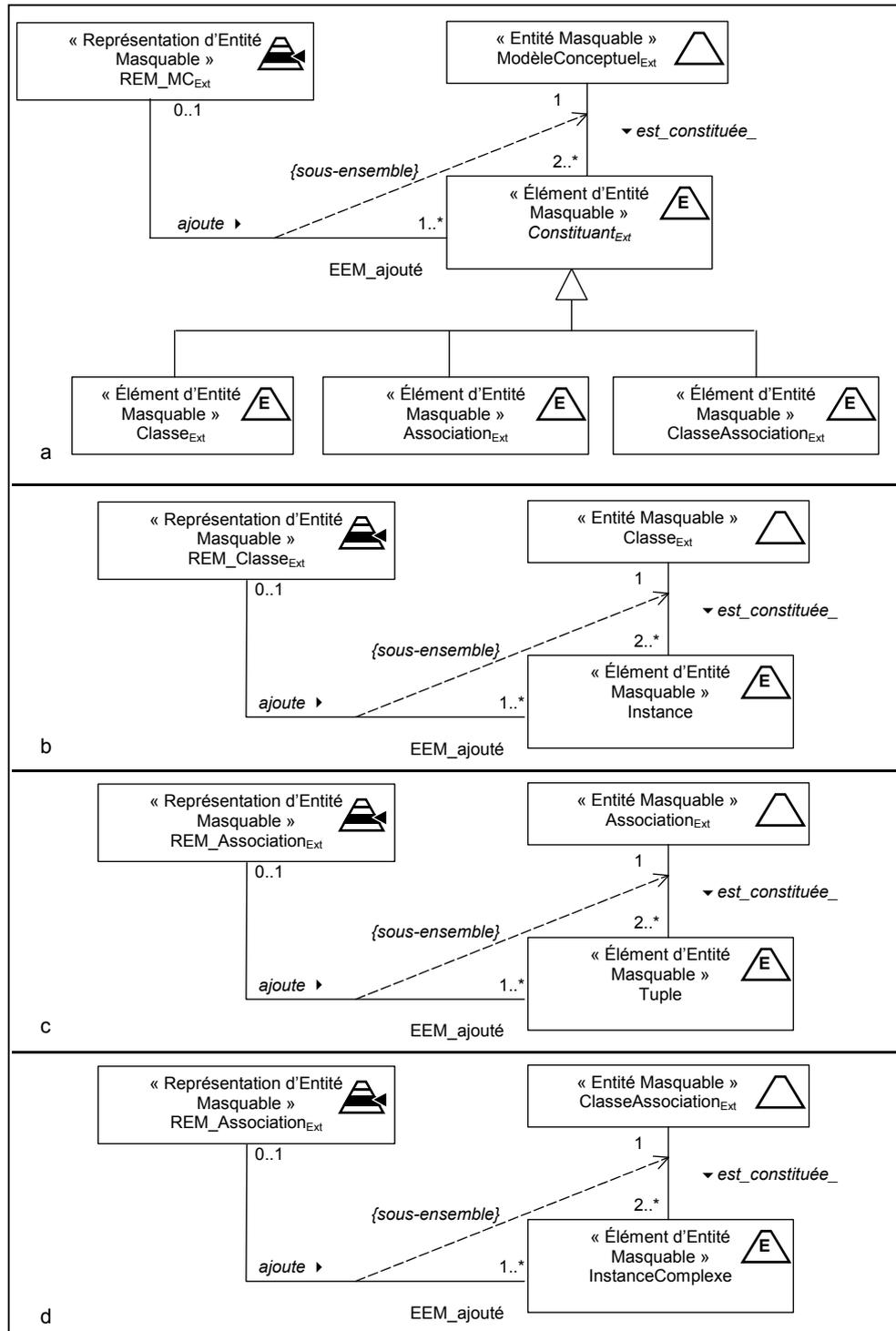


Figure 5.18 – EM potentielles et REM correspondantes pour stratifier en extension un Modèle de données, une Classe, Association et Classe-Association de ce modèle.

De haut en bas : (a) cas du modèle de données considéré comme une Entité Masquable – (b) cas d'une Classe considérée comme une Entité Masquable – (c) cas d'une Association considérée comme une Entité Masquable – (d) cas d'une Classe-Association considérée comme une Entité Masquable.

5.3.4.2 Exemples de stratification en extension

Des exemples de stratification en extension sont donnés dans cette section pour un modèle de données, puis pour les constituants de ce modèle.

a – Stratification en extension d'une EM 'Modèle de données'

L'ensemble des instances et tuples à stratifier correspond à la description en extension d'un modèle de données exprimée dans la section précédente par $E(MD) = \{E(C_1), \dots, E(C_n), E(A_1), \dots, E(A_m), E(CA_1), \dots, E(CA_p)\}$.

La Figure 5.19 illustre graphiquement une stratification en extension du modèle de données GDI. Pour des raisons de lisibilité, nous ne mentionnons que le nom des constituants (indiqué par Ext), et non leur description complète en extension.

La stratification en extension S1 se traduit donc au niveau du contenu par la mise à disposition :

- à un premier niveau, de l'ensemble des instances des classes Spécialiste et Document et des tuples de l'association analyse ;
- à un deuxième niveau, de l'ensemble précédent auquel s'ajoute l'ensemble des instances de la classe association Rapport ;
- à un troisième et dernier niveau, les instances de la classe Inondation et les tuples de l'association porte_sur viennent augmenter l'ensemble précédent.

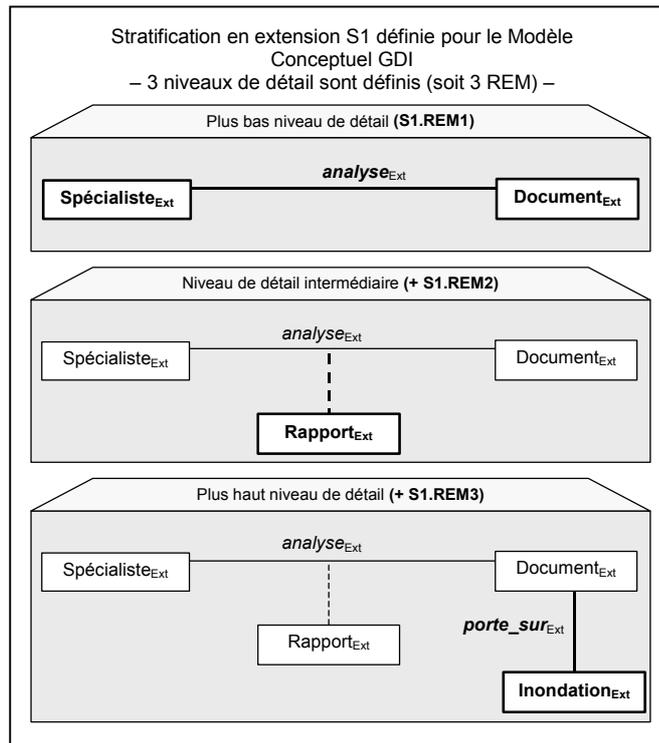


Figure 5.19 – Exemple de stratification en extension d'une EM 'Modèle de données'

La stratification en extension du modèle de données consiste donc à définir des sous-ensembles de constituants du modèle (classes, associations, classes-associations) dont les extensions sont alors accessibles à l'utilisateur à différents niveaux de détail correspondant aux

REM extensionnelles créées. Cependant, les constituants sont eux aussi stratifiés en extension²⁸. Aussi, lors de l'accès aux instances d'une classe (resp. tuples d'association), celui-ci est régi par la stratification en extension de cette classe (resp. association). On accède donc progressivement à une *population stratifiée*, et non pas, comme pour la stratification en intension, à toute la population. Nous abordons cet aspect plus précisément dans la section suivante.

b – Stratification en extension d'une EM 'Classe', 'Association' ou 'Classe-Association'

Considérons de nouveau la classe Historien stratifiée, cette fois, en extension. Les REM(Historien)_{EXT} sont constituées d'éléments choisis parmi l'ensemble définissant l'extension de la classe, i.e. $E(\text{Historien}) = \{OH1, OH2, OH3, OH4\}$.

Nous pouvons, par exemple, stratifier cette population sur la base d'un critère relatif à la compétence des historiens²⁹ : au premier niveau de détail, toute instance telle que la période d'expertise de l'historien s'étend jusqu'au XVI^{ème} siècle inclus est montrée. Au second niveau de détail, les historiens experts pour une période allant au-delà de ce siècle sont ajoutés. La Figure 5.20 illustre une telle stratification. Toutes les instances de la classe sont représentées en utilisant la structure complète de la classe (i.e. les attributs nom, prénom, période, adresse, organisme, code). Deux instances sont visibles au premier niveau de détail (i.e. OH1 et OH4). Au second niveau, les instances OH2 et OH3 complètent l'ensemble.

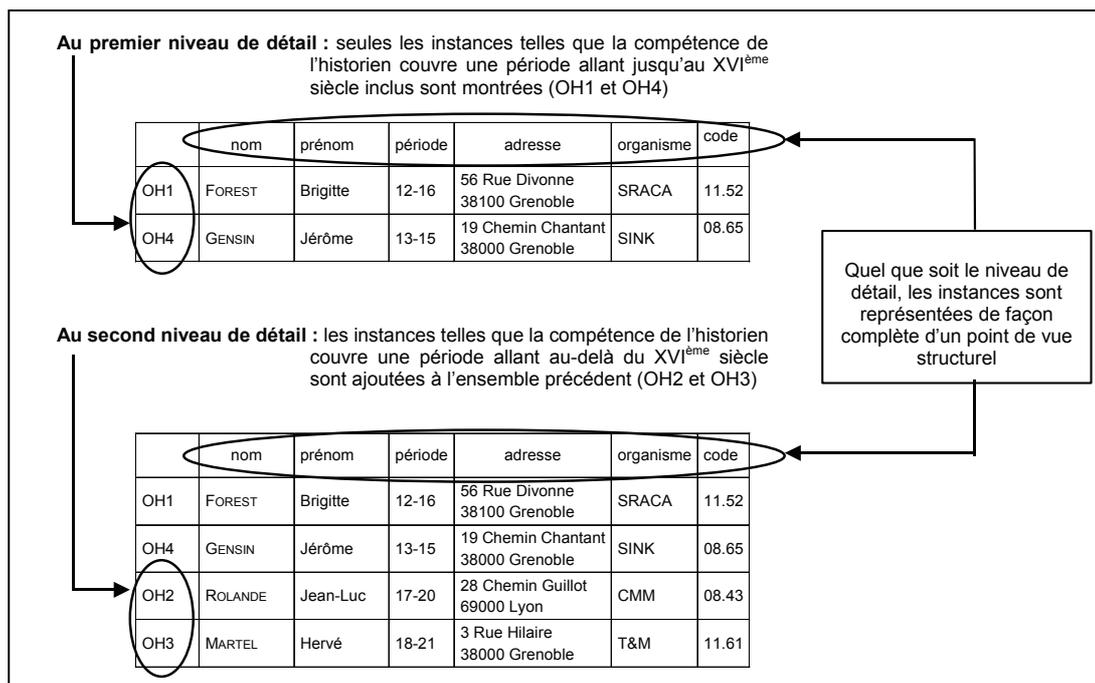


Figure 5.20 – Visibilité des instances pour une stratification en extension d'une classe

La stratification en extension d'une Association ou d'une Classe Association repose sur un principe identique : tous les tuples ou instances complexes sont représentés en utilisant la

²⁸ Nous rappelons que dans le cadre de cette présentation, nous considérons qu'un choix a été fait d'une approche de stratification exclusivement en intension (section précédente), ou en extension (cette section).

²⁹ La façon dont nous exprimons les critères permettant de construire les REM est abordée dans le chapitre 6.

structure complète de leur Association ou Classe Association respective ; les ensembles de population visible comportent plus ou moins de représentants à chaque niveau de détail.

5.3.5 Règles de visibilité du contenu en fonction d'une stratification

Dans les sections précédentes, nous avons décrit comment un modèle de données et/ou ses constituants peuvent être stratifiés en intension ou en extension. Nous avons souligné que l'objectif final de telles stratifications est de permettre un accès progressif au contenu véhiculé par le Modèle de données. Par souci de clarté, nous avons choisi de décrire comment ce contenu est délivré dans des cas relativement simples. Par exemple, pour une stratification en extension d'une classe, les instances ont été présentées en utilisant tous les attributs de la classe.

Néanmoins, il est nécessaire d'exprimer des règles de visibilité plus élaborées. En effet, un modèle de données utilise parfois des constructions (telles que les hiérarchies de classes ou d'associations) pour lesquelles nous devons préciser comment interpréter la sémantique de la stratification. Notamment, la construction de REM peut impliquer de « couper » (virtuellement, puisqu'il n'est ici question que de masquage) de tels liens de hiérarchie. De façon similaire, un tel effet peut être observé pour les liens de relations entre concepts modélisés par des associations ou classes-associations (lorsqu'une association et les classes qu'elle implique n'appartiennent pas à la même REM).

Nous déclinons ci-dessous plusieurs configurations qu'il est possible de rencontrer lors d'une stratification et exposons pour chacune d'elles comment elle est interprétée en termes de visibilité des contenus (i.e. des instances, tuples et instances complexes). Les règles d'interprétation que nous donnons sont, si nécessaire, différenciées selon que la stratification est en intension ou en extension. Lorsque rien n'est précisé, l'interprétation est indiquée quel que soit le type de la stratification.

5.3.5.1 Visibilité des valeurs de rôles d'association

Aucune contrainte ne porte *a priori* sur le choix des constituants d'une REM pour un modèle de données. Une configuration particulière peut survenir. Ainsi, une association (ou une classe-association) peut appartenir à la définition d'une REM extensionnelle dans une stratification de ce modèle alors que les classes reliées par cette association ne sont pas retenues³⁰. Un conflit doit être résolu puisque :

- par définition, les tuples d'une association ont pour valeurs de rôle des instances des classes reliées : si cette association appartient à la REM, alors, à travers les tuples, il est possible d'accéder aux instances des classes liées à cette association. Mais,
- si les classes ne sont pas retenues dans cette REM, cela signifie qu'on ne souhaite pas voir leurs instances à ce niveau de détail.

Nous posons que la visibilité de l'association prime sur celles des classes impliquées : les instances de classes, qui sont valeurs de rôles (et uniquement celles qui le sont), sont accessibles à ce niveau de REM. Cette interprétation repose sur le fait qu'*en tant que valeur de rôle*, une instance de classe appartient à l'extension de l'association. Les instances des classes qui ne sont pas valeurs de rôles ne font pas partie de cette extension et sont donc non accessibles.

Nous illustrons nos propos en nous appuyant sur l'extrait de modèle de données reproduit dans la Figure 5.21. Nous faisons l'hypothèse que la classe association Rapport appartient à une

³⁰ Les classes peuvent apparaître à un niveau supérieur de détail. Néanmoins, tant qu'on n'a pas atteint ce niveau, le cas de figure présenté dans la suite doit être géré.

REM de niveau 1, et que les classes qu'elle implique (Spécialiste et Document) sont masquées à ce même niveau. Les rôles de l'association rapporteur et doc, respectivement de type Spécialiste et Document, permettent de visualiser les instances OS2 et OS3 d'une part, et OD1 et OD2 d'autre part. Ces instances sont en effet des valeurs de rôles dans les instances complexes de Rapport. Dans cet exemple, il se trouve que OD1 et OD2 constituent l'extension complète de la classe Document ; en revanche, les instances OS1 et OS4 de Spécialiste ne sont pas accessibles puisqu'elles ne sont pas valeurs du rôle rapporteur.

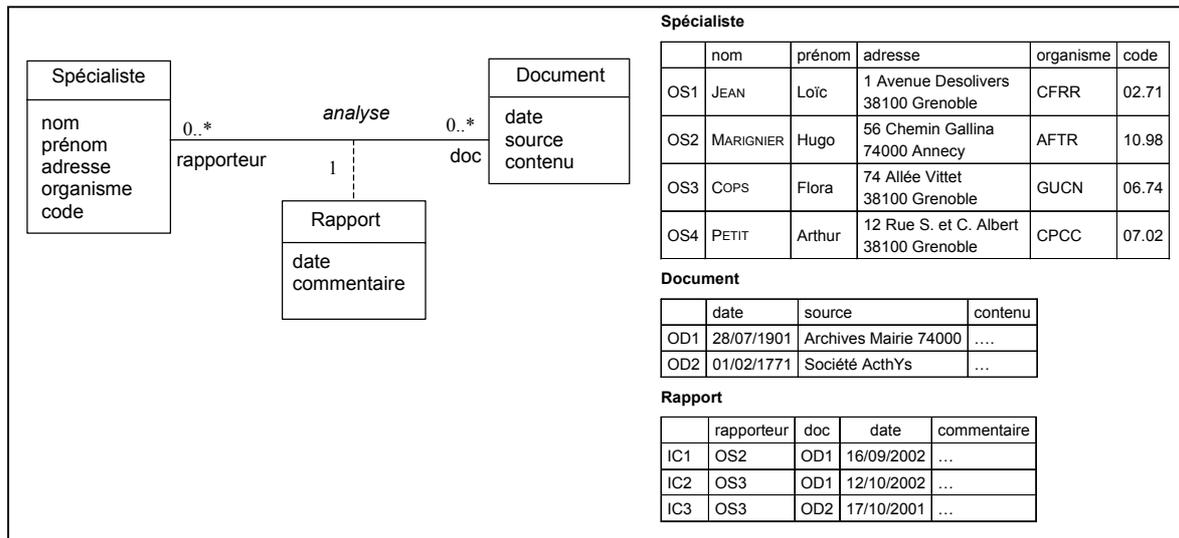


Figure 5.21 – Exemples pour la gestion des associations

Notons qu'un masquage effectif de toutes les instances de classes associées à un rôle est possible. Il suffit pour cela de mettre en œuvre une combinaison de stratifications (cf. section 5.3.2.2). En plus de celle en extension déjà décrite, une stratification en intension de l'association peut être définie pour masquer le rôle en question à un niveau de détail donné. Par exemple, si la classe-association Rapport est stratifiée en intension de telle sorte qu'au premier niveau de détail seul le rôle rapporteur et les attributs date et commentaire appartiennent à la REM, alors le rôle doc est masqué, empêchant ainsi d'accéder aux instances de Document.

5.3.5.2 Gestion des hiérarchies de classes et d'associations

Nous abordons à présent les différentes configurations liées à l'exploitation des hiérarchies de classes et/ou d'associations d'un modèle de données.

Lorsqu'une classe C appartenant à une REM est sous-classe d'une classe C' telle que C n'est pas associée à la même REM, ni à une REM de niveau de détail inférieur, les instances accessibles sont uniquement celles de cette classe C (on ne considère pas les instances de sa sur-classe). De même, pour une association A sous-association d'une autre, dans des conditions de stratification similaire: seuls les tuples de A sont accessibles.

Nous illustrons ce cas à partir des classes et extensions de classes montrées dans la Figure 5.22. Nous supposons que la classe Historien appartient à la REM de niveau 1, et que sa sur-classe Spécialiste appartient à une REM de niveau de détail supérieur. Au niveau de détail donné par REM1, seules les instances d'Historien sont visibles (i.e. OH1 à OH4).

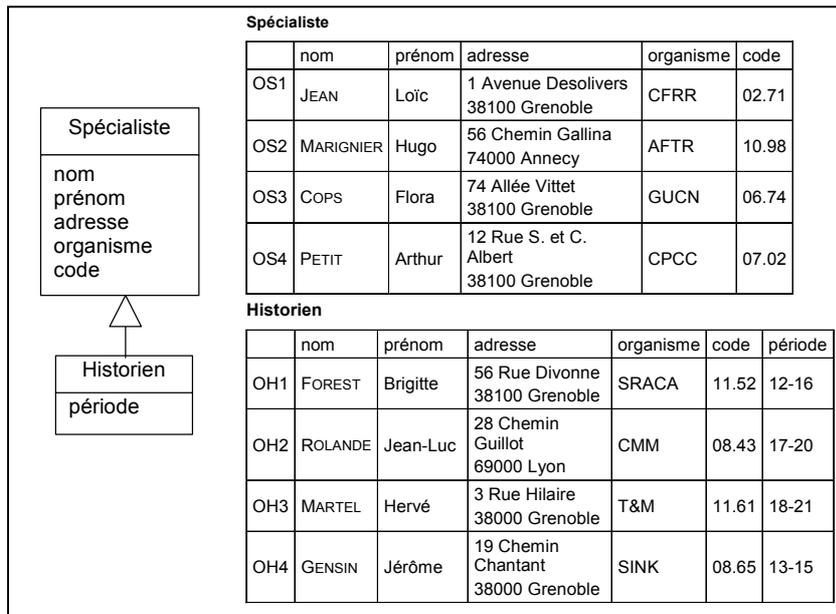


Figure 5.22 – Exemples pour la gestion des hiérarchies de classes

Pour une classe C' appartenant à une REM et sur-classe d'autres classes, les instances accessibles pour C' sont les siennes, d'une part, mais aussi les instances des sous-classes de C' (même si ses sous-classes n'appartiennent pas à la REM en question. Nous considérons en effet ici que l'extension d'une sur-classe comprend les instances de ses sous-classes. Néanmoins, les instances des sous-classes de C' sont, d'un point de vue structurel, considérées comme des instances de C' : seuls les attributs de C' (auxquels s'ajoutent ceux de ses éventuelles sur-classes) sont utilisés pour représenter les instances des sous-classes.

En nous référant à la Figure 5.22, la classe *Spécialiste* considérée en extension permet d'accéder à ses instances OS1 à OS4, mais aussi à celles d'*Historien*. Ces dernières (*i.e.* OH1 à OH4) sont néanmoins présentées avec la structure définie par la classe *Spécialiste* : l'attribut *période* qui est propre à la classe *Historien* n'est pas utilisé. De plus, si la classe *Spécialiste* est stratifiée en intension, les instances de la sous-classe s'y conforment.

5.4. Conclusion sur le Modèle d'Accès Progressif

Nous avons consacré ce chapitre à la présentation du Modèle d'Accès Progressif. Sa vocation est de permettre l'expression, à un niveau conceptuel, d'une organisation des informations favorisant une mise à disposition graduelle de celles-ci.

Les différentes notions liées à l'accès progressif que sont les Entités Masquables (EM), les Représentations d'Entité Masquable (REM) et les Stratifications ont été décrites. Il a été mis en évidence comment une EM, en tant qu'ensemble d'éléments, peut être stratifiée : on définit pour cela différentes REM, des sous-ensembles de l'EM, ordonnés par la relation d'inclusion ensembliste. Deux façons d'envisager la stratification d'une EM ont été étudiées. La stratification en extension agit sur les éléments de l'EM et permet un accès progressif à la population qu'ils constituent. Dans le cas où l'EM correspond à un ensemble de données structurées de même type, une stratification en intension opère sur la structure de l'EM. L'accès progressif s'exprime alors par le fait de donner des représentations de l'EM plus ou moins détaillée, *i.e.* masquant plus ou moins d'attributs de sa structure.

Le Modèle d'Accès Progressif a été décrit par un diagramme de classes reposant sur des notations UML étendues. Un ensemble d'opérations a été proposé pour exploiter les concepts manipulés dans ce modèle. A travers son application à un modèle de données objet, nous avons illustré comment le MAP pouvait être utilisé. L'interprétation des concepts d'un tel modèle de données dans les termes du MAP a été proposée en préambule à la description d'exemples de stratification en intension et en extension.

Le MAP constitue le cadre de référence de la suite de notre proposition. Celle-ci vise la conception de SIW capables de proposer des accès progressifs et personnalisés à leurs utilisateurs. Cet objectif nécessite que les différents modèles participant à la conception du système intègrent les concepts modélisés pour l'accès progressif et collaborent à sa mise en oeuvre. Le chapitre suivant est consacré à la présentation des modèles de conception proposés dans ce but.

6. MODÈLES DE CONCEPTION POUR L'ACCÈS PROGRESSIF

Ce chapitre présente les différents modèles proposés par notre approche pour la conception de SIW. Dans la lignée des méthodes étudiées dans le cadre de l'état de l'art (*cf.* chapitre 3), notre proposition repose sur différents modèles qui représentent le domaine d'application, les fonctionnalités, les utilisateurs, et enfin l'hypermédia qui constitue la face visible du SIW. Les modèles que nous présentons sont, à différents égards, moins complets que ceux proposés par certaines des approches étudiées. La raison essentielle est que nous visons un objectif précis à travers ceux-ci. Nous souhaitons avant tout montrer comment intégrer l'approche par accès progressif dans la conception des SIW. Les modèles proposés s'appuient donc largement sur les spécifications du Modèle d'Accès Progressif (MAP) présentées au chapitre 5. Chacun des modèles contribue à l'objectif final : concevoir et mettre en œuvre un SIW proposant à l'utilisateur un espace personnel dans lequel il évolue de façon progressive. Nous nous concentrons donc sur une définition de modèles que nous qualifions de *guidée par l'accès progressif* : pour chacun, nous envisageons les caractéristiques nécessaires pour qu'il soit un réel support à la mise en œuvre de l'accès progressif.

Le *Modèle du Domaine* représente les objets du monde réel qui seront manipulés dans le SIW. Il met en évidence les concepts et relations entre les concepts du domaine d'application. Nous avons déjà montré, à l'occasion du chapitre 5, comment lui sont appliqués les principes de l'accès progressif. Le Modèle du Domaine est brièvement présenté dans la section 6.1.

Le second modèle de notre approche est appelé *Modèle des Fonctionnalités*. Ce modèle représente et organise au niveau conceptuel les différentes tâches qu'un utilisateur peut effectuer dans le SIW. Guidée par les besoins des utilisateurs, sa définition est par ailleurs conduite avec un objectif de mise en œuvre de l'accès progressif à différents niveaux. Ces niveaux sont portés par des concepts de granularités différentes : l'Espace fonctionnel, le Rôle Fonctionnel et la Fonctionnalité. Le Modèle des Fonctionnalités est décrit dans la section 6.2.

Le *Modèle de l'Hypermédia* est le troisième modèle présenté dans ce chapitre. Les aspects relatifs à la construction de l'hypermédia sont abordés par ce modèle. Nous montrons comment notre approche permet de dériver un hypermédia personnalisé, notamment en s'appuyant sur le

Modèle des Fonctionnalités. Cette dérivation résulte fortement de la sémantique du MAP en ce qui concerne la structuration de l'hypermédia et les possibilités de navigation offertes : un premier sous-modèle de l'hypermédia, appelé Modèle de Structuration & Navigation (des Accès Progressifs) met totalement en œuvre les possibilités d'accès progressif. Le Modèle Hypermédia est complété par un second sous-modèle, dit de Présentation, qui permet de définir comment sont composées les pages de l'hypermédia et quelle est leur apparence. Ce modèle est présenté dans la section 6.3.

Enfin, nous présentons le dernier modèle de notre proposition : le *Modèle des Utilisateurs*. Il distingue les notions de *groupe d'utilisateurs* et d'*utilisateur individuel* en exploitant les spécifications du Modèle des Fonctionnalités. Groupes et utilisateurs sont caractérisés au moyen de profils qui constituent le support d'une adaptabilité portant sur les accès progressifs et la présentation dans le SIW. La section 6.4 est consacrée au Modèle des Utilisateurs.

Nous concluons sur ces différents modèles dans la section 6.5.

6.1. Modèle du Domaine

Les objets du monde réel qui participent à la définition du domaine d'application visé par le SIW sont représentés dans le Modèle du Domaine. Ce modèle est avant tout *conceptuel* dans la mesure où il est destiné à mettre en évidence les concepts du domaine et les relations entre ces concepts. Il est ainsi indépendant de toute considération relative à la dimension hypermédia du futur SIW. Les aspects liés à la structure de l'hypermédia, à la navigation dans l'hypermédia, ou encore à la présentation, seront pris en compte dans le Modèle de l'Hypermédia.

De façon similaire, le Modèle du Domaine est ici décrit indépendamment de toute caractéristique liée à l'accès progressif. Au chapitre précédent, nous avons illustré les possibilités de constructions autorisées par le MAP en vue de permettre l'accès progressif à un modèle de données (section 5.3), ce qu'est, en substance, le Modèle du Domaine. Néanmoins, cette présentation a été faite indépendamment de toute référence à un quelconque besoin de l'utilisateur à l'égard du modèle de données. Ainsi, nous avons stratifié le modèle de données de façon arbitraire en décrivant des sous-ensembles de classes et d'associations, et stratifié de façon similaire des classes ou associations en choisissant tel ensemble d'attributs ou tel rôle pour un niveau de détail donné. Néanmoins, le processus de stratification doit être guidé en vue de proposer des accès progressifs intéressants pour l'utilisateur. Le Modèle de Fonctionnalités, présenté dans la section suivante, établit le lien manquant entre stratifications du Modèle du Domaine et besoins de l'utilisateur.

Nous soulignons qu'une démarche partant de l'analyse et de la conception des besoins des utilisateurs nous paraît essentielle pour construire ce modèle. Nous retenons en particulier l'intérêt que présente une approche basée sur la description de cas d'utilisation. D'une part, ils participent à la production d'un Modèle du Domaine pertinent en regard des besoins utilisateurs et, d'autre part, les cas d'utilisation constituent, dans notre approche, un point de départ pour la spécification des modalités d'accès progressif. Nous abordons cet aspect dans la section suivante, avec la description du Modèle des Fonctionnalités. Nous considérons dans la suite de ce chapitre que le Modèle du Domaine est décrit par un diagramme de classes UML.

6.2. Le Modèle des Fonctionnalités

Dans le chapitre 5, nous avons décrit comment les principes du MAP peuvent être appliqués à un modèle de données. Les stratifications présentées alors visaient essentiellement l'énumération des différentes possibilités autorisées par le MAP pour proposer un accès

progressif au contenu véhiculé par un modèle de données. Pour cela, nous avons adopté une approche décrivant, pour chaque type d'Entité Masquable potentielle, comment sont définies ses stratifications (en intension et/ou en extension). Néanmoins, dans le contexte des SIW, une approche *guidée par les besoins des utilisateurs* est, selon nous, préférable. Deux façons de décliner cette approche nous permettent de pallier deux inconvénients d'une approche guidée par les types d'EM :

1. d'une part, l'approche guidée par les types d'EM n'offre pas de fondement solide pour choisir les stratifications à définir. Diverses difficultés apparaissent telles que les questions liées aux décisions que doit prendre le concepteur (i.e. pourquoi stratifier cette EM plutôt qu'une autre). Ou encore, si ce dernier ne fait aucun choix, se pose alors un problème de coût relatif à la stratification de toutes les EM potentielles, pour tous les utilisateurs potentiels. Nous postulons que se baser sur les besoins des utilisateurs permet de légitimer et d'optimiser le choix des EM à stratifier : les données à considérer comme des EM sont celles impliquées dans la réalisation de fonctionnalités offertes par le SIW pour répondre aux besoins des utilisateurs. Par exemple, imaginons qu'il n'existe aucune fonctionnalité dans le système pour consulter les instances d'une classe donnée, alors, il n'y a pas lieu de stratifier cette classe.
 - La première forme d'approche guidée par les besoins des utilisateurs est celle qui vise à identifier les informations candidates à l'accès progressif. Nous proposons d'exploiter les cas d'utilisation dans cette optique.
2. D'autre part, le fait d'identifier ce qu'il est utile de stratifier n'apporte aucune indication quant à la façon de procéder à la stratification. Sur ce point également, l'approche guidée par les types d'EM est inappropriée car, finalement, très arbitraire.
 - La seconde forme d'approche guidée par les besoins des utilisateurs est celle qui permet de définir la manière de stratifier les informations candidates pour répondre au mieux aux attentes des utilisateurs. Nous proposons une extension aux cas d'utilisation spécifique à l'expression des besoins en matière d'accès progressif. Celle-ci est décrite plus loin dans le chapitre, à l'occasion de la présentation des directives d'une conception exploitant les modèles proposés.

Résultant de cette double approche guidée par les besoins des utilisateurs, le Modèle des Fonctionnalités représente et organise, au niveau conceptuel et dans la logique de l'accès progressif, les fonctionnalités qu'un utilisateur peut mener à bien dans le SIW. Les éléments de modélisation du Modèle des Fonctionnalités sont tout d'abord introduits dans la section 6.2.1. Nous expliquons ensuite, pour les différents concepts présentés, comment ils sont mis en œuvre en référence au MAP.

6.2.1 Les concepts du Modèle des Fonctionnalités

Le Modèle des Fonctionnalités repose sur différents concepts qui permettent d'organiser les activités de chaque utilisateur, et ce, de façon adaptée à ses besoins. Cette organisation repose sur trois niveaux et exploite une approche à base de cas d'utilisation pour la définition des trois concepts décrits à la suite de la Figure 6.1.

Au plus bas niveau, les *Fonctionnalités* sont décrites. Les fonctionnalités sont identifiées à partir des cas d'utilisation dont les acteurs sont des personnes utilisant le système (par opposition aux cas d'utilisation dont les acteurs sont, par exemple, des composants de l'application, d'autres

systèmes, etc.). Des exemples³¹ de fonctionnalités, tirés d'un SIW reposant sur le modèle Gestion Documentaire des Inondations introduit au chapitre 5 (cf. page 96) sont « Consulter la liste des analyses documentaires effectuées par les historiens », « enregistrer un nouvel historien ».

À un niveau intermédiaire, les fonctionnalités sont regroupées en différents Rôles Fonctionnels³². Les fonctionnalités définies pour un même rôle d'acteur permettent d'obtenir un Rôle Fonctionnel. En nous référant au même exemple de SIW, nous pouvons identifier un Rôle Fonctionnel « gestion des spécialistes ». Ce rôle regroupe toutes les fonctionnalités qui incombent à l'acteur chargé de gérer les spécialistes (nominations, affectations à une analyse, etc.). Celle consistant à enregistrer un nouvel historien est l'une de ces fonctionnalités. De plus, un Rôle Fonctionnel « expert en inondation » regroupe pour l'acteur du même nom toutes les fonctionnalités définies pour ce rôle. La fonctionnalité de consultation des analyses documentaires produites par les historiens en fait partie.

Au plus haut niveau, un Espace Fonctionnel, défini pour chaque utilisateur, réunit les différents rôles fonctionnels auxquels cet utilisateur peut prétendre. Un utilisateur donné peut exercer le rôle de plusieurs acteurs. L'Espace Fonctionnel d'un utilisateur donné regroupe l'ensemble des Rôle Fonctionnels qu'il est susceptible de remplir. Supposons l'existence de *Madame Pad*, géographe spécialisée dans les inondations et chargée de la gestion des spécialistes pour le compte du centre documentaire. En tant qu'utilisatrice du système, son espace fonctionnel est constitué des deux rôles fonctionnels précédemment décrits.

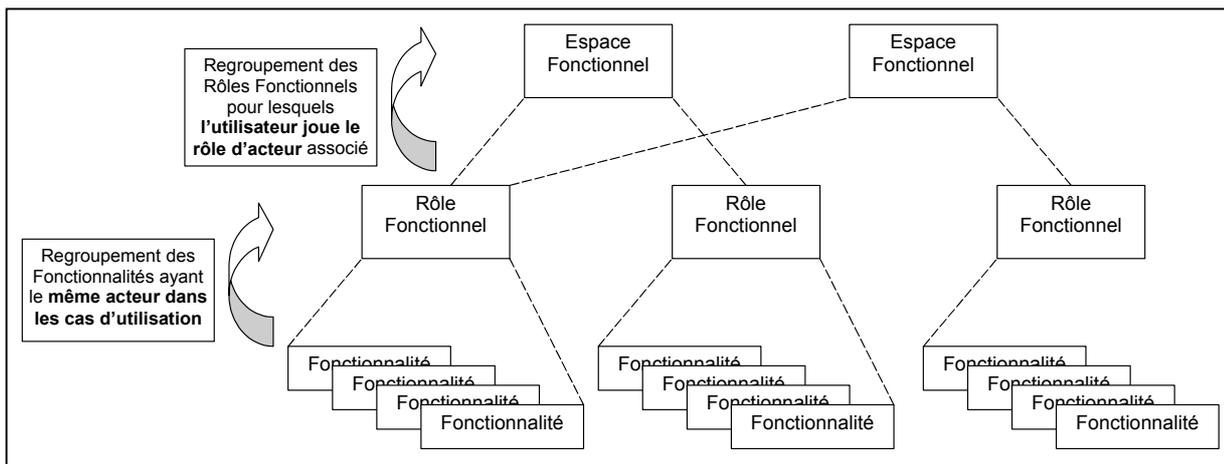


Figure 6.1 – Concepts du Modèle des Fonctionnalités obtenus à partir des cas d'utilisation

6.2.2 Modèle des Fonctionnalités et Accès progressif

Une telle organisation nous permet d'exploiter les principes de l'Accès Progressif défini par le MAP. Ainsi, un Espace Fonctionnel est vu comme une EM constituée de Rôles Fonctionnels. Les Rôles Fonctionnels sont utilisés pour constituer les représentations d'un même espace à des niveaux de détail différents.

³¹ Les exemples que nous donnons dans cette section, bien que fictifs, sont supposés résulter d'une phase d'analyse des besoins.

³² Cette vision peut être rapprochée de celle décrite dans [Gonc00][Gonc01]. Les auteurs établissent une correspondance entre les concepts d'UML et ceux du modèle RBAC (Role Based Access Control) [Ferr01]. Ce que nous appelons *Fonctionnalité* correspond au concept de *Fonction* dans cette proposition, et le concept de *Rôle Fonctionnel* est comparable à celui de *Rôle*.

En nous référant à l'exemple donné précédemment concernant Madame Pad, il peut être défini une stratification de son Espace Fonctionnel telle que :

- Au premier niveau de détail, la REM1 est associée au Rôle Fonctionnel « expert en inondations »,
- Au second, le Rôle Fonctionnel « gestion des spécialistes » est ajouté par la REM2

En descendant d'un niveau de granularité, un Rôle Fonctionnel, constitué de Fonctionnalités, peut à son tour être considéré comme une EM. Ses représentations associées à des niveaux de détail différents donnent accès à plus ou moins de Fonctionnalités.

Ainsi, pour Madame Pad, la stratification du Rôle Fonctionnel « gestion des spécialistes » l'intérêt que présentent, pour son activité quotidienne, les fonctionnalités liées à l'affectation des analyses documentaires aux spécialistes. De telles fonctionnalités (*e.g.* la recherche d'un candidat potentiel pour une analyse décrite page 120, l'affectation de ce spécialiste à cette analyse, l'envoi du message prévenant ce rapporteur nouvellement promu, etc.) sont accessibles au premier niveau de détail. Le reste de la stratification permet de dévoiler progressivement d'autres fonctionnalités, moins couramment utilisées par Madame Pad, telles que l'enregistrement de nouveaux spécialistes, historiens et géographes, modification des données et consultation des informations les concernant.

Dans les deux cas que nous venons de mentionner (*i.e.* stratification d'un Espace Fonctionnel et d'un Rôle Fonctionnel), nous procédons à une application directe du profil UML de stratification décrit au chapitre 5. On note toutefois que la stratification opérée ici est une stratification en extension uniquement : ce sont les instances de Rôle Fonctionnel, et respectivement de Fonctionnalités, qui sont utilisées dans les stratifications.

A un niveau encore plus bas de granularité, une Fonctionnalité peut également être stratifiée selon les principes du MAP, mais à quelques particularités près. Nous consacrons la suite de cette section à présenter l'approche que nous proposons pour concevoir des fonctionnalités de consultation avec accès progressif.

6.2.3 Fonctionnalité de consultation avec accès progressif

Remarque préliminaire – Nous précisons que, dans notre approche, les fonctionnalités auxquelles nous nous intéressons plus particulièrement sont celles pour lesquelles il est pertinent d'associer des mécanismes d'accès progressif. L'apport de l'accès progressif en ce qui concerne un certain nombre de fonctionnalités est, en effet, jugé inexistant. Ces fonctionnalités relèvent par exemple des services communément offerts par les applications tels que la connexion au système, l'envoi d'un e-mail ou encore l'impression d'un document.

L'accès progressif se révèle essentiellement pertinent pour des fonctionnalités qui permettent à l'utilisateur de *consulter* des informations disponibles dans le SIW³³. L'accès progressif intervient dans le fait de stratifier l'ensemble d'informations retourné à l'utilisateur afin de lui éviter d'être submergé par celui-ci.

6.2.3.1 Modélisation de la requête utilisée par une fonctionnalité de consultation

La Figure 6.2 présente la partie du modèle décrivant une fonctionnalité de consultation associée à des mécanismes d'accès progressif. Une telle fonctionnalité de consultation (classe

³³ Toutes les fonctionnalités de consultation ne sont pas forcément associées à des mécanismes d'accès progressif. Par ailleurs, bien que non traitées à ce jour, certaines fonctionnalités impliquant une saisie d'information pourraient également être agrémentées de mécanismes d'accès progressif.

stéréotypée « *Consultation avec Accès Progressif* ») utilise une requête (classe *Requête*) qui permet de sélectionner et de structurer les informations à présenter. La classe *Requête* est modélisée par une composition impliquant plusieurs classes représentant les différentes clauses d'une requête simple³⁴ telle qu'elle est structurée en OQL. Nous représentons :

- la clause *Select* par la classe *Projection* qui définit la structure attendue du résultat de la requête ;
- la clause *From* par la classe *Source* qui introduit les collections (au sens OQL) à partir desquelles est construit le résultat ;
- la clause *Where* par la classe *Sélection* qui spécifie un prédicat permettant de filtrer les collections ;

Une requête produit un résultat modélisé par une classe du même nom. Une relation de dépendance existe entre les classes *Requête* et *Résultat* : toute modification apportée à une requête (plus précisément à l'une de ses classes composantes) a une influence sur le résultat.

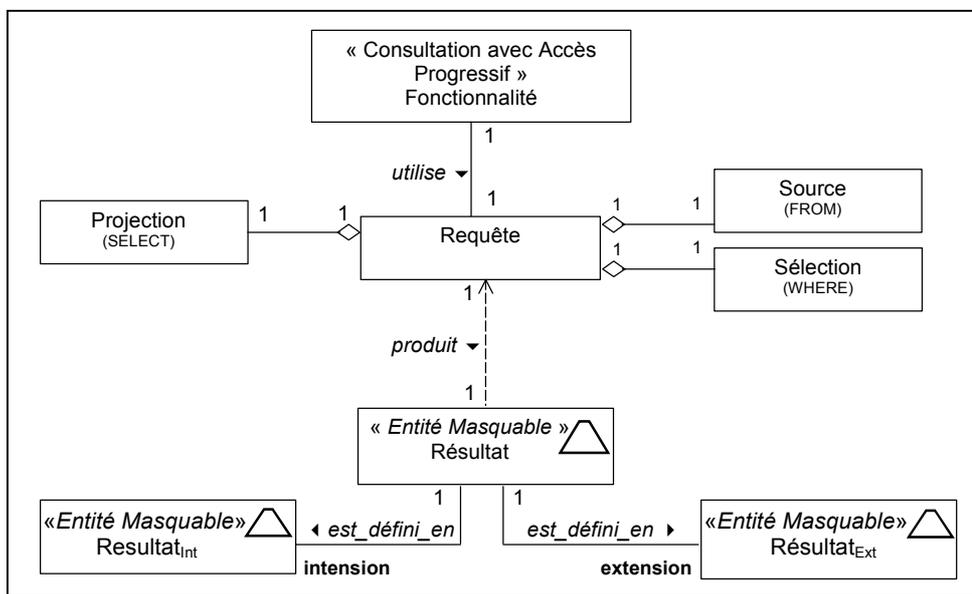


Figure 6.2 – Modélisation d'une Fonctionnalité de Consultation avec Accès Progressif
 Ce type de fonctionnalité utilise une classe *Requête* qui produit un résultat modélisé par une classe *Résultat* vue comme une *Entité Masquable* pour laquelle les définitions en intension et en extension peuvent être stratifiées.

La particularité de ce modèle réside dans le fait de considérer la classe *Résultat* comme une EM. Conformément aux spécifications du MAP, cette classe peut être stratifiée. Plus précisément, ce sont ses définitions en intension et en extension (respectivement *Résultat_{int}* et *Résultat_{ext}*) qui le sont. Nous reportons ainsi sur la classe *Résultat* la capacité de la fonctionnalité à proposer un accès progressif au contenu délivré à l'utilisateur.

³⁴ Nous ne tenons pas compte des clauses optionnelles telles que *having*, *group by*, *order by* dans cette modélisation pour simplifier l'exposé.

6.2.3.2 Stratification du résultat de la requête

La Figure 6.3 donne une représentation plus détaillée de la modélisation précédente. Nous précisons les définitions de l'intension et de l'extension de la classe Résultat et leur appliquons les principes de stratification.

La définition en intension du résultat est donnée directement par la clause SELECT de la requête. En effet, la classe Projection décrit la structure attendue du résultat au moyen d'attributs et de rôles³⁵ issus des classes et associations du diagramme de classes UML décrivant le Modèle du Domaine. L'accès progressif, défini pour un Résultat considéré en intension (classe Résultat_{Int} dans la Figure 6.2), est donc basé sur la stratification de la classe Projection.

La définition en extension du résultat est donné par les extensions (ensembles d'instances et de tuples), indiquées par la source de la requête, auxquelles est appliqué le filtrage défini par la sélection de cette requête. Dans la Figure 6.3, ceci est exprimé au moyen de la classe association Résultat_{Ext} attachée à l'association (s'applique_à) liant les classes Source et Sélection.

Le profil UML de stratification, défini par le MAP (cf. page 85), est ainsi doublement appliqué : à la classe Projection d'une part, et à la classe Résultat_{Ext} d'autre part.

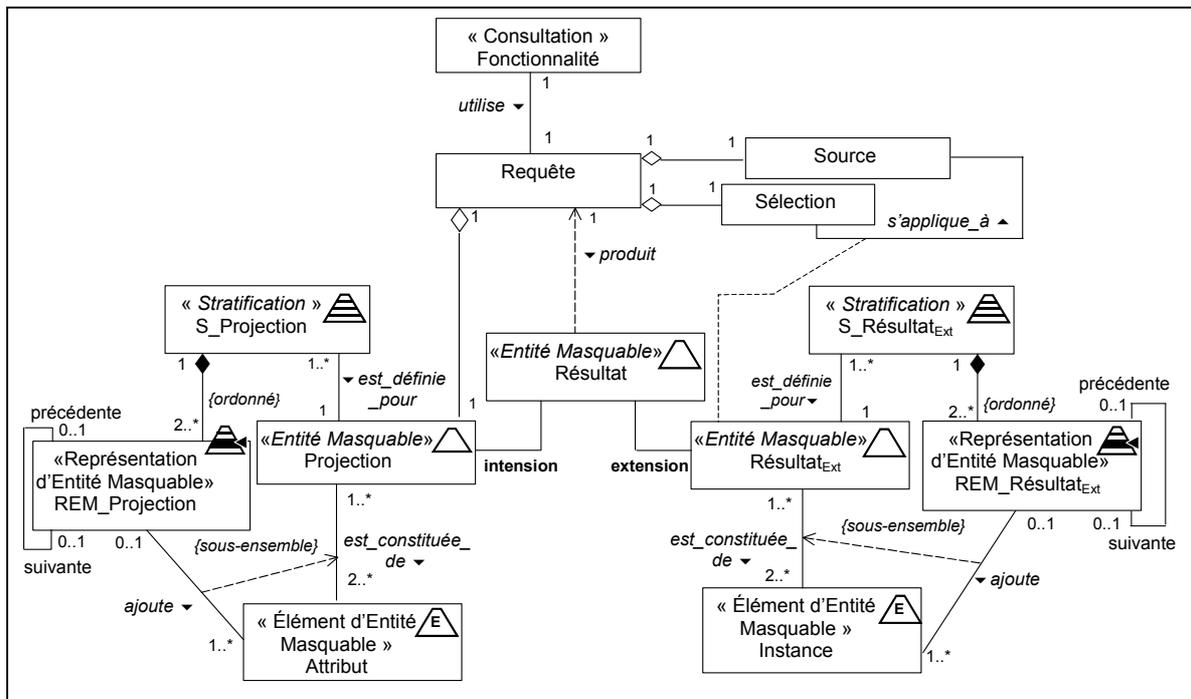


Figure 6.3 – Stratifications en intension et en extension d'une EM Résultat en référence au MAP
Le profil UML de stratification défini par le MAP est appliqué aux deux classes concernées : Projection et Résultat_{Ext}.

Nous proposons donc de modéliser une fonctionnalité en tenant compte de la requête qu'elle utilise et surtout, en considérant le résultat de cette requête comme une entité masquable. L'objectif est d'accroître les possibilités d'expression en vue de l'Accès Progressif au moyen de structures de données qui dépassent le cloisonnement des informations en classes et associations

³⁵ Pour plus de lisibilité, le modèle est simplifié au niveau des classes stéréotypées « Éléments d'Entité Masquable » : une projection est constituée d'attributs mais aussi de rôles. De même l'extension du résultat comporte des instances et des tuples.

(ou en ensembles de ces éléments) tel que nous l'avons présenté au chapitre 5. La modélisation proposée autorise ainsi la construction d'EM plus complexes en bénéficiant de mécanismes inhérents aux requêtes tels que la définition de vues, l'utilisation des jointures, les filtrages sur des valeurs, etc. Ainsi modélisée, une fonctionnalité permet de répondre à des besoins de consultation d'informations plus élaborés, i) en autorisant des clauses de projection et sélection complexes d'informations et ii) en donnant les moyens de stratifier de différentes façons le résultat d'une même requête pour garantir un Accès Progressif personnalisé au contenu qu'elle délivre.

Dans la section suivante, nous présentons trois cas de stratification d'un résultat afin d'illustrer quelques possibilités d'accès progressif à une information délivrée par une fonctionnalité de consultation.

6.2.3.3 Stratifications types : définitions et exemples

Nous mettons en exergue des stratifications non encore abordées jusqu'ici. Le premier cas décrit comment sont exprimés les critères pour une stratification en extension d'un résultat. Le second traite d'une stratification en intension d'un résultat à partir d'une clause de projection complexe ('select' imbriqué). Enfin le troisième cas illustre la gestion de stratifications « croisées » qui combinent stratification en intension *et* en extension d'un même résultat.

a – Expression de critères pour la stratification en extension d'un Résultat

Stratifier un Résultat en extension nécessite la définition de REM pour cette EM Résultat. Cette définition repose sur des requêtes spécifiques à l'Accès Progressif appliquées à la définition en extension de ce Résultat, *i.e.* à Résultat_{Ext}. Ces *requêtes de stratification* visent à définir le contenu (*i.e.* les instances *et/ou* tuples) ajouté à chaque niveau de détail.

La stratification en extension du résultat est donnée par :

```

DEFINE une_Stratification as (
  ListeEEMs : (SELECT *
              FROM R in RésultatExt)
  la_première_REM : (STRUCT
                    ( REM_Prec : [null, autre_REM]
                      EEMajoutés : ( SELECT *
                                    FROM e in ListeEEMs
                                    WHERE expression )
                    ...
  la_dernière_REM : (STRUCT
                    ( REM_Prec : [null, autre_REM]
                      EEMajoutés : ( SELECT *
                                    FROM e in ListeEEMs
                                    WHERE expression ))

```

où l'expression dans les clauses WHERE de définition des REM définit le filtrage à appliquer à la collection ListeEEMs pour retenir les éléments à ajouter (EEMajoutés) à chaque niveau. Le filtrage ne peut porter que sur des valeurs d'attributs *et/ou* de rôles appartenant à la structure des membres de la collection ListeEEMs.

De plus, dans la syntaxe de définition d'une stratification donnée précédemment, la clause WHERE dans chaque définition de REM met en œuvre une propriété particulière. Afin de garantir

que les éléments ajoutés dans une_REM ne soient pas dans les ensembles d'éléments ajoutés par les REM précédentes, de façon transparente, la clause WHERE indique en plus de l'expression :

```

une_REM : (STRUCT
  ( REM_Prec : [null, autre_REM]
    EEMajoutés : ( SELECT *
      FROM e in ListeEEMs
      WHERE expression and
      e not in REM_Prec1.EEMajoutés and
      e not in REM_Prec2.EEMajoutés, ...)
    )
  )

```

avec REM_Prec1 désigne la REM immédiatement précédente de une_REM, REM_Prec2 désigne la REM précédant REM_Prec1, etc.

Une telle requête permet que le filtrage appliqué pour définir une REM soit sans rapport avec celui opéré pour une autre REM (ceci relève en fait d'un choix de conception). Lors de la présentation d'une REM, l'inclusion ensembliste est garantie par le fait de présenter un ensemble constitué de l'union de cette Rem et de celles qui la REM précèdent dans la stratification. Le scénario suivant, qui fait référence au SIW basé sur le modèle de Gestion Documentaire des Inondations (*cf.* page 96), nous permet d'illustrer un tel cas.

Supposons qu'une fonctionnalité permette de consulter les instances d'Historien n'ayant encore produit aucun rapport d'analyse documentaire. Cette fonctionnalité répond au besoin de l'acteur chargé de trouver un Historien pour effectuer un rapport, avec la consigne de demander ce travail à ceux qui n'ont pas été très productifs jusqu'ici. L'historien OH1 étant déjà l'auteur d'un rapport, seules OH2, OH3 et OH4 sont les instances candidates qui constituent ainsi l'extension du résultat de la requête Résultat_{Ext}.

Ce rapport devant être produit rapidement et le document à analyser étant à Grenoble, il est souhaitable de trouver en priorité un Historien habitant cette ville. Néanmoins, le document portant sur une inondation du XIX^{ème} siècle, on accepte, mais avec un degré de priorité moindre, de tenir compte de la période de compétence de l'Historien.

On définit pour cela une REM montrant au premier niveau les Historiens habitant Grenoble. Une seconde REM ajoute au niveau de détail suivant ceux qui dont la période de compétence comprend le XIX^{ème} siècle. La stratification de Résultat_{Ext} serait donc ici telle que :

- REM1 donne accès à OH3 et à OH4 qui habitent Grenoble, et
- REM2 ajoute OH2 dont la période couvre du XVII^{ème} au XX^{ème} siècle. On note que OH3, expert du XVIII^{ème} au XXI^{ème} siècle, appartenant déjà à REM1 n'est pas ajouté bien qu'il réponde positivement au filtre spécifié pour REM2.

Soit, en s'appuyant sur la description donnée précédemment et en supposant :

- que l'adresse d'un historien est décomposée en rue, code postal, ville et
- que la période de compétence d'un historien est donné par un siècle de début et un siècle de fin,

nous définissons :

```

DEFINE ma_Stratification as (
  ListeEEMs : (SELECT *
              FROM H in Historien)
  la_première_REM : (STRUCT
                    ( REM_Prec : null
                      EEMajoutés : ( SELECT *
                                    FROM e in ListeEEMs
                                    WHERE adresse.ville="Grenoble"))
                    )
  la_dernière_REM : (STRUCT
                    ( REM_Prec : la_première_REM
                      EEMajoutés : ( SELECT *
                                    FROM e in ListeEEMs
                                    WHERE e.période.début ≤ 19 and
                                          e.période.fin ≥ 19 and
                                          e not in la_première_REM.EEMajoutés))
                    )
)

```

b – Stratification d'un Résultat pour une Projection présentant une imbrication

Le second cas que nous présentons porte sur la spécification d'un accès progressif au résultat considéré en intension. La clause SELECT fait donc l'objet de la stratification dans cet exemple.

Remarque : La syntaxe donnée précédemment pour une requête de stratification en extension d'un Résultat (Résultat_{Ext}) peut être appliquée ici en remplaçant Résultat_{Ext} par Résultat_{Int}. Nous rappelons que Résultat_{Int} est donné par classe Projection. En d'autres termes, cette requête de stratification en intension doit être comprise comme étant formulée sur le méta-modèle : la liste des éléments à partir desquels est opérée la stratification est composée des attributs qui décrivent la structure du Résultat.

```

DEFINE une_Stratification as (
  ListeEEMs : (SELECT *
              FROM R in RésultatInt) -- i.e.: les attributs décrivant la structure du Résultat
  ...
  une_REM : (STRUCT
            ( REM_Prec : [null, autre_REM]
              EEMajoutés : ( SELECT *
                            FROM e in ListeEEMs
                            WHERE expression )
            )
  ...
)

```

L'Annexe 2 dresse un inventaire des différentes stratifications envisageables en fonction du type de projection utilisé par la requête (*i.e.* projection totale ou restreinte sur une source unique ou multiple). Nous traitons ici uniquement le cas d'un type de projection particulier qui repose sur une imbrication de clauses SELECT.

Nous considérons pour cela la requête donnée ci-dessous. Elle permet d'obtenir les nom, prénom et organisme des Historiens, et pour chacun, les nom, prénom et adresse de ses subordonnés (en supposant que, dans le Modèle GDI, un subordonné est un Spécialiste lié par une association à un autre Spécialiste exprimant cette sémantique). Nous omettons la clause de sélection, mais celle-ci peut parfaitement filtrer les collections (par exemple pour ne retenir que les subordonnés travaillant pour le même organisme que leur responsable).

```

SELECT STRUCT (
  nom_historien : H.nom,
  son_prénom : H.prénom,
  son_organisme : H.organisme
  ses_subordonnés :
    ( SELECT STRUCT (
      sub_nom : S.nom,
      sub_prénom : S.prénom,
      sub_adresse : S.adresse)
      FROM S in H.subordonné))
FROM H in Historien
    
```

Un tel résultat considéré en intension admet une stratification qui, tout comme la projection, a la particularité d'être imbriquée. Ainsi, la première stratification, appelée *principale*, est déterminée à partir d'un ensemble d'éléments constitué des attributs: `nom_historien`, `son_prénom`, `son_organisme`, et `ses_subordonnés`. La propriété d'imbrication intervient dans la mesure où l'attribut `ses_subordonnés` peut, à son tour, être stratifié par définition de REM utilisant les attributs `sub_nom`, `sub_prénom`, `sub_adresse`. On parle de stratification *imbriquée*.

La Figure 6.4 illustre les possibilités d'accès progressif pour une stratification portant sur l'exemple précédent de projection avec imbrication. On note que les mécanismes de masquage/dévoilement sont spécifiques à chaque stratification (*i.e.* principale et imbriquée).

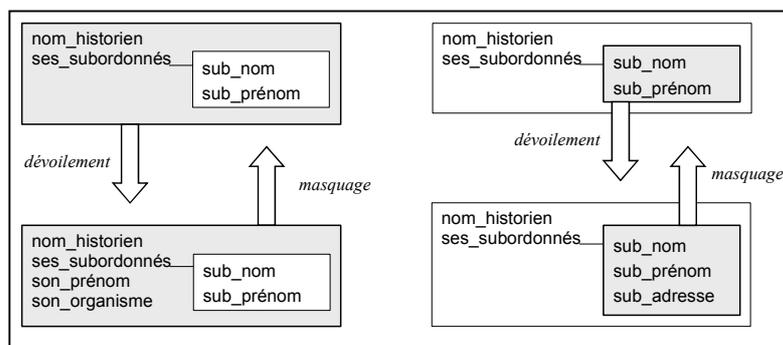


Figure 6.4 – Stratification d'une projection avec imbrication et mécanismes de masquage/dévoilement
 À gauche, les mécanismes de masquage/dévoilement appliqués à la stratification principale ; à droite, appliqués à la stratification imbriquée. À partir de la représentation en bas à gauche, ces mécanismes peuvent être appliqués à la stratification imbriquée ; de même, à partir de la représentation en bas à droite, ils peuvent être appliqués à la stratification principale.

c – Gestion des stratifications croisées

Nous désignons par *stratifications croisées* le cas où un même résultat de requête est stratifié à la fois en intension et en extension. Notre modèle permet d'exprimer ce type de situation, les deux types de stratification étant indépendants. Bien qu'elles portent sur la même EM, chaque

stratification traite en effet un 'aspect' différent : sa définition en intension ou celle en extension. Nous avons montré au chapitre 5 comment de telles stratifications peuvent être combinées (*cf.* page 100). En guise d'illustration, nous appliquons ici une stratification en extension sur une stratification en intension.

Ainsi, nous considérons le résultat stratifié en intension présenté pour les stratifications imbriquées. Nous appliquons à celui-ci une stratification en extension telle que :

- au premier niveau de complétude, seules les instances d'historiens habitant Grenoble et ayant plus de deux subordonnés sont montrés ;
- au second niveau seraient ajoutées les instances d'historiens grenoblois ayant deux subordonnés ou moins,
- au dernier niveau, tous les autres seraient visibles.

Conformément à la sémantique de la combinaison, à chacun de ces trois niveaux, les instances d'historiens et les informations sur leurs subordonnés sont représentées par une structure stratifiée. Il est donc possible d'accéder progressivement, et de façon parallèle, à plus d'individus de la population *et* à plus d'attributs descriptifs les concernant.

Parfaitement exprimables au niveau conceptuel, nous observons toutefois que les stratifications croisées augmentent le nombre de modes d'accès progressif à l'information. Ceci peut aller à l'encontre du but recherché et perturber l'utilisateur face à des possibilités multiples de masquage/dévoilement de l'information. Aussi jugeons-nous nécessaire de mener parallèlement une réflexion sur la façon de présenter les informations stratifiées. Nous revenons sur cet aspect dans le Modèle de l'Hypermédia (*cf.* section 6.3).

6.2.4 Conclusion sur le Modèle des Fonctionnalités

Le Modèle des Fonctionnalités que nous proposons repose sur une organisation des activités en trois niveaux : Fonctionnalités, Rôles Fonctionnels et Espaces Fonctionnels. Ces trois niveaux sont obtenus à partir d'une exploitation simple des cas d'utilisation basée sur leur regroupement.

Chacun de ces niveaux est, par ailleurs, associé à des propriétés d'adaptation par mise en œuvre de l'accès progressif résultant de l'exploitation du MAP. Nous avons plus particulièrement décrit la partie du modèle dédiée à la façon dont nous représentons une Fonctionnalité dans ce but. La modélisation de la requête utilisée par cette fonctionnalité, ainsi que du résultat produit par cette requête permettent d'envisager de multiples possibilités de stratifications, tant en intension qu'en extension.

Dans la section suivante, nous abordons la partie de notre proposition permettant de définir l'hypermédia mis à la disposition de l'utilisateur. Le Modèle de l'Hypermédia est en grande partie dérivé des spécifications que nous venons de décrire dans le cadre du Modèle des Fonctionnalités.

6.3. Le Modèle de l'Hypermédia

Le Modèle de l'Hypermédia est décomposé en deux sous-modèles que nous présentons successivement. Le premier, appelé *Modèle de Structuration et de Navigation*, décrit la dimension structurelle de l'hypermédia et les possibilités de navigation dans celui-ci qui sont inhérentes à l'accès progressif. Il s'agit d'un modèle de niveau logique qui repose très largement sur les principes du Modèle d'Accès Progressif mis en œuvre par le Modèle des Fonctionnalités. Nous présentons ce modèle dans la section 6.3.1.

Le second sous-modèle est appelé *Modèle de Présentation*. Il intègre les spécifications du Modèle de Structuration & Navigation des Accès Progressifs et leur donne une interprétation en termes de composition et d'apparence des pages Web de l'Hypermédia. Ces deux aspects sont traités séparément grâce aux notions de *charte de composition* et de *charte graphique*. La première utilise les éléments de modélisation du Modèle de Structuration & Navigation, la seconde s'applique à ces mêmes éléments. Nous décrivons le Modèle de Présentation dans la section 6.3.2.

6.3.1 Le Modèle de Structuration & Navigation des Accès Progressifs

Le Modèle de Structuration & Navigation (désigné par Modèle S&N) donne la représentation hypermédia du Modèle des Fonctionnalités en exploitant les caractéristiques d'accès progressif de celui-ci. Le Modèle S&N est basé sur des concepts qui font référence au MAP, et que nous relient aux concepts du Modèle des Fonctionnalités qui les mettent en oeuvre. Par exemple, un élément de modélisation du Modèle S&N correspond à l'EM définie par le MAP, et est lié aux concepts du Modèle Fonctionnel considérés comme des EM (e.g. un Espace Fonctionnel). On distingue les concepts de structuration et de navigation.

6.3.1.1 Les concepts pour la structuration

Les concepts pour modéliser la structure hypermédia sont les *contextes* et les *nœuds*. Les contextes sont attachés au concept d'Entité Masquable du MAP. Pour le Modèle des Fonctionnalités, il existe en conséquence trois types de contexte :

- le *contexte d'Espace Fonctionnel* et le *contexte de Rôle Fonctionnel*, qui correspondent respectivement aux classes du même nom considérées comme des EM dans le Modèle des Fonctionnalités, et,
- le *contexte de Fonctionnalité* par lequel nous désignons le contexte attaché à la classe Résultat, *i.e.* l'EM sur laquelle est mise en œuvre l'Accès Progressif à travers une Fonctionnalité dans le Modèle des Fonctionnalités.

Chaque contexte regroupe des nœuds, et plus précisément :

- un *Nœud Stratification* qui fait référence à la classe « Stratification » du MAP. Un tel nœud présente, au niveau de l'hypermédia, l'information relative à la stratification d'une EM.
- plusieurs *Nœuds REM* en référence à la classe du même nom dans le MAP. Il existe autant de Nœuds REM pour un Nœud Stratification qu'il existe d'instances de REM pour l'instance de Stratification. Ces nœuds comportent un (ou plusieurs) élément(s) de structuration interne appelé(s) *Nœud(s) EEM*.
- les *Nœuds EEM* correspondent à la classe EEM du MAP. Il existe autant de Nœuds EEM dans un Nœud REM que d'instances d'EEM ajoutées par l'instance de REM associée au nœud REM.

En fonction du contexte considéré parmi ceux identifiés pour le Modèle des Fonctionnalités, ces nœuds véhiculent différents contenus. Ceci est illustré par la Figure 6.5. Le modèle générique décrit en référence au MAP (à gauche) est appliqué à trois reprises, c'est-à-dire à chacun des trois contextes identifiés dans le Modèle des Fonctionnalités (à droite).

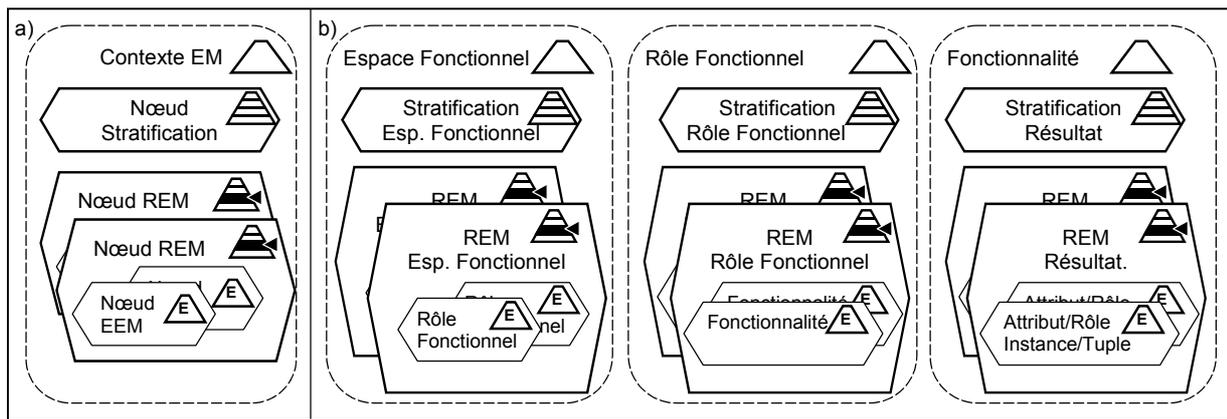


Figure 6.5 – Les concepts de structuration du Modèle de Structuration & Navigation
 a) les concepts génériques décrits en référence au MAP et leur représentation graphique – b) application des concepts aux trois contextes spécifiques du Modèle des Fonctionnalités.

En plus des contextes et des nœuds, le Modèle S&N est agrémenté de concepts pour la navigation que nous décrivons dans la section suivante.

6.3.1.2 Les concepts pour la navigation

Les liens existant entre les différents concepts de structuration (*i.e.* contextes et nœuds) font également référence aux spécifications du MAP. Les liens traduisent en effet certaines relations existant entre concepts du MAP. Nous décrivons ces liens et montrons comment ils sont utilisés dans la structure hypermédia dérivée du Modèle des Fonctionnalités présentée dans la section précédente. On distingue les liens de navigation *internes à un contexte* et ceux permettant un *changement de contexte*.

a – Liens de navigation interne à un contexte

Les liens internes à un contexte sont les liens de *masquage / dévoilement* portés par les Nœuds Stratification. Ils sont respectivement associés aux opérations de masquage et de dévoilement de la classe « Stratification ». L'activation de ces liens engendre une navigation en destination d'un Nœud REM appartenant au même contexte. Étant donné le Nœud REM courant, associé la REM_i , suivre le lien de dévoilement permet d'atteindre le Nœud REM associé à la REM_{i+1} . Le lien de *masquage* a l'effet l'inverse, conformément aux spécifications du MAP : le nœud atteint est celui associé à la REM_{i-1} . Lorsque le Nœud REM courant est celui associé à la REM de plus bas niveau de complétude, le lien de masquage n'est pas activable dans le Nœud Stratification. S'il est associé à la REM de plus haut niveau de complétude, le lien de dévoilement est absent.

L'encart en haut à droite de la Figure 6.6 représente la navigation d'un Nœud REM vers le Nœud REM précédent (ou suivant) par activation des liens de masquage (M) et dévoilement (D) dont est doté le Nœud Stratification. L'exemple montre comment plus ou moins de Rôles Fonctionnels (notés 1er Rôle F., 2e Rôle F., 3e Rôle F.) sont visibles en fonction de l'activation d'un dévoilement ou d'un masquage depuis le Nœud Stratification de l'Esp. Fonct.. Les Nœuds Stratification et REM des deux autres contextes (*i.e.* Rôle Fonctionnel et Fonctionnalité) mettent en œuvre les mêmes mécanismes.

b – Liens de navigation entre contextes

Les liens de navigation permettant de passer d'un contexte à un autre sont soumis à une relation existant entre les Nœuds EEM du contexte source d'une part, et le contexte destination

d'autre part. Ainsi, l'EEM attaché au Nœud EEM et l'EM attaché au contexte destination doivent être définis sur le même concept. En d'autres termes, il doit exister un concept C tel qu'il est considéré comme un EEM dans une stratification *et* considéré comme l'EM d'une autre stratification. C'est notamment le cas dans le Modèle des Fonctionnalités en ce qui concerne le concept de Rôle Fonctionnel et celui de Fonctionnalité (en assimilant une Fonctionnalité au Résultat de la Requête, ce Résultat étant le concept réellement considéré comme l'EM).

Les liens de passage d'un contexte à l'autre apparaissent dans la Figure 6.6. Les liens de navigation entre contextes exploitent les trois niveaux de description du Modèle des Fonctionnalités entre lesquels la condition relationnelle décrite précédemment est respectée. Ainsi, dans le contexte de L'Espace Fonctionnel, le Nœud EEM « 2ème Rôle Fonctionnel » permet d'atteindre le contexte du même nom. Depuis ce contexte, à partir du Nœud « 1ère Fonctionnalité », il est possible d'atteindre le contexte correspondant. Il en résulte un hypermédia à la structure arborescente telle que cette structure peut être étendue :

- en largeur grâce au dévoilement des REM permettant d'accéder à un niveau de détail plus élevé, et
- en profondeur grâce au passage à un contexte de niveau de granularité inférieur (qui peut à son tour éventuellement être étendu en largeur).

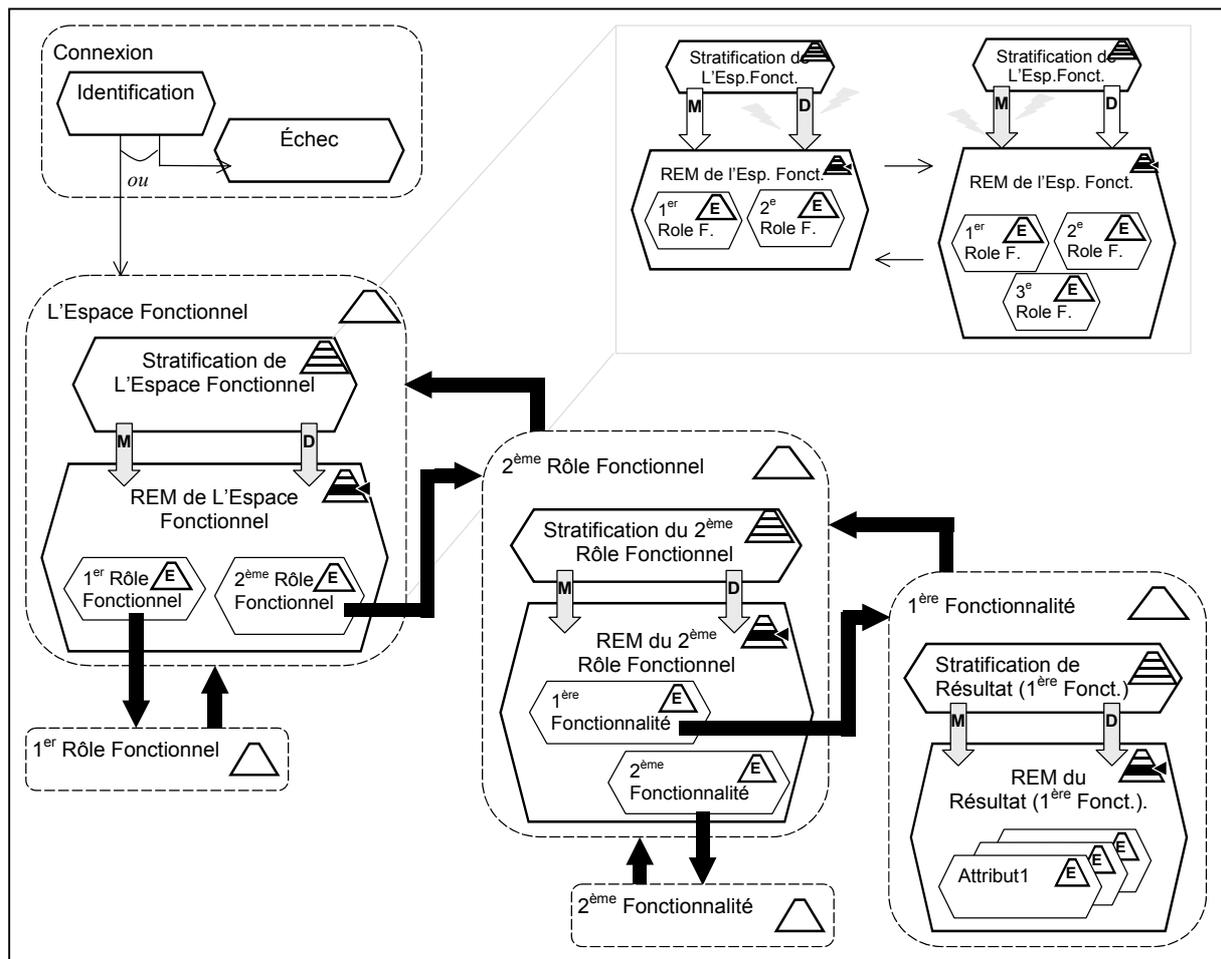


Figure 6.6 – Représentation du Modèle de Structuration & Navigation
 En gris, portant le caractère M ou D, les flèches représentent les liens de navigation interne au contexte. Les effets des liens de Masquage et Dévoilement sont donnés par l'encart en haut à droite. En noir, les flèches symbolisent les liens de navigation permettant de changer de contexte.

Le lien de navigation inverse (de contexte à contexte) permet de revenir au contexte source, dans sa configuration d'origine (avec, dans le Nœud REM du contexte source, le même niveau de détail montré qu'avant l'activation du lien).

6.3.1.3 Exemple

La Figure 6.7 donne une représentation partielle du Modèle S&N dérivé à partir du Modèle des Fonctionnalités tel qu'il a été instancié pour répondre aux besoins d'un utilisateur particulier. Nous réutilisons ici l'exemple introduit dans la section 6.2.2 qui décrit la stratification de l'Espace Fonctionnel de Madame Pad, ainsi que celle d'un de ses Rôles Fonctionnels.

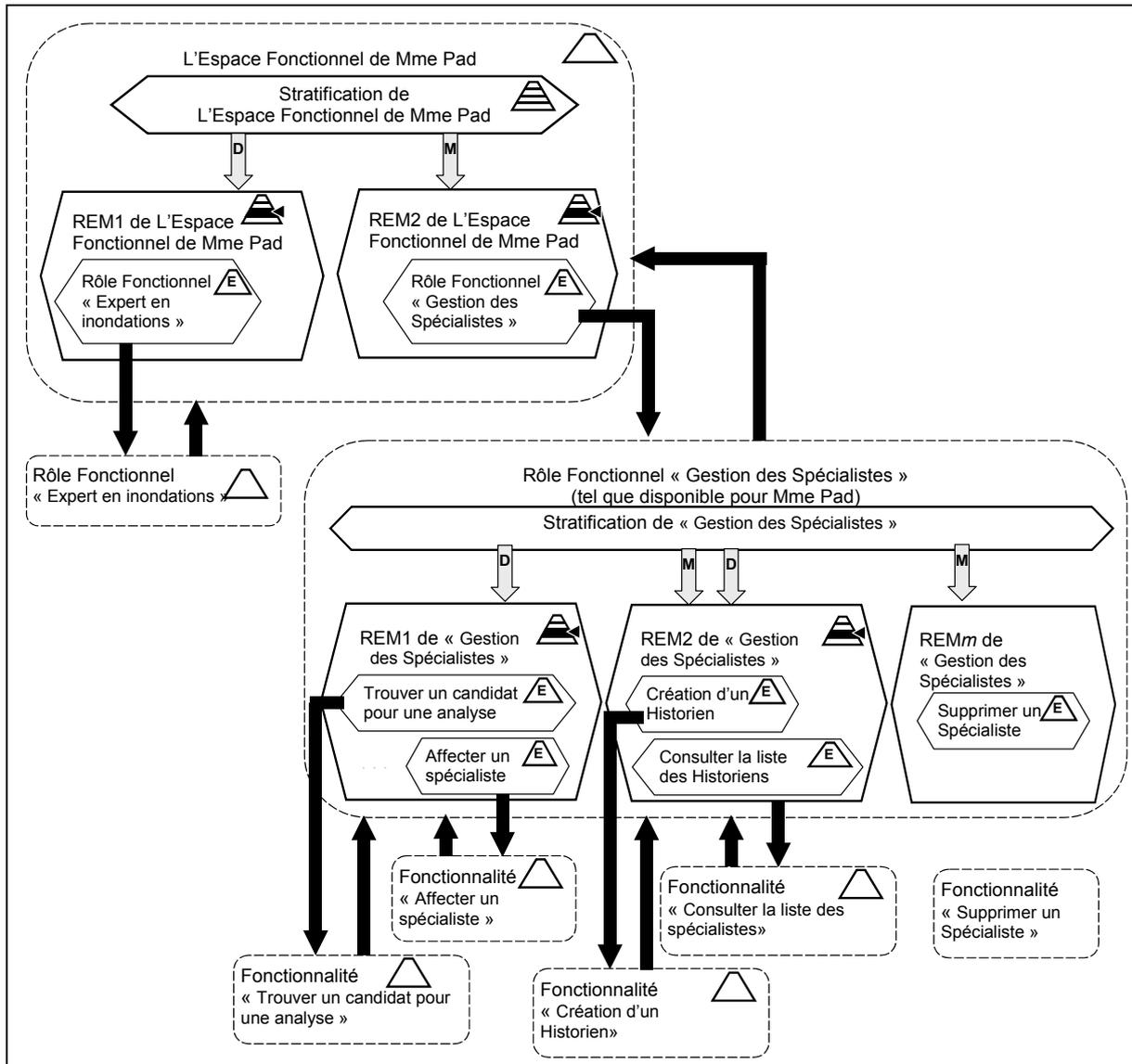


Figure 6.7 – Représentation du Modèle S&N dérivé à partir du Modèle des Fonctionnalités instancié pour un utilisateur particulier.

Les représentations graphiques utilisées ici sont celles présentées dans la Figure 6.6.

Seules les représentations concernant les contextes de l'Espace Fonctionnel de Madame Pad et celles du Rôle Fonctionnel Gestion des Spécialistes sont données entièrement en termes de stratification.

Atteindre le rôle de *Gestion des Spécialistes* nécessite de passer au deuxième niveau de détail de la stratification définie pour l'Espace Fonctionnel. Si le Nœud REM courant est celui du Rôle Fonctionnel *Expert en inondations*, *i.e.* le premier niveau de détail de cette stratification, alors en suivant le lien de dévoilement, l'utilisatrice ajoute à son espace fonctionnel, la partie de l'hypermédia concernant le rôle de gestionnaire. Inversement, depuis ce second niveau de détail, par masquage de la stratification, Madame Pad revient à un espace limité au Rôle Fonctionnel *Expert en inondations*.

Supposons que le contexte dans lequel évolue l'utilisatrice soit celui de son Espace Fonctionnel, tel que le Nœud REM courant est donné par le second niveau de détail de la stratification de cet espace. Par activation du lien de changement de contexte depuis le Nœud EEM correspondant au Rôle Fonctionnel *Gestion des Spécialistes*, elle atteint le contexte associé à ce même rôle, vu cette fois comme une Entité Masquable.

Dans ce contexte, Madame Pad dispose d'une stratification décrite par m REM du Rôle Fonctionnel *Gestion des Spécialistes*. En fonction des liens de dévoilement/masquage qu'elle active, plus ou moins de Fonctionnalités lui sont accessibles. Par suite, en activant le lien de changement de contexte dont est doté tout Nœud EEM associé à une Fonctionnalité, Madame Pad rejoindra le contexte associé (la figure ne détaille pas ces contextes en termes de stratification).

6.3.1.4 Augmentation du modèle par le concepteur

Le Modèle S&N que nous avons présenté caractérise tout SIW conçu en suivant notre approche. Dérivé des spécifications du Modèle des Fonctionnalités, ce modèle décrit la structure et les chemins de navigation de l'hypermédia selon la logique de l'accès progressif. Dans notre approche, ce modèle est suffisant, au sens où le concepteur peut utiliser ces seules spécifications comme ossature de l'hypermédia. Néanmoins, il a la possibilité s'il le souhaite d'ajouter à cette structure des contextes, nœuds et/ou liens.

Les contextes, nœuds et liens avec lesquels le modèle peut être augmenté diffèrent des éléments décrits dans les sections précédentes dans la mesure où ils ne sont pas porteurs d'une sémantique liée à l'accès progressif. Graphiquement, les contextes et nœuds sont identiques aux représentations données à l'exception des symboles repris du MAP qui sont omis. Les liens sont, quant à eux, représentés par des flèches en traits fins. Un exemple des notations utilisées est donné dans la Figure 6.6, en haut à gauche, par le contexte connexion.

L'augmentation du Modèle S&N par le concepteur correspond, par exemple, à l'ajout d'un lien entre deux Fonctionnalités (ayant le même contexte source, *i.e.* appartenant au même Rôle Fonctionnel). Ainsi, dans l'exemple décrit précédemment, le concepteur peut établir un lien de la Fonctionnalité « *Trouver un candidat pour une analyse* » vers celle consistant à « *Affecter un spécialiste* ». Ceci permettant de procéder à une telle affectation directement à partir d'une instance de spécialiste appartenant à la liste des candidats renvoyée par la première fonctionnalité. Le concepteur accroît ainsi les possibilités de navigation dans l'hypermédia.

Le Modèle S&N, augmenté ou non, constitue une donnée du processus de conception que le concepteur va exploiter à l'étape suivante pour construire les pages de l'hypermédia.

6.3.2 Modélisation de la présentation de l'Hypermédia

Le modèle de présentation est destiné à la création des pages Web. Deux dimensions sont considérées :

- l'organisation à la fois des pages et des pages entre elles, à travers la définition de *chartes de composition*, et

- leur apparence basée sur la spécification de *chartes graphiques*.

La notion de charte est à rapprocher d'un modèle de présentation auquel se conforment tous les éléments qu'on lui associe. Notre objectif est de garantir une certaine cohérence entre les différentes pages du site. Cette cohésion de l'interface est, selon nous, particulièrement nécessaire dans les SIW avec accès progressif, les possibilités de navigation étant nombreuses. Bien que très organisés (hiérarchiquement) et associés à une sémantique forte (naviguer pour accéder à plus d'information si besoin), les chemins de navigation peuvent avoir l'effet inverse de celui escompté par l'accès progressif (i.e. désorienter l'utilisateur). Le recours à des chartes nous permet d'offrir des repères à l'utilisateur, tant en ce qui concerne le contenu de chaque page (charte de composition), que l'apparence de ces pages (charte graphique). Toute page générée dans le SIW respecte ces deux spécifications.

6.3.2.1 Chartes de composition

Une charte de composition utilise les concepts de *pages* qui regroupent plusieurs éléments qui doivent être présentés en même temps à l'utilisateur. Une page comporte au moins une région appelée *corps*, et peut, de façon optionnelle, être constituée d'autres régions, dénommées *entête*, *pied de page*, *barre d'outils*, et *barre de navigation*.

Le corps d'une page est constitué de *cadres*, éventuellement imbriqués. Chaque cadre présente une unité sémantique. Les *composants* insérés dans un cadre ont tous une relation avec le concept associé au cadre.

Les composants de cadre sont les éléments qui accueillent le contenu à diffuser. Les autres régions comportent également de tels composants. Les composants permettent d'inclure du contenu obtenu à partir des modèles du SIW (Modèles des Fonctionnalités et du Domaine notamment), toute autre information de présentation (i.e. texte libre par exemple), ou encore du code (applets, scripts, etc.) pour leur associer un comportement spécifique (ouverture de messagerie électronique, impression, etc.).

Le concepteur décrit des pages à l'aide de ces éléments de modélisation. Dans le cadre de notre travail, nous nous intéressons plus particulièrement à la création de pages rendant compte des possibilités d'accès progressif. Dès lors, nous établissons des correspondances entre les éléments de la charte de composition et les concepts du Modèle S&N vus précédemment :

- une page correspond à un contexte ;
- son corps accueille des cadres qui sont attachés à des nœuds appartenant au contexte auquel la page est associée ;
- différents composants sont associés à des mécanismes récurrents de l'accès progressif (par exemple, des composants portent la sémantique des liens de masquage et dévoilement, de changement de contexte, etc.).

Une charte de composition par défaut, basée en partie sur ces correspondances, est proposée au concepteur. Celle-ci décrit plusieurs pages types que le concepteur peut utiliser comme base de composition. La charte de composition par défaut est constituée de trois pages types. Ces pages correspondent respectivement aux trois contextes majeurs identifiés dans le Modèle S&N.

La Figure 6.8 illustre ces pages types. Nous définissons ainsi, la page type « Espace Fonctionnel » (en haut à gauche), la page type « Rôle Fonctionnel » (en haut à droite) et la page type « Fonctionnalité » (en bas à droite). Ces trois pages types présentent des compositions en grande partie similaires. Chacune d'elle comporte un corps composé de deux cadres spécifiques, les cadres Stratification et REM.

Le premier cadre, lié à un Nœud Stratification, permet la présentation des informations relatives à cette stratification. Un composant requête « R » permet d'obtenir certaines informations telles que le nom du Rôle Fonctionnel ou de la Fonctionnalité, etc. Ce cadre contient également deux composants types « M » et « D » qui correspondent aux mécanismes de masquage/dévoilement.

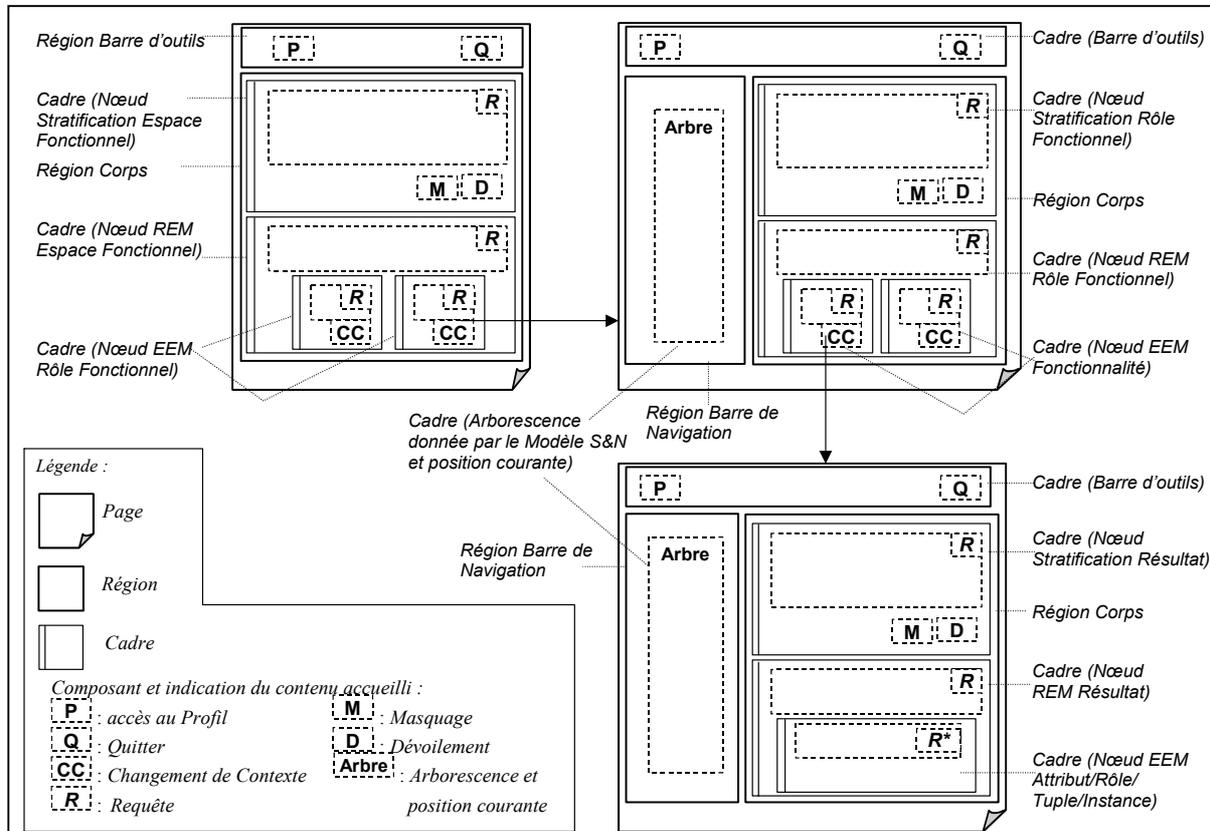


Figure 6.8 – Représentation graphique de la charte de composition par défaut

Le second cadre est lié à tous les Nœuds REM du contexte correspondant à cette page. Il est destiné à délivrer la représentation associée au niveau de détail courant. Lorsqu'un composant dédié au masquage/dévoilement est activé dans le cadre Stratification, le contenu du cadre REM est mis à jour en conséquence. Cette mise à jour du cadre se traduit par l'ajout ou le retrait (dévoilement/masquage) de cadres internes attachés aux Nœuds EEM. Par défaut, le contenu présenté initialement est donné par la REM de plus bas niveau de détail.

Dans le cas des pages Espaces Fonctionnels et Rôle Fonctionnel types, chaque Nœud EEM est associé à un cadre distinct qui comprend notamment le composant « CC » traduisant la sémantique du changement de contexte. La flèche partant d'un composant CC aboutit à la page type concernée. En ce qui concerne la page type Fonctionnalité, ceci n'est pas le cas : tous les nœuds EEM sont regroupés dans le même cadre (les informations les concernant résultent d'une requête appliquée plusieurs fois 'R*'). Néanmoins, cette représentation est la configuration la plus simple. Dans l'optique où le concepteur définit un lien de cette fonctionnalité vers une autre³⁶, telle que chaque EEM soit paramètre de la fonctionnalité cible, alors on spécifie que

³⁶ Par exemple, comme lorsqu'une Fonctionnalité permettant de consulter la liste des Historien est liée à celle permettant la consultation de l'information sur leurs subordonnés.

chaque Nœud EEM est accueilli par un cadre qui comporte un composant associé établissant le lien vers la fonctionnalité cible.

Dans le cas des pages types Rôle Fonctionnel et Fonctionnalité, une région de type barre de navigation contient le composant « Arbre ». Ce composant permet d'afficher l'arborescence de l'hypermédia donnée par le Modèle S&N et présentée sous forme de dossiers/sous-dossiers. La hiérarchie montrée est celle de l'espace fonctionnel, des rôles fonctionnels qui le composent et pour chacun, des fonctionnalités. Chaque élément de cette arborescence est présenté au niveau de complétude atteint par l'utilisateur.

On retrouve dans chacune de ces pages une région barre d'outils permettant à l'utilisateur d'accéder à son profil (cf. section 6.4) ou de quitter le SIW.

Dans la Figure 6.8, les différents éléments d'une page sont présentés selon une organisation spatiale particulière. Celle-ci est en fait décrite par la charte graphique décrite dans la section 6.3.2.2 (il s'agit ici de la charte graphique par défaut).

6.3.2.2 Charte graphique

La charte de composition décrit les différentes régions d'une page et leur composition permettant ainsi d'homogénéiser les pages du site en termes de types de contenu. Une charte graphique vise un objectif similaire, mais porte sur l'apparence. Elle définit des règles d'organisation spatiale (dans les pages, dans les régions, etc.), ainsi que des caractéristiques visuelles (couleurs, polices, arrières plans, etc.).

Nous proposons au concepteur plusieurs présentations types qui s'appliquent aux différents éléments de la charte de composition.

Une présentation type est associée à une page type. Elle décrit essentiellement le positionnement des régions dans la page type. Ce positionnement est spécifié de telle façon que le vide laissé par l'absence éventuelle d'une région est automatiquement comblé. Ce cas se présente notamment lorsqu'un utilisateur souhaite cacher une région particulière (*e.g.* la barre d'outil). De plus, une présentation type associée à une page type décrit les attributs visuels que le concepteur souhaite voir partagés par toutes les régions, cadres et composants (*e.g.* pour utiliser la même police de caractères dans tous ces éléments).

Chaque type de région rencontré dans une page est associé à une présentation type spécifique. Celle-ci décrit le positionnement des éventuels éléments inclus (*e.g.* celui des cadres associés respectivement à un Nœud Stratification et à un Nœud REM dans le corps) ainsi que les attributs graphiques partagés à l'intérieur de cette région.

Et, selon le même principe descendant, chaque type de cadre et chaque type de composant sont associés à une présentation type. Dans le cas des composants, la présentation type détermine notamment s'il est proposé sous la forme d'un lien, d'un bouton, d'un tableau, etc.

Les présentations types qui constituent la charte graphique par défaut restent relativement simples. Nos objectifs, à travers elles, sont de deux ordres. Nous avons déjà souligné le premier qui est de garantir une certaine cohérence visuelle entre les éléments de sémantique équivalente dans les pages du SIW. Le second objectif est de veiller à ce que, dans l'interface livrée à l'utilisateur, les éléments relatifs à l'accès progressif soient suffisamment différenciables des attributs classiques d'un hypermédia. Notamment, afin de faciliter l'utilisation des mécanismes du protocole d'exploitation du MAP (masquage/dévoilement, changement de contexte, etc.), les composants qui les mettent en œuvre ne doivent pas être confondus avec des liens de navigation traditionnels. Plus qu'un confort à l'utilisation, ceci est, selon nous, un atout pour l'appropriation des principes de l'accès progressif par l'utilisateur.

6.3.3 Conclusion sur le Modèle de l'Hypermédia

Le Modèle de l'Hypermédia que nous proposons se divise en deux axes. Le premier axe concerne la Structuration et la Navigation. Le sous-modèle qui leur est dédié offre une représentation logique des éléments relatifs à l'accès progressif inspirée des concepts hypermédia. Cette représentation est directement dérivée du Modèle des Fonctionnalités : chacun des trois niveaux fonctionnels (Espace Fonctionnel, Rôle Fonctionnel et Fonctionnalité) est traduit par un contexte. Ce contexte est structuré en différents types de nœuds et de liens respectant la logique de l'accès progressif. Nous avons décrit comment naviguer à l'intérieur d'un contexte, en exploitant le masquage et le dévoilement de stratifications, ainsi que les possibilités de passage d'un contexte à l'autre. Il résulte de ce modèle une organisation arborescente de l'hypermédia par le biais d'une navigation en profondeur, entre les contextes de niveaux différents (e.g. du contexte Espace Fonctionnel aux contextes Rôles Fonctionnels). Cette organisation a, de plus, la particularité d'être extensible transversalement : le dévoilement d'une stratification dans un contexte donnant la possibilité d'accéder à d'autres contextes de niveau inférieur (e.g. le dévoilement de la stratification de l'Espace Fonctionnel permettant d'atteindre des contextes Rôles Fonctionnels non visibles jusque là).

Le second axe du Modèle de l'Hypermédia porte sur les aspects relatifs à la présentation. Un modèle, basé sur la définition de chartes, décrit comment sont créées les pages de l'hypermédia. Les chartes s'appuient les concepts du Modèles S&N pour définir la composition des pages et l'organisation entre elles, ainsi que l'apparence qu'on souhaite leur donner. Ces deux dimensions sont prises en compte respectivement par les chartes de composition et de présentation. Nous avons recours à la définition de pages types et de présentations types auxquelles se conforment les pages Web et les composants qu'elles englobent. Notre objectif est de garantir une certaine cohésion des différentes pages du site qui, par ailleurs, rendent compte de façon intuitive de l'accès progressif.

Dans la section suivante, nous présentons le Modèle des Utilisateurs. Conformément aux principes que nous avons suivis jusqu'ici, nous montrons comment ce modèle participe à la mise en œuvre de l'accès progressif, notamment à travers les possibilités d'adaptation des spécifications proposées jusqu'ici.

6.4. Le Modèle des Utilisateurs

Dans une démarche d'analyse et de conception basée sur UML, l'utilisateur du futur système apparaît comme un *acteur*, i.e. en tant qu'entité externe au système (i.e. non représentée dans le système) et qui interagit avec ce dernier. En revanche, nous avons montré, lors de l'étude des différentes méthodes de conception d'applications sur le Web (cf. chapitre 4), que celles-ci utilisent de plus en plus une représentation interne de l'utilisateur. En effet, dans l'optique de concevoir des systèmes capables de s'adapter à leurs utilisateurs, acquérir et utiliser des informations à leur propos est en effet incontournable.

Notre approche considère donc l'utilisateur comme une entité représentée à l'intérieur du système au moyen d'un modèle spécifique ; ce *Modèle des Utilisateurs* constitue la source d'informations nécessaires à la mise en œuvre de l'adaptabilité du SIW. Nous montrons dans cette section que différentes connexions du Modèle des Utilisateurs avec les autres modèles de l'approche permettent cette mise en œuvre.

Le Modèle des Utilisateurs est un support pour une adaptation destinée à des groupes d'utilisateurs, mais également à des utilisateurs individuels. Cette distinction est présentée dans la section suivante.

6.4.1 Distinction entre Groupes et Utilisateurs

6.4.1.1 Motivations

Les SIW sont destinés à un public varié composé d'individus ne présentant pas les mêmes besoins à l'égard du système, n'ayant pas les mêmes droits sur les données de celui-ci, faisant preuves d'expériences ou de connaissances variées, exprimant des préférences diverses, etc. L'identification de différents groupes d'utilisateurs est un moyen de réduire la complexité sous-jacente à une telle hétérogénéité.

Néanmoins, bien qu'appartenant à un même groupe, les utilisateurs présentent des caractéristiques individuelles variant parfois dans une large mesure. En ce sens, que le système soit capable de s'adapter à différents groupes n'est pas suffisant. La prise en compte d'un utilisateur en tant qu'individu est également nécessaire. L'objectif est de proposer en priorité des adaptations basées sur des spécifications personnelles ; en l'absence de celles-ci, les spécifications des groupes sont alors utilisées.

Enfin, un utilisateur donné peut également, d'une fois à l'autre, avoir des besoins différents, préférer une option à une autre, etc. A cet égard, on ne peut se satisfaire d'une représentation figée dans le temps de l'utilisateur pour mettre en œuvre l'adaptabilité.

En matière de modélisation des utilisateurs, notre approche vise trois objectifs : *i*) considérer la représentation de différents groupes, *ii*) représenter les utilisateurs individuellement et *iii*) garantir la possibilité d'exprimer toute évolution de ces représentations. Ceci contribue, selon nous, à tendre vers de meilleures capacités d'adaptation du système.

6.4.1.2 Une distinction basée sur une approche fonctionnelle

Le concept de *Groupe d'utilisateurs* (ou *groupe*) permet de fédérer les informations relatives à plusieurs utilisateurs qui partagent certaines caractéristiques. Dans notre approche, le critère de regroupement des utilisateurs individuels est fondé sur le concept de Rôle Fonctionnel présenté dans la section 6.2. La Figure 6.9 propose un diagramme de classes pour illustrer cette modélisation.

Une instance de la classe Groupe est associée à une instance de la classe Rôle Fonctionnel, cette dernière étant constituée de Fonctionnalités. Tous les membres de la classe Groupe (i.e. donnés par la classe Utilisateur) ont ainsi en commun le fait d'avoir accès à l'ensemble des Fonctionnalités qui constituent le Rôle Fonctionnel associé à ce groupe. L'identification d'un groupe dans le Modèle des Utilisateurs relève donc de celle des différents Rôles Fonctionnels (ceux-ci étant directement obtenus à partir des cas d'utilisation – cf. section 6.2.1).

Un Utilisateur peut appartenir à un ou plusieurs groupe(s), et peut, en conséquence, remplir les différents Rôles Fonctionnels associés aux groupes dont il est membre. L'Espace Fonctionnel dont un Utilisateur est propriétaire est constitué de ces Rôles Fonctionnels. L'association est constitué de entre un Espace Fonctionnel et des Rôles Fonctionnels est ainsi déductible de l'association appartient entre les classes Utilisateur et Groupe.

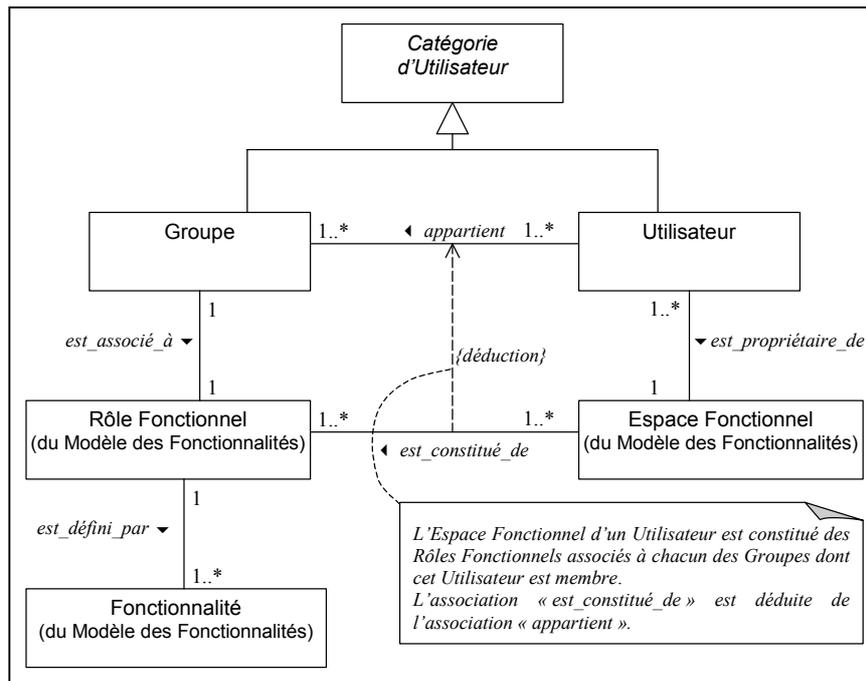


Figure 6.9 – Modèle des Utilisateurs : distinction entre Groupes et Utilisateurs basée sur les concepts du Modèle des Fonctionnalités

6.4.2 Profils de groupes et d'utilisateurs

Après avoir introduit les notions de Groupe et d'Utilisateur, nous décrivons les informations les concernant. Notons que nous ne mentionnons pas, par souci de clarté, un certain nombre d'informations qui ont vocation à identifier et qualifier les instances des classes Groupe et Utilisateur. Il s'agit classiquement d'attributs tels que nom et description pour les groupes, ou pour les utilisateurs, d'attributs pour l'authentification (login, mot de passe), l'identification (nom, prénom de l'utilisateur qui permettent une personnalisation très simple par affichage de ces données dans la page d'accueil et donne le sentiment à l'utilisateur d'être un interlocuteur privilégié [Kram00]), etc.

Les autres types d'informations modélisées sont répartis en différentes catégories que nous appelons *profils*. La Figure 6.10 illustre la prise en compte de tels profils dans le Modèle des Utilisateurs. Notons que les profils considérés dans le cadre de ce travail sont ceux directement en rapport avec les deux modèles que nous avons présentés précédemment, *i.e.* le Modèle des Fonctionnalités et le Modèle de l'Hypermédia.

La classe Groupe est liée par une composition à trois *profils de groupe* modélisés par la classe G_Profil. Celle-ci se spécialise en G_AccèsProgressif, G_Préférences et G_Comportement.

De façon similaire, la classe Utilisateur est liée par une composition à, au plus, trois *profils d'utilisateur* que modélise la classe U_Profil. Une spécialisation de cette classe, similaire à celle de la classe G_Profil est proposée : on distingue les classes U_AccèsProgressif, U_Préférences et U_Comportement.

Les profils de groupe entretiennent des relations avec les profils correspondants décrits pour les utilisateurs. Ces relations sont représentées par les associations entre les sous-classes de G_Profil et U_Profil dans la Figure 6.10.

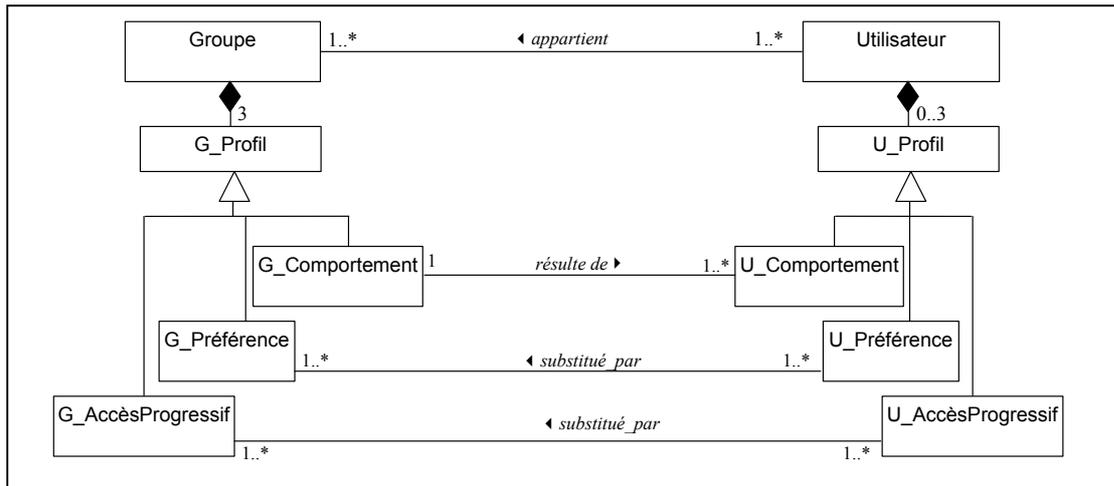


Figure 6.10 – Le Modèles des Utilisateurs : profils décrits pour les groupes et les utilisateurs

Dans les sections suivantes, nous décrivons comment la mise en œuvre des capacités d’adaptation du SIW est supportée par le Modèle des Utilisateurs à travers ces profils. De façon plus précise, nous montrons comment :

- l’adaptation des stratifications est garantie par les profils d’Accès Progressif G_AccèsProgressif et U_AccèsProgressif ;
- l’adaptation des aspects relatifs à la présentation de l’hypermédia est supportée par les Profils de Préférences G_Préférences et U_Préférences ; et enfin,
- les Profils de Comportement G_Comportement et U_Comportement introduisent les principes d’une adaptativité du SIW en matière d’accès progressif.

Nous décrivons également les liens entre les versions d’un même profil, dédiées respectivement à un groupe et à un utilisateur.

6.4.3 Adaptabilité des stratifications : les Profils d’Accès Progressif

Les profils d’Accès Progressif sont destinés à maintenir les descriptions de stratification définies spécifiquement pour un groupe ou un utilisateur afin de répondre au mieux à leurs attentes en la matière. Pour cela, il est tout d’abord nécessaire de pouvoir exprimer des stratifications adaptées à ces groupes et utilisateurs. Ceci passe par une relation établie entre le MAP et le Modèle des Utilisateurs.

6.4.3.1 Le lien entre le MAP et le Modèle des Utilisateurs

La possibilité de définir des stratifications adaptées aux groupes et aux utilisateurs est liée à une caractéristique du MAP que nous n’avons pas mentionnée au chapitre 5 afin de simplifier la présentation du modèle. L’association binaire *est_définie_pour* entre les classes *Stratification* et *Entité Masquable* telle que présentée alors (cf. page 85) est en fait une association ternaire impliquant également la surclasse *Catégorie d’Utilisateur* des classes *Groupe* et *Utilisateur*. La Figure 6.11 illustre cette caractéristique. Nous ne reproduisons que les éléments affectés par celle-ci, le reste du MAP étant inchangé et renommons cette association par « *définition* » afin d’éviter les confusions.

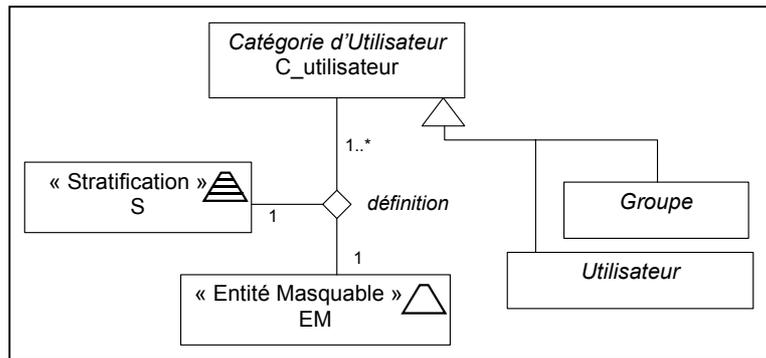


Figure 6.11 – Modélisation pour une stratification adaptée aux groupes et aux utilisateurs

Les multiplicités de l'association *définition* expriment que :

- une entité masquable correspond à un couple d'instances (un_C_Utilisateur, une_S) donné. Nous précisons qu'en réalité, la multiplicité portée à 1 du côté de l'entité masquable est indépendante de l'utilisateur. Elle est conservée de l'association *est définie pour* qui spécifie qu'une stratification correspond à une et une seule entité masquable.
- plusieurs utilisateurs peuvent être liés à un même couple d'instances (un_EM, une_S) donné, ce qui permet à plusieurs utilisateurs de partager la même stratification d'entité masquable.
- une seule stratification est définie pour un couple d'instances (un_C_Utilisateur, une_EM) donné.

La description des Profils d'Accès Progressif de Catégorie d'Utilisateur est basée sur une exploitation de cette modélisation. Elle est présentée ci-après en distinguant les profils G_AccèsProgressif et U_AccèsProgressif.

6.4.3.2 Description des Profils d'Accès Progressif

Lorsque la classe Catégorie d'Utilisateur est substituée par sa sous-classe Groupe dans la modélisation proposée par la Figure 6.11, nous considérons que le type de l'association *définition* est le type-record [Card91] :

<g :Groupe, s: Stratification, em: EM>

Le profil G_AccèsProgressif d'un groupe g_1 est donné par l'ensemble de toutes les stratifications s_i tel que s_i est la valeur du rôle s dans tout tuple de l'association *définition* vue comme la record-value :

<g = g_1 , s = s_i , em = em_k >

Lorsque la classe Catégorie d'Utilisateur est substituée par sa sous-classe Utilisateur, nous considérons que le type de l'association *définition* est le type-record [Card91] :

<u : Utilisateur , s: Stratification, em: EM>

Le profil U_AccèsProgressif d'un utilisateur u_1 est donné par l'ensemble de toutes les stratifications s_i tel que s_i est la valeur du rôle s dans tout tuple de l'association *définition* vue comme la record-value :

<u = u_1 , s = s_i , em = em_k >

L'association nommée « *substituée par* » dans la Figure 6.11 décrit la relation établie entre les profils G_AccèsProgressif et U_AccèsProgressif. Il s'agit d'une relation de substitution qui

permet d'affecter à l'utilisateur le profil d'Accès Progressif du groupe lorsqu'un profil spécifique à l'utilisateur n'est pas défini pour une stratification. Cette relation est telle que :

Pour toute instance u_n de la classe Utilisateur et toute instance g_m de la classe Groupe, telles qu'il existe un tuple vu comme la record-value $\langle u = u_n, g = g_m \rangle$ du type donné par le record-type $\langle u = \text{Utilisateur}, g = \text{Groupe} \rangle$ décrivant l'association appartient entre ces classes, s'il n'existe pas de tuple $\langle u = u_n, s = s_i, em = em_k \rangle$ défini pour le couple de valeurs $\langle u = u_n, em = em_k \rangle$,

alors, lorsque l'utilisateur u_n accède, en tant que membre du groupe g_m à l'EM em_k ,

s'il existe une stratification s_j telle que s_j est la valeur du rôle s dans tout tuple de l'association *définition* impliquant la Classe Groupe, vu comme la record-value :

$$\langle g = g_1, s = s_j, em = em_k \rangle$$

alors, l'accès à l'EM em_k par l'utilisateur u_n respectera les spécifications de la stratification s_j donnée par le profil d'Accès Progressif du groupe g_m .

Ces définitions des profils d'Accès Progressif et de la relation de substitution qui les unit sont valables pour les Entités Masquables Résultats de Requêtes utilisées par les Fonctionnalités et Rôles Fonctionnels (et ce, qu'elles soient stratifiées en intension ou en extension). L'EM Espace Fonctionnel constitue une exception dans la mesure où l'espace fonctionnel est un concept personnel qui n'a pas de sens au niveau du groupe. Il n'existe pas de définition de stratification adaptée au groupe pour cette EM, et, par voie de conséquence, la relation de substitution ne peut être appliquée.

Le protocole d'exploitation du MAP que nous avons présenté au chapitre 5 ne faisait pas référence aux capacités d'adaptation des stratifications. Néanmoins, l'introduction de cette dimension a un impact sur toutes les opérations que nous avons définies dans ce protocole (cf. section 5.2). Nous abordons cet aspect ci-dessous.

6.4.3.3 Impact de l'adaptabilité sur la gestion des stratifications

De façon évidente, en ce qui concerne la création d'une stratification, celle-ci doit désormais être liée à son destinataire. Et, logiquement, l'accès à une EM par un utilisateur, consiste à délivrer la stratification qui est la plus adaptée à celui-ci. On interrogera d'abord le profil $U_AccèsProgressif$ de cet utilisateur, puis à défaut d'avoir trouvé une stratification *ad-hoc*, le profil du Groupe sera consulté.

L'impact de l'adaptabilité est également noté dans le cas de la suppression d'une stratification. Cette opération se décline désormais en deux versions. Soit il s'agit de supprimer une stratification définie pour un Groupe, soit de supprimer une stratification définie pour un utilisateur individuel.

La suppression d'une stratification personnelle sp d'un utilisateur u pour une EM em consiste à détruire l'objet sp et tous les objets rem_j de cette stratification. Les liaisons dans lesquelles étaient impliqués ces objets sont alors détruites (i.e. le lien de référence entre sp et em , et ceux unissant les objets rem_j et leurs objets eem_k respectifs). De même, la liaison entre le profil $U_AccèsProgressif$ de l'utilisateur u cette stratification sp est supprimée. En revanche, les objets u , em et eem_k ne sont pas affectés par cette suppression.

Lorsqu'une stratification définie pour un groupe est à supprimer, il convient de prêter attention aux deux cas suivants :

- s'il s'agit d'une stratification s définie pour un seul groupe g , ou partagée par plusieurs groupes g_i mais qui n'a plus de sens pour aucun de ceux-ci, le processus décrit ci-dessus est repris ; on supprime la référence du Profil G_AccèsProgressif de g ou de tous les profils des groupes g_i à la stratification s , les liens entre les rem_j de s et leurs eem_k sont détruits, et s et ses objets rem_j sont supprimés.
- S'il s'agit d'une stratification partagée par plusieurs groupes g_i , dont le groupe particulier g , seul groupe pour lequel il est souhaitable de supprimer la stratification, alors on supprime la référence du Profil G_AccèsProgressif de g à la stratification s . Les autres groupes conservent la stratification s définie pour em .

On note que tout utilisateur d'un groupe pour lequel une stratification est supprimée conserve l'usage de la stratification qu'il aurait personnellement définie (ou aurait été définie par le concepteur à son attention).

Après avoir décrit les profils d'Accès Progressifs et leur implication dans la mise en œuvre de l'adaptabilité des stratifications, nous montrons brièvement comment de façon analogue, les profils de Présentation concourent à la production d'hypermédias adaptés.

6.4.4 Adaptabilité de la présentation : les Profils d'Accès Progressif et de Préférences

L'adaptabilité de la structure et de l'apparence de l'hypermédia résulte de spécifications maintenues par les profils d'Accès Progressif d'une part, et de Préférences d'autre part.

6.4.4.1 Adaptabilité de la composition résultant des profils d'Accès Progressif

Dans la section consacrée au Modèle de L'Hypermédia, nous avons montré que la définition de son ossature est donnée par le Modèle de Structuration et Navigation. Nous rappelons que les concepts décrits par le Modèle S&N reposent fortement sur des correspondances établies avec les trois concepts majeurs du Modèle des Fonctionnalités, i.e. Espace Fonctionnel, Rôle Fonctionnel et Fonctionnalité. L'application du MAP au Modèle des Fonctionnalités nous a amené à considérer chacun de ces trois concepts comme des Entités Masquables. Dans la section précédente, nous avons exposé comment le Modèle des Utilisateurs permet de décrire des stratifications adaptées, quel que soit le type d'EM. En particulier, des stratifications adaptées du Modèle des Fonctionnalités peuvent être décrites.

Dès lors, le Modèle S&N dérivé à partir d'une instance du Modèle des Fonctionnalités constitue déjà une représentation adaptée de l'hypermédia qui sera proposé à l'utilisateur :

- d'une part, cette instance de modèle décrit l'espace, les rôles et les fonctionnalités auxquels peut prétendre un utilisateur et,
- d'autre part, le profil d'Accès Progressif de cet utilisateur définit comment ceux-ci sont stratifiés.

L'exemple suivant illustre la façon dont l'adaptabilité décrite ci-dessus est portée par le Modèle S&N. Considérons le Rôle Fonctionnel de *gestion de spécialistes* qui regroupe toutes les fonctionnalités permettant de gérer les spécialistes du centre documentaire que deux utilisateurs peuvent remplir, en ayant défini des stratifications différentes de celui-ci :

- Mme Airbe, particulièrement concernée par le recrutement de ces spécialistes, définit une stratification de ce rôle fonctionnel telle qu'au premier niveau de détail, les fonctionnalités en rapport avec le recrutement sont accessibles (création de nouveaux spécialistes, historiens et géographes, modification des données et consultation des informations les concernant). Au niveau de détail suivant, elle accède aux autres fonctionnalités de moindre intérêt par rapport à ses préoccupations (e.g. trouver un candidat pour effectuer une analyse

documentaire et l'affecter à celle-ci). La Figure 6.12 illustre le contexte du Rôle Fonctionnel de *Gestion des Spécialistes* tel qu'il est dérivé, selon les descriptions du Modèle S&N, à partir de l'instance du Modèle de Fonctionnalités concernant cette utilisatrice et de son profil d'Accès Progressif.

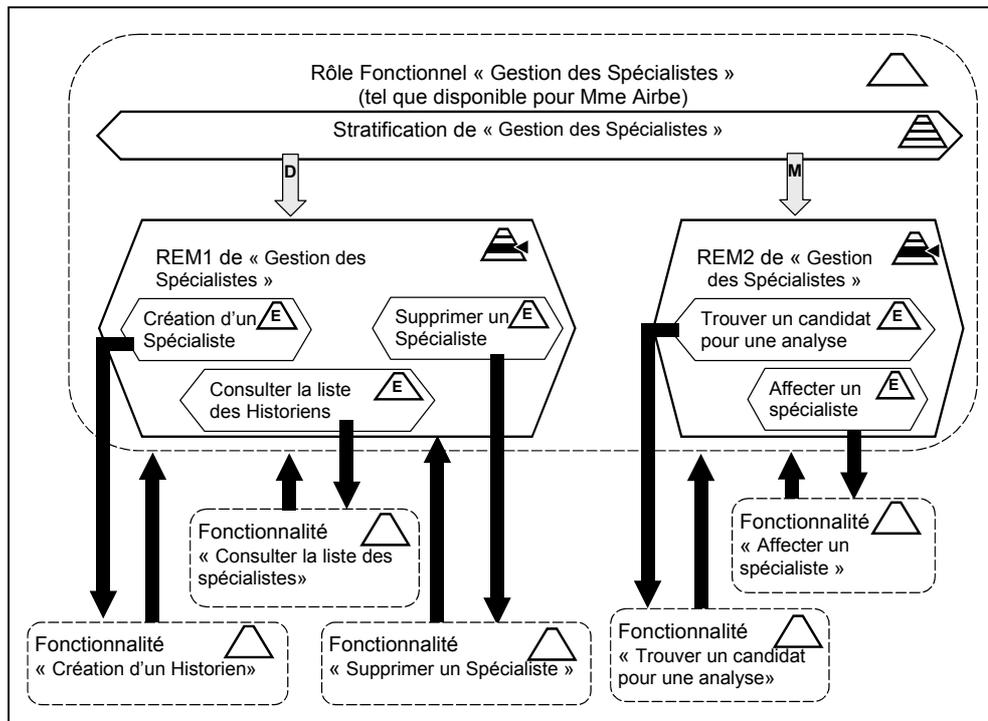


Figure 6.12 – Contexte extrait du Modèle S&N correspondant à un Rôle Fonctionnel
Ce Rôle Fonctionnel est une instance attachée au Modèle des Fonctionnalités et est stratifié conformément au profil d'Accès Progressif d'un utilisateur donné.

Observons à présent le cas d'un second utilisateur, utilisant le même Rôle Fonctionnel, mais l'ayant stratifié différemment :

Nous avons déjà mentionné l'existence, pour Mme Pad (cf. page 128), d'une stratification différente pour ce même Rôle Fonctionnel. Pour celle-ci, le premier niveau de détail donne accès aux fonctionnalités liées à l'affectation des analyses documentaires aux spécialistes (e.g. recherche d'un candidat potentiel pour une analyse, attribution d'une analyse, envoi du message prévenant ce rapporteur, etc.). Le reste de la stratification permet de dévoiler progressivement d'autres fonctionnalités, et notamment celles auxquelles Mme Airbe a, pour sa part, choisi de définir un accès plus immédiat. En l'absence de Mme Airbe, Mme Pad devant assurer les activités relatives au recrutement des spécialistes, elle peut accéder aux fonctionnalités correspondantes en dévoilant la stratification de ce Rôle Fonctionnel. De façon similaire au cas précédent, nous présentons le contexte du Rôle Fonctionnel de *Gestion des Spécialistes* tel qu'il est dérivé, selon les descriptions du Modèle S&N, à partir de l'instance du Modèle de Fonctionnalités concernant cette seconde utilisatrice et de son profil d'Accès Progressif (cf. Figure 6.13).

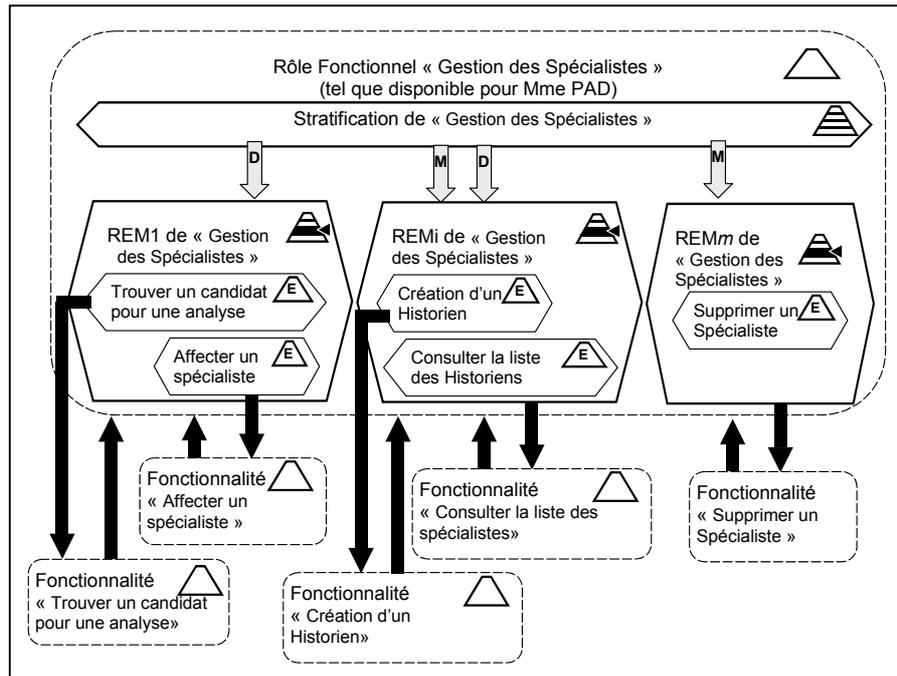


Figure 6.13 – Contexte pour un profil d'Accès Progressif différent
 Ce contexte correspond au même Rôle Fonctionnel que précédemment mais est un extrait du Modèle S&N dérivé du Modèle des Fonctionnalités pour un profil d'Accès Progressif différent.

Ces deux exemples nous permettent d'observer que la structure de l'hypermédia du Modèle S&N est globalement identique quel que soit l'utilisateur auquel on le propose. Ceci résulte de la dérivation systématique du Modèle S&N à partir d'une instance de Modèle des Fonctionnalités. L'adaptabilité néanmoins inhérente à ce modèle provient de la prise en compte des profils d'Accès Progressif défini dans la section précédente. Bien que remplissant le même Rôle Fonctionnel à l'égard du système, deux utilisateurs dont les profils d'Accès Progressifs sont distincts disposent de deux organisations arborescentes de l'hypermédia spécifiques.

Basé sur les profils d'Accès Progressif, le modèle S&N de l'hypermédia dérivé pour un groupe est utilisé en cas d'impossibilité de dérivation spécifique à un utilisateur (*i.e.* la relation de substitution entre G_AccèsProgressif et U_AccèsProgressif est utilisée). On note toutefois que l'impact de cette substitution n'a d'effet que sur les contextes de Rôles Fonctionnels et de Fonctionnalités, le contexte d'Espace Fonctionnel ne pouvant faire l'objet d'une stratification que dans un profil U_AccèsProgressif.

Nous expliquons brièvement comment l'hypermédia peut également être adapté en terme d'apparence, et ce spécifiquement à un groupe ou à un utilisateur

6.4.4.2 Adaptabilité de l'apparence grâce aux profils de Préférences

L'adaptation de l'apparence d'un hypermédia repose sur la définition de chartes de composition et/ou de présentation propres à chaque groupe ou à chaque utilisateur. Les profils de Présentation G_Préférences et U_Préférences comportent des références aux pages types et aux présentations types à appliquer.

La relation de substitution existant, entre ces versions de groupe et personnelle de profil, est définie de façon analogue à celle décrite pour les Profils d'Accès Progressif (cf. section 6.4.3.2).

La Figure 6.14 illustre comment, à partir d'un même contexte issu d'un Modèle S&N, deux pages différentes en terme de composition et d'apparence sont créées. Ceci résulte de l'exploitation de deux profils de Préférences distincts qui référencent des pages types et présentations types différentes. L'exemple est tiré d'un SIW pour l'achat en ligne d'articles de sports, conçu et généré avec l'environnement KIWIS qui implémente notre proposition (cf. chapitre 8).

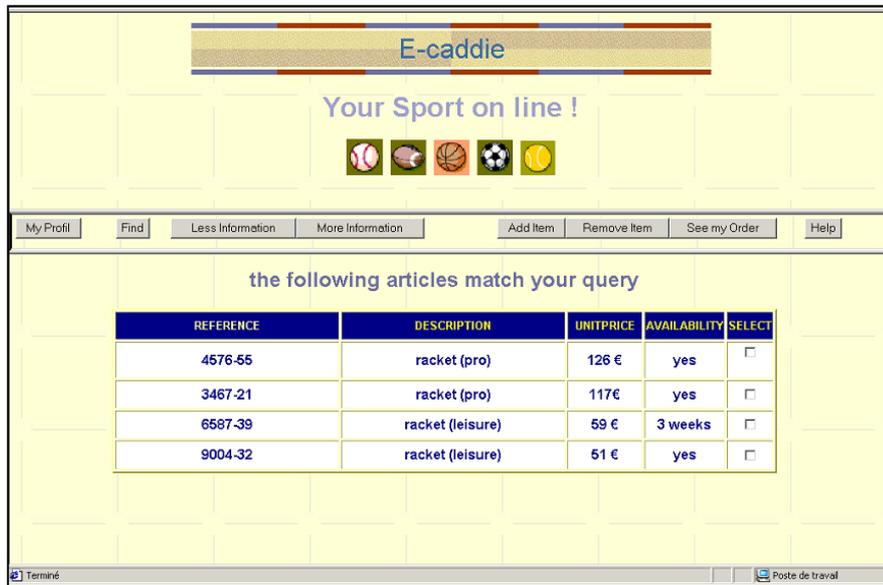


Figure 6.14 – Deux exemples de pages créées à partir du même contexte d'un Modèle S&N. L'apparence des pages diffère suite à l'application des spécifications maintenues dans deux profils de Préférences distincts.

Nous abordons le dernier point relatif aux capacités d'adaptation que le Modèle des Utilisateurs est destiné à supporter. Nous introduisons, grâce aux profils de Comportement, les possibilités d'adaptativité en matière d'accès progressif que nous souhaitons mettre en place.

6.4.5 *Adaptativité du SIW : les profils de Comportement.*

Les comportements d'un utilisateur face au SIW permettent d'analyser l'utilisation qui est faite du système. Ces comportements sont principalement constitués des interactions utilisateur/système que l'on détecte le plus souvent à travers l'action de boutons de commandes, l'activation de liens hypertextes, etc. D'autres indicateurs du comportement peuvent être observés comme, par exemple, le temps passé sur une page, ou encore l'analyse des mouvements oculaires de l'utilisateur.

L'analyse de ces informations comportementales permet d'établir un diagnostic d'utilisation du SIW. Ce diagnostic permet de mettre en évidence certains dysfonctionnements ou inadéquations du système. Par exemple, une fonctionnalité offerte au premier niveau de REM qui n'est que très rarement, voire jamais, utilisée pourra être détectée. Les raisons de cette non-utilisation (complexité, inutilité, incompréhension, etc.) sont ensuite à déterminer ainsi que les actions à entreprendre (modification de la stratification pour que la fonctionnalité apparaisse à un niveau de détail plus élevé, suppression de celle-ci, aide à l'utilisateur, etc.). A l'inverse, la détection d'une fonctionnalité extrêmement exploitée, bien que son utilisation nécessite de dévoiler plusieurs fois la stratification avant de l'atteindre, incite à en faciliter l'usage en redéfinissant la stratification en conséquence.

Lors de la présentation du MAP, nous avons décrit des opérations permettant de réorganiser les stratifications (*cf.* section 5.2.3). Ces opérations décrivent :

- comment ajouter ou supprimer un élément dans une REM,
- ajouter, supprimer ou déplacer une REM dans une stratification, et
- comment fusionner plusieurs REM en une seule, ou l'opération inverse, fractionner une REM en plusieurs.

Coupler ces opérations avec des mécanismes de détection du comportement des utilisateurs et en automatiser l'application, constituent un gage d'adaptativité du SIW.

En amont de la détection des comportements, une modélisation des informations comportementales de l'utilisateur est nécessaire afin d'établir quels sont les comportements à observer, comment interpréter leur détection, quelle action entreprendre suite à celle-ci, etc.

A titre d'exemple, il nous paraît judicieux de détecter la répétition, dans un laps de temps très court, d'une action de dévoilement. Le dévoilement permettant de passer à un niveau de détail supérieur, si celui-ci est immédiatement et systématiquement suivi d'un autre dévoilement, alors nous pouvons envisager que l'utilisateur « passe » volontairement ce niveau intermédiaire. Ceci signifie certainement que l'information dont il a besoin est accessible trop loin dans la stratification (i.e. à un niveau de détail trop élevé) et qu'en revanche des informations inutiles lui sont délivrées à un bas niveau de détail. L'appel à une fonction de réorganisation de la stratification concernée permettrait de la réadapter en conséquence.

Dans cette perspective, le profil *U_Comportement* vise à stocker les informations détectées à partir des actions de l'utilisateur. Le profil *G_Comportement* est destiné à stocker une représentation d'informations comportementales calculées à partir de tous les profils *U_Comportement* des utilisateurs membres du groupe auquel il est attaché. Des calculs basés la moyenne des comportements individuels, ou retenant uniquement les actions les plus fréquentes, permettent d'établir un profil de groupe qui sera éventuellement utilisé pour procéder à des adaptations des stratifications définies pour le groupe.

6.4.6 Conclusion sur le Modèle des Utilisateurs

Le Modèle des Utilisateurs que nous proposons répond plus particulièrement aux besoins de SIW pour lesquels des modalités d'accès progressif doivent être adaptées aux différents groupes d'utilisateurs et utilisateurs individuels.

Ce modèle différencie les notions de groupes et d'utilisateurs. Les groupes sont définis selon une approche fonctionnelle exploitant les spécifications du Modèle des Fonctionnalités. Un utilisateur peut appartenir à plusieurs groupes.

Différents profils sont proposés pour caractériser les groupes et les utilisateurs. Ces profils contribuent à ce que le Modèle des Utilisateurs soit le support de capacités d'adaptation destinée à des groupes d'utilisateurs, mais également à des utilisateurs individuels. Une certaine forme d'adaptation à l'utilisateur est garantie en l'absence de profils spécifiquement définis pour lui, les profils de groupe se substituant aux profils individuels de leurs membres.

Les profils d'Accès Progressif exploitent une relation établie entre le MAP et le Modèle des Utilisateurs. Cette relation, modélisée par une association ternaire entre Stratification, Entité Masquable et Utilisateur (ou Groupe) permet de définir des stratification adaptées. En particulier, le Modèle des Fonctionnalités bénéficie de cette possibilité, les concepts qu'il met en œuvre étant considérés comme des EM. Une conséquence directe de cela réside dans la dérivation d'un Modèle de Structuration & Navigation également adapté à l'utilisateur. L'hypermédia proposé à celui-ci intègre alors l'ensemble des modalités d'accès progressif définies à son attention. A cette adaptabilité du système s'ajoute celle résultant de la définition, et par suite de l'application de chartes de composition et de présentation personnalisées maintenues par les profils de Présentation. Enfin, nous avons suggéré comme l'emploi de profils de Comportement est envisagé pour compléter le Modèle des Utilisateurs. Ceci passe dans un premier temps par la détection des actions des utilisateurs et le stockage des informations acquises dans ces profils. L'interprétation de ces profils permet ensuite agir en conséquence. Notamment, en ce qui concerne l'accès progressif, la réorganisation automatique des stratifications est envisagée, sur la base des opérations définies dans ce but dans le MAP.

6.5. Conclusion

Nous avons présenté quatre modèles à partir desquels la conception de SIW peut être engagée. Nous rappelons les objectifs majeurs de chacun d'eux.

Le *Modèle du Domaine* représente, en substance, le modèle de données sous-jacent au SIW.

Le *Modèle des Fonctionnalités* décrit les différentes tâches qu'un utilisateur peut mener à bien dans le SIW. Sa particularité réside dans l'organisation des tâches qu'il propose. D'une part, l'organisation repose sur différents niveaux portés par des concepts de granularités différentes : l'Espace Fonctionnel, le Rôle Fonctionnel et la Fonctionnalité. D'autre part ces concepts appliquent les principes du MAP et permettent ainsi de décrire un accès progressif à cette organisation.

Le *Modèle de l'Hypermédia* traite des aspects relatifs à la construction de l'hypermédia. Une représentation logique de sa structure et de ses liens de navigation est proposée par le sous-modèle de Structuration & Navigation qui rend compte de l'organisation fonctionnelle du modèle précédent. Le Modèle Hypermédia est complété par le second sous-modèle de Présentation, qui définit la composition des pages de l'hypermédia et leur apparence au moyen de chartes graphiques assurant leur cohésion.

Enfin, le *Modèle des Utilisateurs* repose sur le Modèle des Fonctionnalités pour identifier de groupe d'utilisateurs et d'utilisateurs individuels. Des profils caractérisent ces groupes et utilisateurs ; ils offrent ainsi un support pour une adaptabilité des accès progressifs et la présentation dans le SIW.

Dans le chapitre suivant, nous abordons l'utilisation de ces modèles dans la conception de SIW mettant en œuvre l'accès progressif.

7. CONCEPTION DE SIW A ACCÈS PROGRESSIF

Nous avons présenté les différents modèles que nous proposons pour la conception de SIW. Nous rappelons que ces modèles sont destinés à la conception de SIW offrant un accès progressif et adaptés à leurs données et à leurs fonctionnalités. L'objectif de ce chapitre est de montrer comment de tels modèles sont utilisés à cet effet.

Dans un premier temps, nous donnons une vue globale résumant notre proposition en termes de modèles. Nous nous appuyons pour cela sur les relations qu'entretiennent les modèles proposés.

Nous proposons ensuite une description de la démarche de conception préconisée reposant sur ces modèles (*cf.* section 7.2).

Dans la section 7.3, nous mettons en évidence le rôle sous-jacent mais néanmoins central que joue le MAP (présenté au chapitre 5) dans la conception de SIW telle qu'elle est ici envisagée. Ce rôle apparaît notamment à travers les relations que le MAP entretient avec les différents modèles de conception.

Nous discutons, dans la section 7.4, les particularités de notre proposition en la comparant aux méthodes étudiées au chapitre 3. Cette comparaison s'appuie sur la synthèse de l'état de l'art proposée au chapitre 4.

7.1. Relations entre les modèles de conception

Dans le chapitre 6, nous avons décrit séparément les différents modèles de notre proposition. Des relations entre ces modèles ont été suggérées, puisque certains de ces modèles s'appuient sur les spécifications de certains autres. Dans cette section, nous mettons en évidence de façon plus explicite leurs relations. Elles sont illustrées par la Figure 7.1.

Dans cette organisation, le Modèle du Domaine est utilisé comme ressource par le Modèle des Fonctionnalités. Plus précisément, la définition des Fonctionnalités comprend la définition de requêtes qui s'exécutent sur le Modèle du Domaine.

Nous avons montré que le Modèle des Fonctionnalités entretient une relation avec chacun des sous-modèles du Modèle de l'Hypermédia. Ainsi, le Modèle de Structuration & Navigation constitue une traduction du Modèle des Fonctionnalités dans des termes caractéristiques de l'approche hypermédia. Le Modèle de Structuration & Navigation fournit une représentation logique d'une ossature hiérarchique, composée de contextes et de nœuds dérivés des concepts fonctionnels (Espace et Rôle Fonctionnel, Fonctionnalités), ainsi que de liens qui mettent en évidence les possibilités de navigation inhérentes à l'accès progressif (e.g. masquage et dévoilement).

Le Modèle de Présentation traduit à son tour les spécifications du Modèle de Structuration & Navigation. Il s'agit cette fois de proposer une représentation conceptuelle des pages et des liens qui composent l'hypermédia. Les chartes de composition sont spécifiées à cet effet, sur la base de correspondances établies entre les concepts du Modèle de Structuration & Navigation et ceux employés par ces chartes (e.g. un contexte du Modèle S&N donne une page type dans une charte de composition). Une relation directe des chartes de composition est établie avec le Modèle de Fonctionnalités. Certains éléments (e.g. les composants) spécifient des requêtes portant sur le Modèle des Fonctionnalités. Ils permettent ainsi d'intégrer, dans l'hypermédia, les données de ce modèle (e.g. informations sur les Fonctionnalités telles que le nom), et, par transition, celles du Modèle du Domaine (e.g. résultat de requêtes utilisées par les Fonctionnalités). Les chartes graphiques complètent la spécification de la présentation de l'hypermédia. Leur description s'applique aux différents éléments des chartes de composition.

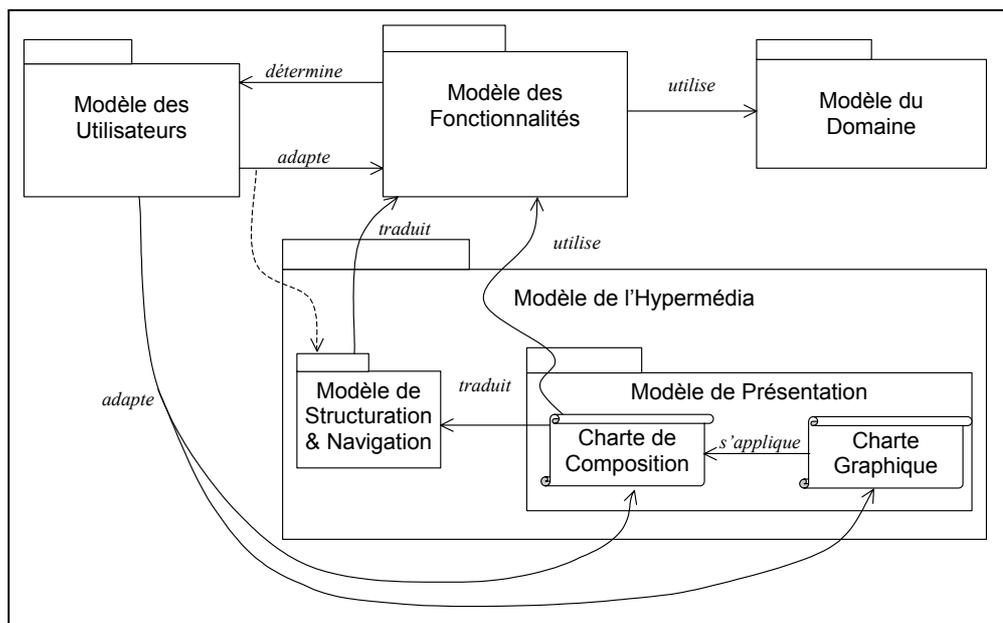


Figure 7.1 – Relations entre les différents modèles de conception

Le Modèle des Utilisateurs entretient de multiples relations avec les autres modèles de conception. D'un côté, il est déterminé par le Modèle des Fonctionnalités, la notion de groupe étant fondée sur une approche fonctionnelle qui l'associe au concept de Rôle Fonctionnel. Dans l'autre, une relation d'adaptation est identifiée. Le Modèle des Utilisateurs, à travers les profils d'Accès Progressif, permet d'adapter aux différents groupes et utilisateurs, les stratifications établies aux différents niveaux de granularité définis par le Modèle des Fonctionnalités. Il en résulte une adaptation implicite du Modèle de Structuration & Navigation traduit à partir du Modèle des Fonctionnalités.

Par ailleurs, le Modèle des Utilisateurs adapte le Modèle de Présentation, et ce à deux égards. Les profils de Préférences permettent de référencer des chartes de composition, d'une part, et des chartes graphiques d'autre part, spécifiquement élaborées pour des groupes ou utilisateurs donnés.

Dans la section suivante, nous donnons les directives pour une démarche de conception de SIW utilisant les modèles décrits.

7.2. Démarche de conception

A travers la mise en évidence des relations qui unissent les différents modèles de notre proposition, il est possible de déterminer quelques directives de conception. La démarche que nous préconisons est en effet en partie dictée par des relations de dépendance existant entre certains modèles. La Figure 7.2 illustre les étapes de cette démarche et leur enchaînement.

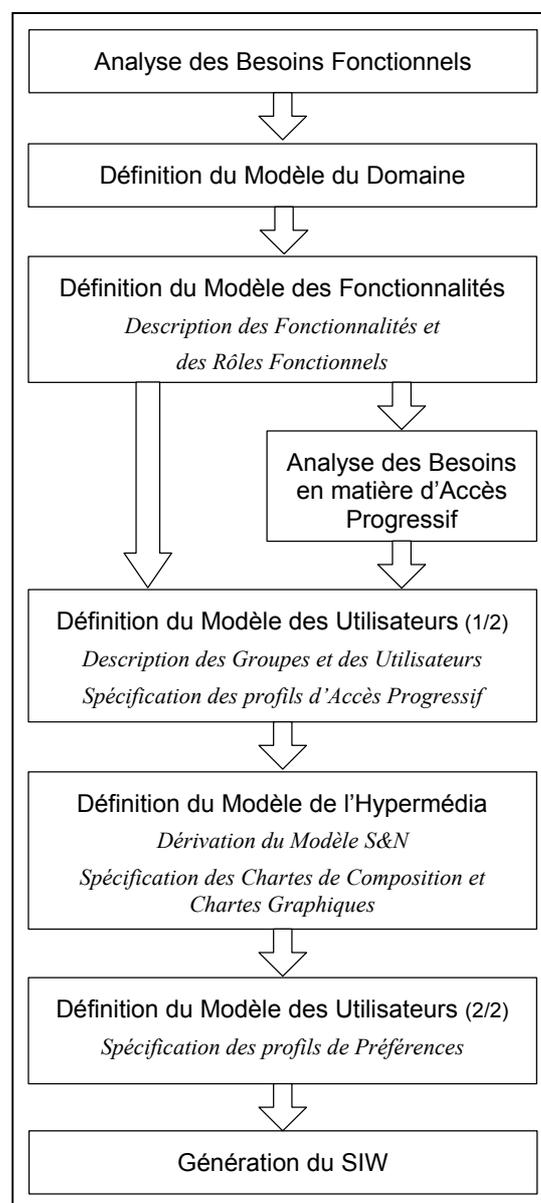


Figure 7.2 – Étapes de la démarche de conception préconisée

Nous identifions cinq étapes majeures de conception. Nous précisons qu'en amont, une phase d'analyse des besoins fonctionnels est une étape préliminaire indispensable à toute démarche de conception reposant sur nos modèles. Basée sur des cas d'utilisation, elle permettra de construire le Modèle du Domaine et le Modèle des Fonctionnalités. En aval, la génération du SIW termine le cycle de conception du SIW. Prise en charge par l'environnement KWIS, cette étape est présentée au chapitre 8.

Nous décrivons brièvement les activités du concepteur à chacune de ces étapes :

Etape 1. Définition du Modèle du Domaine _____

La définition du Modèle du Domaine consiste à créer les classes et associations qui traduisent les concepts manipulés par l'application.

Etape 2. Définition du Modèle des Fonctionnalités _____

Il s'agit ici de décrire toutes les Fonctionnalités du SIW en se basant sur les spécifications des cas d'utilisation décrites lors de l'analyse. La définition d'une fonctionnalité consiste à lui associer une identité (un nom) et la description de ce qu'elle permet. Cette définition comprend également la spécification des requêtes sur le Modèle du Domaine qu'elle utilise.

Ensuite, les Rôles Fonctionnels sont décrits : tout d'abord identifiés (nom, description), ils sont ensuite reliés aux fonctionnalités qui les composent.

Etape 3. Définition du Modèle des Utilisateurs (1/2) _____

A cette étape, les groupes sont identifiés. A chacun de ces groupes, le concepteur associe le Rôle Fonctionnel qui lui incombe. Les utilisateurs individuels connus du concepteur sont également créés. A chaque utilisateur est ensuite associé le (ou les) groupe(s) dont il est membre.

Dans un deuxième temps, les profils d'Accès Progressif sont définis pour les groupes et utilisateurs qui ont été créés. Cela consiste à définir les stratifications qu'on souhaite mettre en œuvre pour les Rôles Fonctionnels et Fonctionnalités des différents groupes. De même, les stratifications relatives aux Espaces Fonctionnels des utilisateurs connus sont définies, ainsi que celles que le concepteur souhaiterait redéfinir pour les Rôles Fonctionnels et Fonctionnalités. Cette redéfinition éventuelle peut être opérée pour tous les groupes dont l'utilisateur est membre.

Etape 4. Afin de faciliter la description des stratifications, nous proposons au concepteur de SIW des notations spécifiques permettant l'expression des besoins en matière d'Accès Progressif. Ces notations, basées sur les cas d'utilisation et les diagrammes de séquence UML, ainsi que la démarche associée pour les utiliser sont présentées en Annexe 3.

Etape 5. Définition du Modèle de l'Hypermédia _____

La définition du Modèle de l'Hypermédia débute par la dérivation du Modèle de Structuration & Navigation. Cette dérivation est effectuée pour chacun des groupes et permet de visualiser l'ossature de l'hypermédia qui lui sera associée. Le concepteur augmente le modèle produit si nécessaire (ajout de contexte, de liens, etc.).

Chaque modèle obtenu est traduit en vue d'obtenir le Modèle de Présentation de l'hypermédia. Le concepteur associe les pages types de la charte de composition par défaut présentées dans la section 6.3.2.1, sous réserve qu'il n'ait pas augmenté le Modèle S&N. Le cas échéant, il doit engager la spécification d'une nouvelle charte de

composition. Il définit alors la page type à associer à chaque contexte du Modèle S&N et le contenu de cette page. Une fois définie, une telle charte est stockée en vue de sa réutilisation pour traduire le Modèle S&N d'un autre groupe.

La définition du Modèle de l'Hypermédia se poursuit par la définition de l'apparence de chaque page type. Le concepteur peut soit leur associer la charte graphique par défaut, soit spécifier puis associer une nouvelle charte graphique.

La définition de nouvelles chartes de composition et/ou chartes graphiques peut également être destinée à des utilisateurs individuels.

Etape 6. Définition du Modèle des Utilisateurs (2/2)

Cette étape consiste simplement à mettre à jour les profils de Préférences. Le concepteur référence, dans chaque profil de groupe et, le cas échéant, dans chaque profil d'utilisateur, la charte de composition et la charte graphique adéquates.

A l'issue de ces étapes, la génération du SIW peut être engagée. Dans le chapitre 8, nous illustrons les actions effectuées par le concepteur à chacune de ces étapes (*cf.* section 8.2). C'est en effet sur celles-ci que repose le processus de création d'un SIW dans KIWIS, l'environnement de conception que nous avons implémenté.

Dans la section suivante, nous mettons en évidence le rôle central joué par le MAP dans cette approche.

7.3. Rôle central du MAP dans la conception de SIW

Les modèles ont été définis avec l'objectif majeur de répondre aux besoins d'une conception guidée par la mise en œuvre des modalités d'un accès progressif à l'information et aux fonctionnalités. Pour atteindre cet objectif, nous avons largement exploité les possibilités offertes par le MAP qui constitue, comme annoncé au chapitre 5, le modèle de référence de notre approche.

Nous avons souhaité explorer toutes les dimensions qui, dans un SIW, peuvent être la cible des concepts et des mécanismes de l'accès progressif. Ceci est mis en évidence par la Figure 7.3 qui décrit les relations entretenues par les différents modèles que nous proposons avec le MAP.

Nous avons montré à l'occasion du chapitre 5 comment un modèle de données applique les principes du MAP, dès lors que l'on établit une correspondance entre ses concepts (structures de données) et les concepts mis en œuvre par le MAP. Le Modèle du Domaine, proposé pour décrire le domaine d'application du SIW, ne déroge pas à ce principe. Bien que cela n'ait pas été redéfini explicitement dans ce chapitre, ce modèle et ses constituants (classes, associations et classes-associations) exploitent le MAP de façon similaire. Concrètement, l'application du MAP au Modèle du Domaine apparaît dans l'utilisation qui est faite du Modèle du Domaine par le Modèle des Fonctionnalités. Le concept de Requête modélisé dans le Modèle des Fonctionnalités représente les interrogations qui portent sur le Modèle du Domaine. Le Résultat de telles Requêtes, en tant qu'Entité Masquable, exploite ainsi les concepts du Modèle du Domaine interprétés également comme des concepts du MAP.

Le Modèle des Fonctionnalités applique également les concepts et principes du MAP. L'implication des Résultats de Requêtes utilisées par les Fonctionnalités dans ce sens vient d'être rappelée. Ce modèle est cependant en relation avec le MAP à deux autres niveaux, constituant ainsi un exemple d'application récursive du MAP. Nous avons ainsi montré comment le concept d'Espace Fonctionnel, considéré comme une EM, peut être stratifié en différentes REM donnant un Accès Progressif à plus ou moins d'éléments : les Rôles Fonctionnels. A son tour, chacun des

Rôles Fonctionnels est vu comme une EM, dont la stratification offre des possibilités d'Accès Progressif aux Fonctionnalités, les éléments constitutifs d'un tel rôle dans le modèle. Enfin, toute Fonctionnalité applique elle-même les principes de l'accès progressif, la charge de mettre en œuvre cet accès étant déléguée aux Résultats de Requête comme rappelé ci-dessus.

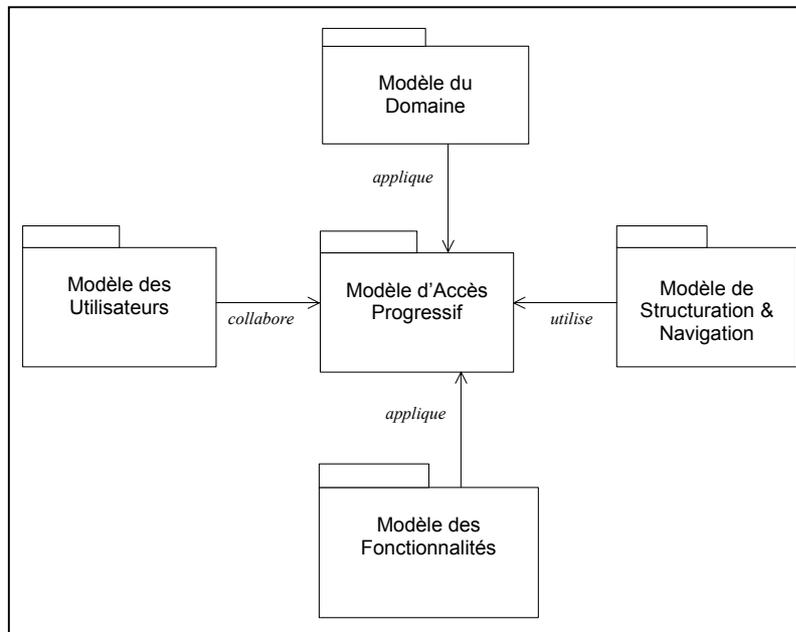


Figure 7.3 – Relations entre les modèles de conception et le MAP

Le Modèle de l'Hypermédia entretient une relation un peu différente avec le MAP. Un de ses sous-modèles, le Modèle de Structuration & Navigation, utilise les concepts décrits par le MAP. C'est en effet en référence aux spécifications du MAP que les concepts proposés pour modéliser l'ossature de l'hypermédia sont définis. Il en résulte une structure et des possibilités de navigation spécifiques qui constituent le premier pas vers une traduction dans les termes de l'approche hypermédia de l'approche par accès progressif.

Enfin, nous définissons la relation qu'entretiennent le Modèle des Utilisateurs et le MAP comme une relation de collaboration. Cette collaboration est à double sens. D'une part, elle permet au Modèle des Utilisateurs d'enrichir ses capacités descriptives en intégrant les concepts et principes du MAP. Il devient possible de décrire un utilisateur en tenant compte des caractéristiques qu'il attend du système quant à la quantité d'informations délivrées. L'accès progressif permet, par ailleurs, de ne pas limiter la diffusion de l'information, mais de l'échelonner. D'autre part, la collaboration dans le sens inverse permet au MAP de prendre une dimension supplémentaire : celle de l'adaptation. Le Modèle des Utilisateurs rend toute stratification personnalisable, permettant ainsi de multiplier le pouvoir d'expression du MAP.

Dans la section suivante, notre proposition est comparée aux méthodes de conception d'applications basées sur le Web dont nous avons décrit les principes dans le chapitre 3.

7.4. Comparaison de la proposition avec l'état de l'art

Dans cette section, nous reprenons les quatre rubriques (i.e. modélisation de l'utilisateur, dimensions adaptées, modélisation et mise en œuvre de l'adaptation, et, capacités d'adaptation du SIW conçu) qui nous ont permis d'évaluer les méthodes décrites dans l'état de l'art, afin de

comparer notre proposition à ces méthodes. Nous invitons le lecteur à consulter le chapitre 4 pour un rappel de la signification des critères retenus. Chacun des tableaux présentés au chapitre 4 est repris dans cette section. Nous ajoutons à droite de ces tableaux une colonne dédiée à notre proposition (baptisée KIWIS, en référence à l'environnement de conception que nous présentons au chapitre 8).

Une cinquième rubrique complète cette analyse comparative. Elle est consacrée

7.4.1 Modélisation de l'utilisateur

La première rubrique que nous traitons est celle de la modélisation de l'utilisateur en vue de l'adaptation (*cf.* Tableau 7.1).

Les caractéristiques que nous avons évaluées au chapitre 4, sont toutes abordées par notre proposition, à des degrés divers. Ainsi, le Modèle des Utilisateurs proposé tient effectivement compte des groupes et des utilisateurs individuels. Ce modèle, parce qu'extensible par ajout de profils adéquats peut tout fait supporter des caractéristiques spécifiques au domaine ou autres, ou encore des conditions d'exploitation. Il revient alors au concepteur de définir ces profils et les modalités de leur prise en compte, comme cela est le cas dans d'autres méthodes. Notre objectif essentiel, clairement axé sur l'accès progressif, a nécessité avant toute autre chose de donner les fondements de cette approche. Néanmoins, au-delà de ces fondements, diverses sources d'informations peuvent contribuer à l'approche par Accès Progressif. La prise en compte de caractéristiques telles qu'un niveau de compétences peut, par exemple, être un atout. Nous pensons notamment au rôle que peut jouer une telle connaissance dans la définition des stratifications, et, en particulier, dans le domaine de l'enseignement à distance. Par exemple, un niveau de connaissance peut constituer une contrainte bloquante des mécanismes de dévoilement : tant que ce niveau n'est pas atteint, l'utilisateur ne peut accéder aux niveaux de détail supérieur.

Le Modèle des Utilisateurs permet à travers les profils d'Accès Progressif et de Préférences, de gérer les attentes et besoins spécifiques des utilisateurs en termes :

- de contenu, puisque l'utilisateur peut accéder en priorité à celui qui est le plus pertinent selon lui, et ce, que ce contenu soit fonctionnel ou informationnel ;
- de structure de l'hypermédia résultant des stratifications définies sur le Modèle des Fonctionnalités et répercutées automatiquement sur les structures de pages et de liens de navigation ;
- de présentation à travers des chartes de composition et graphiques spécifiques référencées dans les profils de préférences.

L'acquisition des informations relatives aux utilisateurs est explicite. Le concepteur peut définir lui-même ces informations, de même que l'utilisateur en a la possibilité à travers l'interface du SIW généré. Dans notre approche, l'utilisateur a en effet un droit sur les informations le concernant et peut, à tout moment, accéder à ses profils pour en modifier les valeurs (redéfinition des stratifications notamment). Nous mentionnons un mode d'acquisition implicite des informations. Bien que non menée à terme à ce jour, la proposition basée sur les profils comportementaux va en effet dans ce sens.

Tableau 7.1 – Évaluation de notre proposition concernant la modélisation des utilisateurs.

| NOM DU CRITÈRE | | WSDM | HERA | AWIS-M | UWE | WEBML | KIWIS |
|----------------------------------|--|-------------|-----------------------|-----------------------|----------------|---|------------------------------|
| Existence | | oui | oui | oui | oui | En tant que partie du modèle conceptuel | oui |
| Granularité | | groupe | individu | groupe (individu) | individu | groupe / individu | Groupe / individu |
| Caractéristiques | Connaissances spécifiques au Domaine d'Application | (oui) | non | oui, à définir | oui, à définir | oui, à définir | Oui, à définir |
| | Connaissances Autres | non | non | oui, à définir | oui, à définir | non | Oui, à définir |
| | Conditions d'Exploitation | non | oui | non | non | non | Oui, à définir |
| | Fonctionnalités | oui | non | oui | non | non | oui |
| | Accès Progressif | non | non | non | non | non | oui |
| Préférences | Contenu | non | non | non | non | non | oui |
| | Structure de l'hypermédia | non | non | non | non | non | oui |
| | Présentation | (oui) | oui | non | oui, à définir | Non | oui |
| Acquisition | | explicite | explicite / implicite | explicite / implicite | implicite | implicite | Explicite (implicite) |
| Droits sur le modèle Utilisateur | | non précisé | non précisé | non précisé | non précisé | non précisé | oui |

7.4.2 Dimensions adaptées

La seconde dimension d'évaluation est relative aux dimensions adaptées. Le Tableau 7.2 montre que nous traitons, à travers les différents modèles, chacun des critères retenus. Nous précisons que notre approche aborde tous ces aspects avec l'objectif spécifique de la mise en œuvre de l'Accès Progressif.

L'Accès Progressif constitue en lui-même une dimension d'adaptation. D'une part, il l'est dans la réponse qu'il apporte aux besoins d'accès à l'information. Il permet aux utilisateurs de gérer leur propre espace d'information en distinguant ce à quoi ils souhaitent accéder de façon immédiate, et ce qui peut être atteint si besoin, de façon progressive. D'autre part, l'accès progressif est proposé dans une logique d'adaptation puisque chaque stratification peut être personnelle.

Tableau 7.2 – Évaluation de notre proposition concernant les dimensions adaptées

| CRITÈRE | RMM/HERA | OOHDM | WSDM | AWIS-M | UWE | WEBML | KIWIS |
|-----------------|---|--|--|---|--|--|--|
| Contenu | (RMM) <i>m-slice</i> | <i>Navigational contexts, Navigational classes et classes Incontext</i> | <i>User Object Model et Perspective Object Model</i> | <i>Élément Atomique de Navigation, Nœud de Navigation</i> | <i>Classes Variantes</i> | <i>Modèle de dérivation</i> | Profil d'Accès Progressif |
| Structure | | | | <i>Adaptable</i> | propriétés exprimées dans le modèle de structure de Navigation | une vue du site pour chaque groupe | Modèle de Structuration & Navigation |
| Navigation | | | <i>Navigation track</i> | <i>Directive de Navigation</i> | dépend des spécifications du modèle utilisateur | une vue du site pour chaque groupe et dispositif d'accès | |
| Présentation | (Hera) <i>m-slice, adaptability et adaptativity conditions et update rules</i> | | | | | | Profil de Préférences (Charte de Composition Et Charte Graphique spécifiques) |
| Fonctionnalités | | <i>Navigational contexts, Navigational classes et classes Incontext</i> définis en fonction des tâches | <i>User Object Model et Perspective Object Model</i> définis en fonction des besoins | <i>Unité Sémantique de Navigation</i> définie en fonction des objectifs | | Une vue du site pour chaque groupe | Modèle des Fonctionnalités Et Profils d'Accès Progressif |

7.4.3 Modélisation et mise en œuvre de l'adaptation

La troisième rubrique concerne la modélisation et la mise en œuvre de l'adaptation. La particularité de notre approche réside dans la proposition d'une étape spécifique d'analyse des besoins en matière d'accès progressif. Les autres approches n'incluant pas l'accès progressif, il est logique que notre proposition soit la seule à présenter cette particularité.

On note que notre approche rejoint en partie la proposition AWIS-M [Gnah01] en ce qui concerne la mise en œuvre de l'adaptation. La construction de l'hypermédia est en effet dynamique, par opposition aux autres propositions qui reposent exclusivement sur le choix de la version à présenter parmi plusieurs.

Notre approche se distingue néanmoins des autres propositions dans la mesure où, pour mettre en œuvre l'adaptabilité, nous n'avons pas de règles spécifiques à définir, ni de modèle

dédié. Ce sont en effet les relations qu'entretiennent les modèles conceptuels qui portent les principes de l'adaptabilité. La tâche du concepteur est donc facilitée, puisque le seul fait d'avoir recours aux modèles proposés garantit la production d'un SIW capable d'adaptation.

Tableau 7.3 – Évaluation de notre proposition concernant la modélisation et la mise en œuvre de l'adaptation

| CRITÈRE | OOHDM | WSDM | WEBML | HERA | AWIS-M | UWE | KIWIS |
|---|------------------|----------------------------------|------------------|---------------------|--------------------------------------|----------------------------------|--|
| Analyse des besoins en matière d'adaptation | | Analyse des besoins fonctionnels | | | Analyse des besoins fonctionnels | Analyse des besoins fonctionnels | Analyse des besoins fonctionnels et analyse des besoins en matière d'accès progressif |
| Modèle conceptuel de l'adaptation | | | règles | Conditions / règles | Modèle mathématique de l'hyperespace | Modèle d'adaptation | Adaptation inhérente aux relations entre les modèles |
| Technique de mise en œuvre de l'adaptation | Choix de version | Choix de version | Choix de version | Conditions / règles | Construction dynamique | Moteur d'adaptation | Construction dynamique |

7.4.4 Capacités d'adaptation du SIW

Les capacités d'adaptation des SIW conçus en suivant notre approche sont essentiellement à ce jour l'adaptation minimale et l'adaptabilité (telles que définies dans la section 4.4).

Tableau 7.4 – Évaluation des capacités d'adaptation des SIW conçus avec notre proposition.

| CRITÈRE | RMM | OOHDM | WSDM | HERA | WEBML | AWIS-M | UWE | KIWIS |
|---------------------|-----|-------|------|-------|-------|--------|-------|------------|
| Adaptation minimale | + | + | + | + | + | + | + | + |
| Adaptabilité | - | - | - | + / - | + / - | + / - | + / - | + |
| Adaptativité | - | - | - | + / - | + / - | + / - | + | (+) |

La première apparaît dans le fait, consécutivement à la connexion de l'utilisateur, de choisir 'la' meilleure version de pages Web à présenter. Dans certaines approches, il s'agit de la retrouver parmi un ensemble de versions stockées ; dans notre proposition, le processus est plus dynamique : il s'agit de retrouver les spécifications du groupe de l'utilisateur et, à partir de celles-ci, de générer les pages nécessaires. Il suffit donc que l'utilisateur soit connu en tant que membre d'un groupe pour bénéficier d'une adaptation minimale.

L'adaptabilité est assurée par la possibilité offerte à l'utilisateur de contrôler le processus d'adaptation. Deux types de contrôle sont, par ailleurs, possibles. Le premier est inhérent à l'accès progressif : l'utilisateur gère à tout moment la quantité d'information qui est présentée en ayant recours aux mécanismes de masquage et de dévoilement. Le second contrôle offert à l'utilisateur réside dans la gestion de sa propre représentation au sein du système. En accédant à ses profils (d'accès progressif et de préférence) et en modifiant les spécifications de ceux-ci, il influence le processus d'adaptation. Le système est donc capable de réagir à des demandes explicites d'adaptation de la part de l'utilisateur.

Enfin, l'adaptativité est définie comme l'aptitude du système à observer le comportement de l'utilisateur et à procéder en conséquence à des adaptations. Les spécifications que nous avons données, au chapitre 6, pour les profils comportementaux vont nettement dans ce sens.

7.4.5 Comparaison des Concepts d'Accès Progressif et d'autres concepts pour la représentation des données

Les *vues* (définies en bases de données – et largement appliquées depuis aux SGBD orientés-objets [Abit90] ou à XML [Abit99] – sont utilisées pour offrir à l'utilisateur une représentation pertinente des informations en fonction de ses besoins. Les REM peuvent être considérées comme des listes ordonnées par inclusion ensembliste de vues opérant la même projection sur les données. A notre connaissance, il n'existe pas de travaux proposant une organisation de plusieurs vues en une structure similaire à celle décrite par les stratifications ici et favorisant de ce fait l'accès progressif à l'information.

Les *objets morcelés* [Bard98] et les REM partagent l'idée d'un découpage favorisant l'accès à différentes versions d'un même concept. Une différence essentielle entre ces approches et la nôtre est cependant à noter. Les objets morcelés, issus de la théorie des prototypes, sont définis sur les objets alors que les stratifications concernent une définition structurelle de plus haut niveau d'abstraction qui vise les classes, mais aussi les associations et le modèle de données lui-même. La stratification d'une classe en différentes REM peut toutefois s'inspirer de la sémantique véhiculée par les morceaux d'un objet morcelé. Les objets morcelés offrent différents aspects ou points de vue d'une même entité. Le morceau racine d'une hiérarchie de morceaux correspond au point de vue le plus général de l'entité représentée par l'objet morcelé. Les autres morceaux de la hiérarchie dénotent des points de vue plus spécifiques. La création de REM peut être calquée sur le découpage en points de vues de l'entité. Il serait alors nécessaire d'ordonner les morceaux pour traduire l'accès progressif à l'information inhérent aux REM.

La notion de *contexte* [Mots99] ou celle d'Hyperclasse [Turk01][Turk02] peuvent elles-aussi être envisagées comme base pour la stratification, appliquée non plus aux classes comme pourrait l'être la notion d'objet morcelé, mais au schéma. Globalement ces deux notions permettent de définir des sous-schémas particuliers du schéma global. Notre approche peut être vue comme la spécification de contextes ou d'Hyperclasse à *taille variable*: l'espace informationnel de l'utilisateur est stratifié pour permettre son élargissement ou sa réduction. Ce que nous appelons *contexte à taille variable* peut être rapproché de la proposition de [Theo99] où la notion de contexte est enrichie par le concept de *référence* qui sert à masquer en partie l'information. Cependant, cette approche n'est pas reliée à une démarche d'adaptation à l'utilisateur ce qui, selon nous, est une nécessité pour en exploiter totalement les possibilités.

Dans WebML [Ceri00], l'unité de contenu appelée *Data Unit* (cf. Annexe 1) permet de définir plusieurs représentations d'une même information, par exemple, une version textuelle et multimédia. Les auteurs envisagent également la possibilité d'utiliser les *Data Units* pour décrire une version longue et une version courte d'une même information. Néanmoins, nous notons plusieurs différences. Premièrement, les différentes versions ne sont pas construites en vue de

proposer un accès progressif. Il s'agit, pour les auteurs, de présenter l'une *ou* l'autre de ces versions. Aucune notion comparable aux mécanismes de masquage et de dévoilement n'est mise en œuvre pour passer de l'une à l'autre, et naviguer à des niveaux de détails différents. Deuxièmement, la description de ces représentations multiples est une possibilité offerte au concepteur, mais ne constitue pas une ligne directrice de la conception comme c'est le cas avec l'Accès Progressif. Enfin, la possibilité de définir différentes versions est offerte uniquement sur les entités du modèle structurel de WebML. Ces entités correspondent globalement, dans notre approche, aux classes du Modèle du Domaine. L'Accès Progressif s'applique donc à un plus large panel de concepts qui sont, d'une part, relatifs aux données (*i.e.* classes, mais aussi associations, et résultats de requêtes), et d'autre part, portant sur des aspects fonctionnels (*i.e.* Espace et Rôle Fonctionnels).

7.5. Conclusion sur la conception de SIW à Accès Progressif

Nous avons présenté les différents modèles que nous proposons pour la conception de SIW. Nous rappelons que ces modèles sont destinés à la conception de SIW offrant un accès progressif et adapté à leurs données et à leurs fonctionnalités. L'objectif de ce chapitre était de montrer comment de tels modèles sont utilisés à cet effet.

Dans un premier temps, nous avons donné une vue globale résumant notre proposition en termes de modèles. Nous nous sommes appuyé pour cela sur les relations qu'entretiennent les modèles proposés.

Nous avons ensuite proposé une description de la démarche de conception préconisée reposant sur ces modèles (*cf.* section 7.2).

Dans la section 7.3, nous avons mis en évidence le rôle sous-jacent mais néanmoins central que joue le MAP (présenté au chapitre 5) dans la conception de SIW telle qu'elle est ici envisagée. Ce rôle apparaît notamment à travers les relations que le MAP entretient avec les différents modèles de conception.

Nous avons enfin exposé, dans la section 7.4, les particularités de notre proposition en la comparant aux méthodes étudiées au chapitre 3. Cette comparaison s'appuie sur la synthèse de l'état de l'art proposée au chapitre 4. Quelques comparaisons avec des approches visant à offrir des représentations particulières sur les données d'un système ont également été présentées.

RÉALISATION

8. L'ENVIRONNEMENT KIWIS

Ce chapitre présente l'environnement KIWIS développé afin de valider et de rendre opérationnelle l'approche que nous avons présentée. Cet environnement répond à un double objectif. D'une part, il offre un *environnement pour la conception* de SIW capables de supporter l'accès progressif. En ce sens, KIWIS est destiné aux concepteurs de SIW. D'autre part, il offre un *environnement pour l'utilisation* des systèmes qu'il permet de concevoir et de générer. Il constitue donc également une plate-forme pour les utilisateurs finaux des SIW.

Dans la première section, nous décrivons l'architecture de cet environnement. D'un point de vue logiciel, comme d'un point de vue fonctionnel, l'architecture est en très grande partie commune aux deux activités (conception et utilisation) supportées par KIWIS. Cet environnement a été conçu comme un SIW dont le domaine métier relève de la conception et de la mise en œuvre de SIW. Une description de l'analyse et de la conception menée préalablement au développement de KIWIS est donnée dans [Eyme02].

Nous présentons ensuite l'environnement dédié au concepteur dans la section 8.2. A cette occasion, nous mettons en évidence le processus de création d'un SIW à Accès Progressif. Le processus décrit repose sur la démarche de conception que nous avons présentée au chapitre 6.

Dans la section 8.3, nous décrivons l'environnement dont dispose l'utilisateur lorsqu'il se connecte à un SIW généré avec KIWIS. Une attention particulière sera portée à la mise en relief des possibilités offertes à l'utilisateur en terme d'accès progressif et de gestion des adaptations le concernant.

8.1. Architecture du prototype

8.1.1 Architecture Logicielle

L'architecture logicielle retenue est présentée dans la Figure 8.1. Nous la décrivons dans les sections suivantes en abordant, dans un premier temps, les outils utilisés. Dans un second temps, le prototype KIWIS est décrit. Enfin, l'organisation du stockage des données est brièvement présentée.

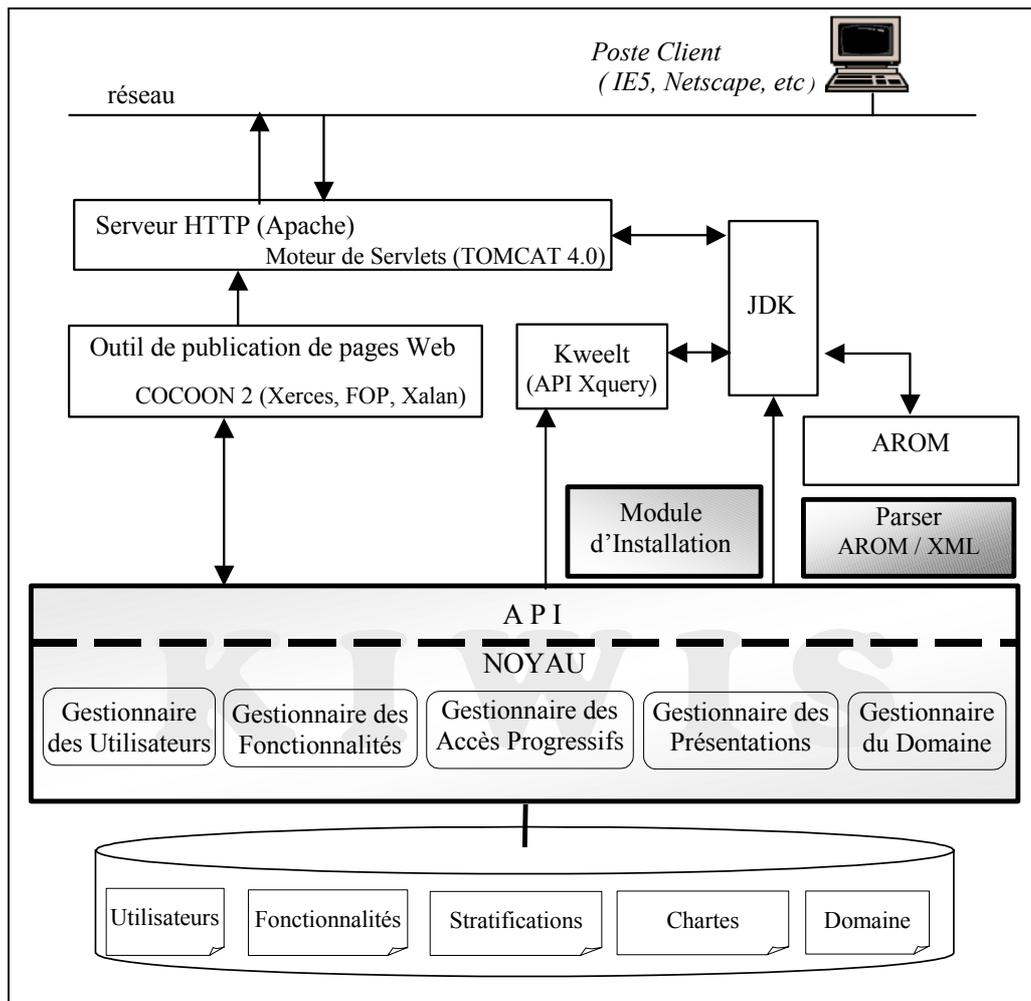


Figure 8.1 – Architecture de l'environnement KIWIS.

8.1.1.1 Intégration d'outils existants

Nous décrivons rapidement les outils existants utilisés dans cette architecture :

- le serveur WEB Apache 1.3 associé au moteur de Servlets Tomcat 4.0 qui est utilisé comme serveur d'applications. La communication client / serveur repose sur le protocole http. Le poste client doit être doté d'un navigateur soumis à la seule contrainte d'accepter l'utilisation de cookies et de JavaScript.
- l'outil de Publication Web Cocoon 2 qui permet la publication de contenu XML sur le Web. Pour fonctionner, Cocoon (une servlet Java) doit être installé au sein d'un moteur de servlets (ici Tomcat). Lors de l'installation de Cocoon, le parseur XML Xerces, le transformateur XSLT Xalan et le formateur XSL:FO FOP sont installés dans l'environnement Tomcat. Le modèle de publication proposé par Cocoon est décrit en Annexe 4.
- le module d'interrogation de fichiers XML Kweelt. Il s'agit d'une implémentation Java de la proposition du langage de requêtes pour XML appelée Quilt. La version 2 de cet outil intègre la spécification W3C XQuery.
- l'environnement de développement Java JDK 1.3.

- le système de représentation de connaissances par objets (SRCO) AROM³⁷ [Page00] que nous utilisons comme outil de modélisation et, en partie, pour le stockage de données. L'IME (Interactive Modeling Environment) d'AROM permet la représentation graphique de modèles de représentation de connaissances au moyen de notations proches d'UML. L'instanciation de la base de connaissances – définition des objets des classes et des tuples d'associations – se fait sous le contrôle (vérifications de types, de cohérence, et inférences de valeurs de variables) d'AROM. L'API Java de AROM permet d'interfacer simplement les bases de connaissances créées avec les différents modules de KIWIS chargés de les exploiter, notamment parce que ces modules sont eux-mêmes implémentés dans des technologies voisines (Java et Servlets).

8.1.1.2 Le prototype KIWIS

Le prototype KIWIS complète cette architecture logicielle. Il offre une API Java permettant la communication avec les différents outils de l'environnement (notamment Cocoon, Kweelt, le JDK, les bases de données, etc.). Cinq modules, appelés *gestionnaires*, constituent le noyau du système. Ce sont des paquetages de classes Java chargés de différentes fonctions dans le système. Nous les décrivons succinctement, sans détailler les relations qu'ils entretiennent, ceci faisant l'objet de la section 8.1.2.

1. Le *Gestionnaire du Domaine* est le module chargé de gérer les données du Modèle du Domaine (au format XML). Interlocuteur privilégié de AROM, il communique avec ce SRCO et en traduit les connaissances en utilisant un parser XML/AROM.
2. Le *Gestionnaire des Utilisateurs* permet notamment la gestion du Modèle des Utilisateurs. Il est impliqué dans la création des utilisateurs de KIWIS, la mise à jour des informations les concernant, leur suppression, etc. L'authentification des concepteurs et utilisateurs lors de leur connexion lui incombe également.
3. Le *Gestionnaire des Fonctionnalités* assure la gestion (création, modification suppression) des espaces fonctionnels, rôles fonctionnels et fonctionnalités.
4. Le *Gestionnaire des Accès Progressifs* est responsable des informations relatives aux stratifications (création, modification, suppression) référencées par les profils d'Accès Progressif.
5. Le *Gestionnaire des Présentations* maintient la bibliothèque des chartes de composition et graphiques.

En plus de l'API et du noyau de KIWIS, l'architecture de l'environnement comprend deux composants développés au sein de l'équipe SIGMA :

- un parser AROM/XML utilisé principalement pour traduire les bases de connaissances AROM au format XML. Ce langage étant à la base du processus de publication sur le Web, toutes les informations décrites et instanciées avec AROM doivent être retranscrites en XML.
- un module d'Installation qui contient l'ensemble des composants logiciels nécessaires à l'installation de l'environnement KIWIS sur les postes clients (aussi bien en version concepteur qu'utilisateur). L'utilisation de ce module permet d'installer l'ensemble des outils mentionnés. Une fois ces composants installés sur la machine *hostname*, le concepteur

³⁷ AROM est l'acronyme de « Allier Relations et Objets pour Modéliser »

peut définir des SIW qui seront accessibles depuis les différents postes clients à l'url `http://hostname:8080/kiwis`.

8.1.1.3 Stockage des données

Des sources multiples d'informations sont utilisées par un SIW conçu et généré avec KIWIS. L'organisation des bases de données est la suivante.

- Base des Utilisateurs : Elle comporte les informations relatives aux groupes et utilisateurs individuels. Leurs profils sont gérés au sein d'une même base, à l'exception du profil d'Accès Progressif qui constitue le cœur de la base Stratification. Il existe un schéma XSD décrivant les groupes et un schéma XSD décrivant les utilisateurs, liés par une relation d'importation du second par le premier. Plusieurs fichiers XML de description d'instances se conforment à ces descriptions ; il existe un fichier par instance de groupe et un fichier par instance d'utilisateurs. Un fichier XML décrivant un groupe fait référence aux fichiers XML qui décrivent les utilisateurs membres de ce groupe.
- Base des Fonctionnalités : Les Fonctionnalités stockées dans cette base sont considérées dans leurs versions génériques : les requêtes qu'elles utilisent y sont décrites, mais leurs résultats ne sont pas stratifiés. Les stratifications de résultats font partie intégrante du profil d'Accès Progressif des utilisateurs, et sont donc stockées dans la base de Stratification.
- Base des Stratifications : cette base correspond au profil d'Accès Progressif modélisé dans le modèle des Utilisateurs. Un fichier XSD décrit le schéma d'une stratification telle que nous l'avons présentée dans le Modèle d'Accès Progressif. Des fichiers XSD importent ce schéma et l'étendent pour donner une représentation de tous les types d'informations considérées comme des EM dans le cadre de l'application. Des fichiers XML, conformes aux fichiers XSD précédents,instancient pour chaque groupe et chaque utilisateur, les stratifications possibles.
- Base des Présentations : Les chartes de composition et graphiques sont stockées dans cette base. Il s'agit de feuilles de style XSL. Les profils de Préférences maintenus dans la base des Utilisateurs font référence aux données de la base des Présentations.
- Base du Domaine : Cette base stocke le Modèle du Domaine ainsi que les instances des classes et tuples des associations qu'il décrit. Des fichiers XSD et XML sont utilisés à ces fins. En fonction de la complexité du modèle, et éventuellement du caractère réparti de celui-ci, plusieurs fichiers (liés par des relations d'importation, de références, etc.) sont utilisés.

8.1.2 Architectures Fonctionnelles

L'objectif de cette section est d'illustrer la façon dont les gestionnaires (*i.e.* le noyau de KIWIS) interagissent, entre eux et avec les autres composants de l'architecture logicielle, pour traiter les demandes émanant des utilisateurs de KIWIS. Nous distinguons l'architecture fonctionnelle de conception, de l'architecture fonctionnelle d'utilisation.

8.1.2.1 Architecture fonctionnelle de conception

La Figure 8.2 illustre les relations qu'entretiennent les différents gestionnaires lorsque KIWIS est utilisé pour concevoir des SIW. Nous décrivons brièvement l'architecture fonctionnelle présentée, celle-ci étant relativement proche de la description que nous avons donnée des relations entre modèles de conception (*cf.* section 7.1). À l'exception du Gestionnaire des Stratifications, chaque gestionnaire est en effet associé à un des quatre modèles de notre

proposition. Le Gestionnaire des Stratifications repose principalement sur les spécifications du MAP. La figure indique également pour chaque gestionnaire la base de données dont il assure la gestion.

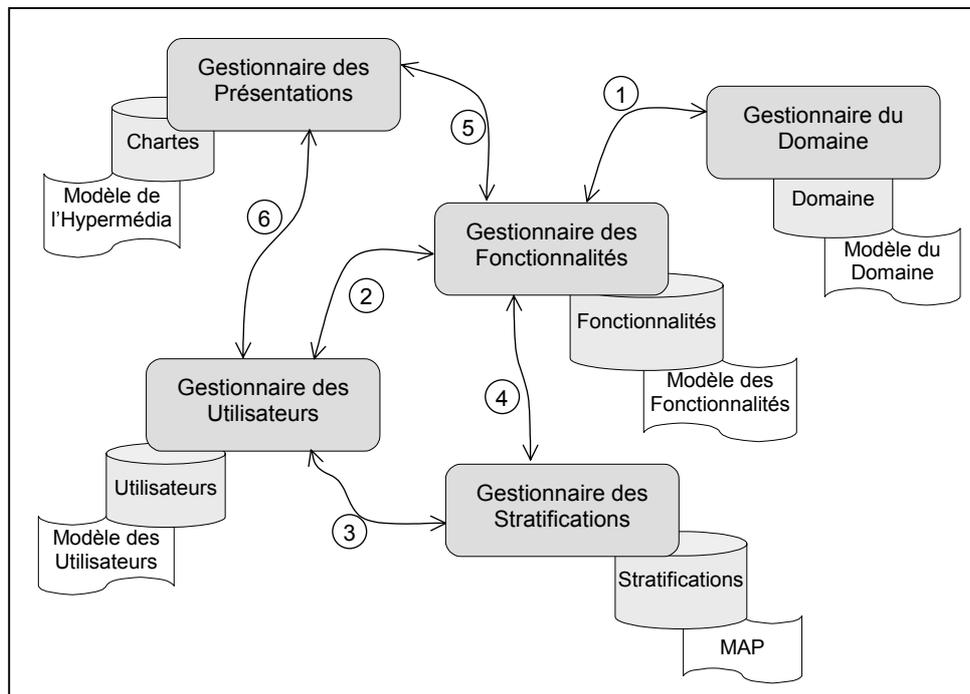


Figure 8.2 – Architecture fonctionnelle de KIWIS : Relations des Gestionnaires lors de la Conception

Nous considérons ici que le Modèle du Domaine a été éventuellement créé en utilisant l'IME de AROM, et que sa traduction au format XML a été effectuée à la demande du Gestionnaire du Domaine, en utilisant le parser AROM/XML.

Le concepteur a ensuite recours au Gestionnaire des Fonctionnalités pour instancier le modèle associé (création de données dans la base des Fonctionnalités). Des liens sont établis vers la base du Domaine : ils correspondent aux spécifications de requêtes utilisées par les Fonctionnalités. La relation ① indique la communication entre le Gestionnaire des Fonctionnalités et le Gestionnaire du Domaine dans ce but.

La création des Fonctionnalités permet ensuite de déterminer les données (Groupes) à créer dans la base des Utilisateurs. Le Gestionnaire des Fonctionnalités communique au Gestionnaire des Utilisateurs les informations dont il dispose à ce propos ②. Le concepteur définit donc le Modèle des Utilisateurs en utilisant le Gestionnaire des Utilisateurs.

Il poursuit la conception en définissant les profils d'Accès Progressif nécessaires. Le Gestionnaire des Stratifications permet la création dans la base du même nom des différentes stratifications. Dans cette optique, plusieurs protocoles de communication sont établis entre les trois gestionnaires impliqués :

② : choix des données à stratifier dans la base des fonctionnalités pour les différents utilisateurs

③ : établissement des relations entre les profils d'Accès Progressif des utilisateurs et les stratifications créées

④ : création des relations entre stratifications et données sources (les EM du Modèle des Fonctionnalités).

L'étape suivante de conception concerne la définition des caractéristiques de présentation. Le Gestionnaire des Présentations est utilisé dans ce but. Il fait appel au Gestionnaire des Fonctionnalités pour obtenir l'ossature de l'hypermédia (Modèle S&N) ⑤. Le concepteur définit les chartes de composition et les chartes graphiques qu'il souhaite associer à l'hypermédia ainsi décrit. Le Gestionnaire des Utilisateurs et celui des Présentations communiquent pour mettre à jour dans leurs bases respectives les liens entre les profils de Préférences et les chartes.

Nous présentons à présent l'architecture fonctionnelle de KIWIS lorsqu'il est exploité par un utilisateur accédant à un SIW à Accès Progressif.

8.1.2.2 Architecture fonctionnelle d'utilisation

L'architecture fonctionnelle lors de l'utilisation est basée sur des communications entre gestionnaires qui sont très proches de celles décrites dans la section précédente. C'est avant tout dans les échanges (demandes de services entre gestionnaires) que se situe la différence.

Nous décrivons cette architecture fonctionnelle en deux temps. Tout d'abord, nous montrons comment est traité l'appel à une fonctionnalité du SIW par un utilisateur. Nous décrivons comment la page résultat est construite. Ensuite, nous expliquons le traitement impliquant des mécanismes d'accès progressif qui peuvent être appliqués à la page résultat.

La Figure 8.3 illustre les relations entre les gestionnaires lors de l'appel à une fonctionnalité du SIW (telle, par exemple, que la demande de consultation de la liste des Spécialistes, en référence au Modèle de Gestion Documentaire des Inondations utilisé à plusieurs reprises dans ce mémoire – cf. page 96).

Nous montrons comment les différents gestionnaires collaborent pour traiter un tel appel et pour produire une réponse qui soit adaptée en terme de contenu stratifié et de présentation.

L'appel à la fonctionnalité est transmis au Gestionnaire des Fonctionnalités ①. Celui-ci informe le Gestionnaire des Utilisateurs de cette demande ②. Ceci a pour effet d'engager la procédure de gestion permettant de retourner un contenu adapté en termes de stratification. Le Gestionnaire des Utilisateurs consulte la base des Utilisateurs et identifie le profil d'Accès Progressif à appliquer (*i.e.* un profil spécifique à cet utilisateur ou, à défaut, celui du groupe). Ce profil identifié, il transmet l'information au Gestionnaire des Stratifications ③. A son tour ce gestionnaire recherche dans la base qu'il contrôle, les spécifications donnant la stratification à appliquer aux données. Il communique cette information au Gestionnaire des Fonctionnalités ④, qui construit le résultat adéquat en interrogeant le Gestionnaire du Domaine ⑤.

En parallèle, le Gestionnaire des Fonctionnalités a également communiqué au Gestionnaire des Présentations qu'un appel à une fonctionnalité a été reçu (②bis). Le Gestionnaire des Présentations recherche la page type associée, en demandant au Gestionnaire des Utilisateurs d'identifier les chartes à appliquer pour cet utilisateur (③bis).

Le Gestionnaire des Fonctionnalités transmet alors au module de publication de pages Web Cocoon, un fichier XML contenant le résultat ⑥ (en termes de contenu et de spécifications pour la stratification). Le Gestionnaire des Présentations envoie, pour sa part, la feuille de style XSL à appliquer (⑥bis).

Cocoon se charge alors de construire et de renvoyer la page Web résultat ⑦ qui présente les données de façon conforme au premier niveau de stratification défini pour cette fonctionnalité et

cet utilisateur, et en respect des spécifications des chartes de composition et graphique correspondant au profil de cet utilisateur.

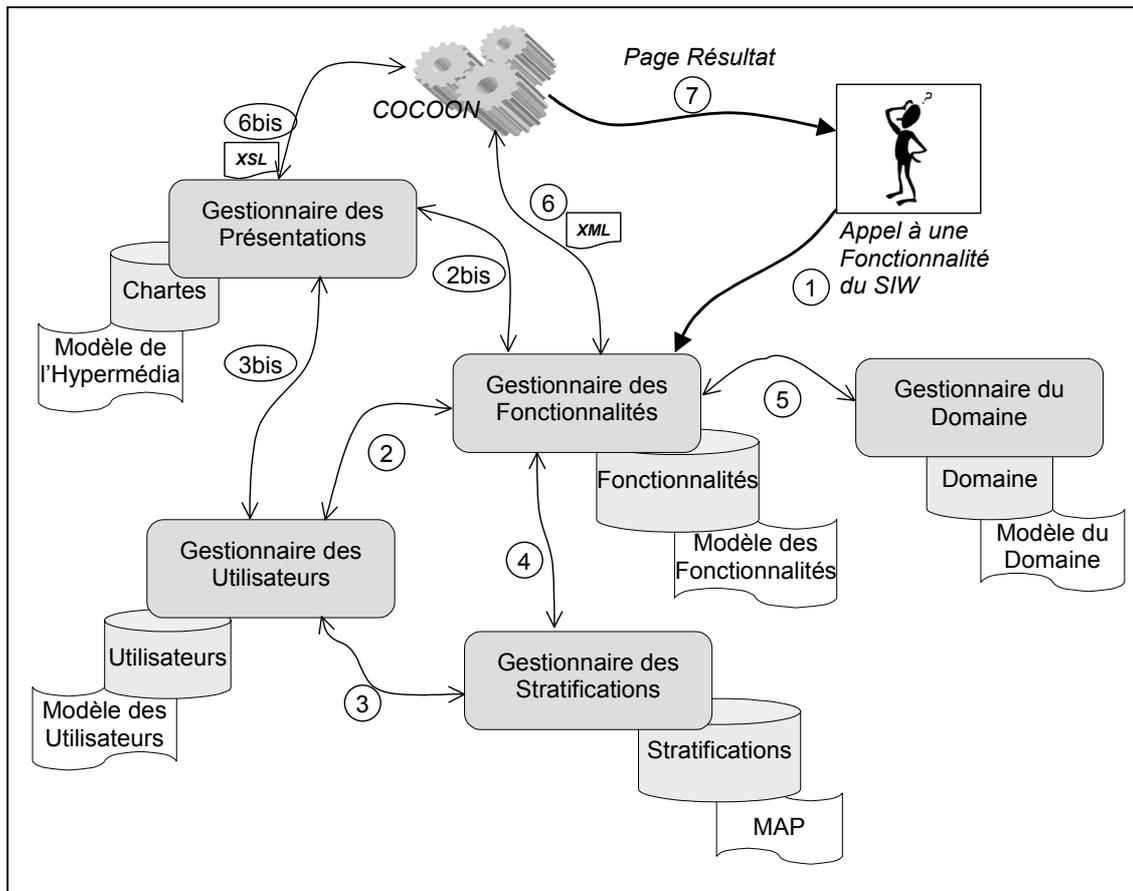


Figure 8.3 – Architecture Fonctionnelle de KIWIS lors de l'appel à une fonctionnalité du SIW

A partir de cette page résultat, l'utilisateur a donc la possibilité de procéder au dévoilement de la stratification. Il ne peut pas, à ce stade, masquer cette stratification, les données étant présentées au premier niveau de détail. L'activation du dévoilement est en fait directement envoyée au moteur de servlets Tomcat (cf. Figure 8.1). Une servlet traite la demande de dévoilement en transmettant à Cocoon les nouveaux paramètres de publication de la page. Ceci consiste pour Cocoon à ré-appliquer la feuille de style en incluant les données du fichier XML marquées comme étant de niveau supérieur de détail. Si l'utilisateur souhaite ensuite masquer les données, un schéma similaire est appliqué en exploitant les données du fichier associées au niveau de détail inférieur à celui actuellement affiché.

La section suivante est consacrée à la présentation de l'environnement offert au concepteur.

8.2. L'environnement du concepteur

Nous décrivons l'environnement dédié à l'activité de conception du SIW. Notre présentation suit pas à pas le cheminement du concepteur dans l'environnement. Des copies d'écran, tel qu'il est présenté au concepteur lors des différentes étapes, illustrent nos propos.

8.2.1 Connexion et page d'accueil du concepteur

Par défaut, le concepteur se connecte à l'adresse <http://hostname:8080/kiwis> et accède à la page reproduite par la Figure 8.4.

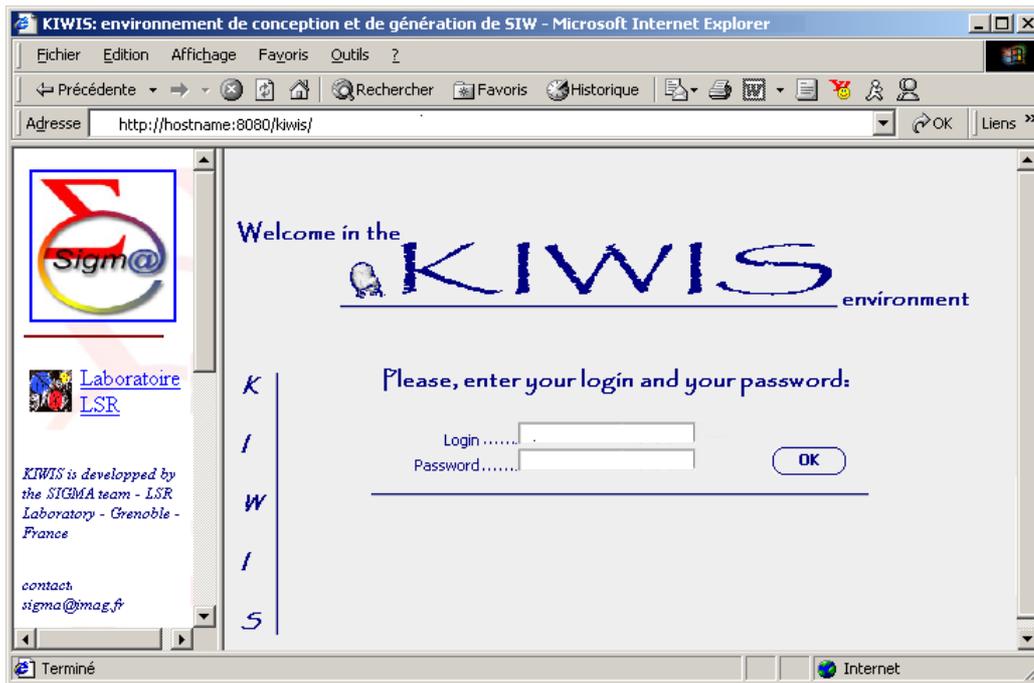


Figure 8.4 – Page de connexion à l'environnement KIWIS

Il saisit son login et son mot de passe qui lui auront été communiqués à l'installation, puis clique sur « OK ». Au moment de la soumission du formulaire, la servlet qui se charge de l'authentification des utilisateurs exécute les actions suivantes :

- récupération du *login*, du *mot de passe*
- création d'une requête adressée au Gestionnaire des Fonctionnalités qui est traitée comme illustré dans l'exemple de la section 8.1.2.2 (à ceci près que les étapes relatives à la stratification n'ont pas cours ici).
- création d'une nouvelle session pour cet utilisateur (le concepteur, utilisateur de KIWIS)
- renvoi d'une page d'accueil adaptée en termes de présentation au profil de Préférence du concepteur et contenant les fonctionnalités qu'il lui est possible d'engager : création d'un nouveau SIW, modification ou suppression d'un SIW existant, déploiement d'un SIW déjà spécifié.

La Figure 8.5 illustre la page présentée Mr Bilasco, un concepteur qui utilise KIWIS. La liste des SIW qu'il peut modifier, supprimer et générer est construite en fonction des droits du concepteur qui est connecté.

Nous nous concentrons ici sur la démarche de conception d'un SIW telle qu'elle est proposée par KIWIS. En cliquant sur l'option « *Create a new IS* », le concepteur est dirigé vers la page de création d'un nouveau système.

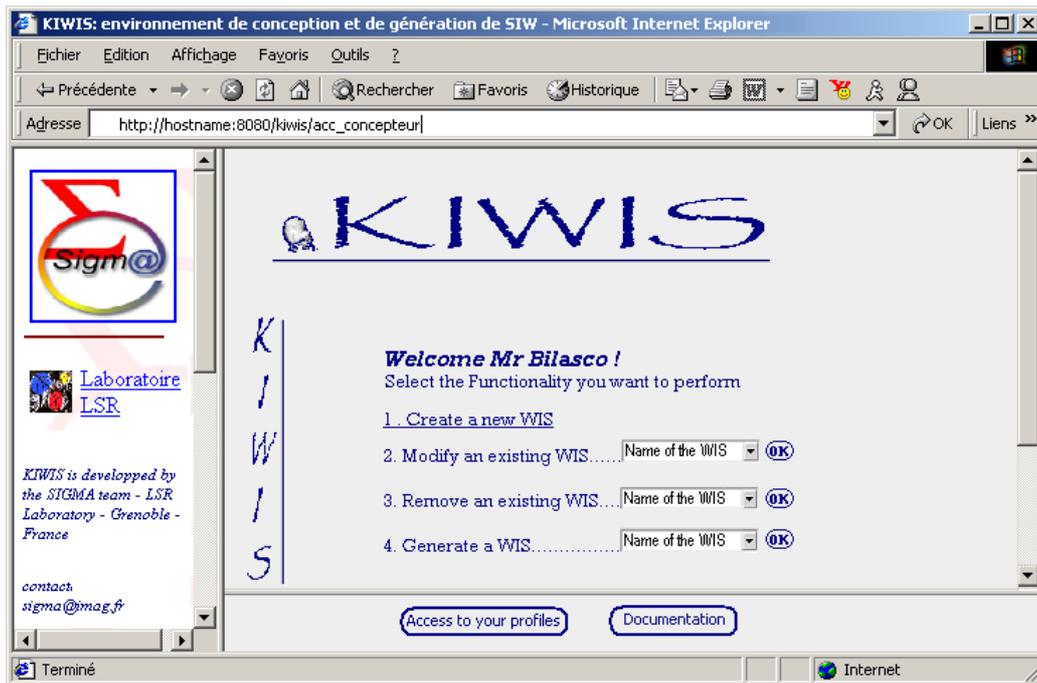


Figure 8.5 – La page d'accueil du concepteur dans KIWIS

8.2.2 Création d'un nouveau SIW

Le concepteur saisit les informations de création d'un nouveau système (champs « *WIS Name* », « *Short Description* » et « *Detailed Description* ») et choisit un logo (une image) pour ce système. La validation rend effective cette création. La Figure 8.6 montre la page de création d'un nouveau SIW.

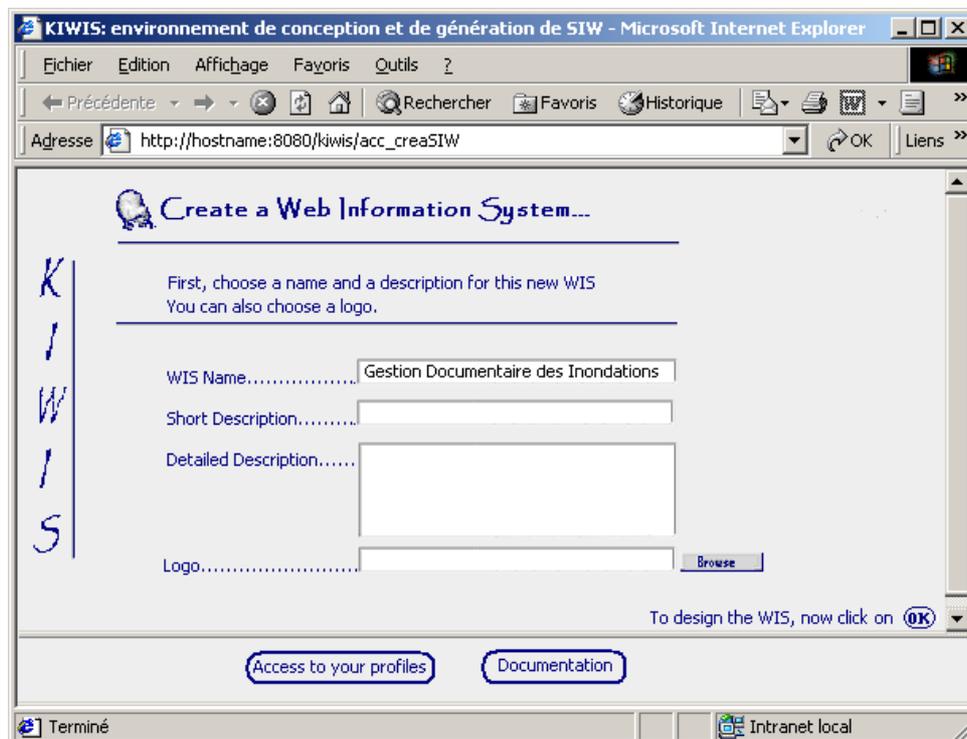


Figure 8.6 – La page de création d'un nouveau SIW

Comme dans toutes les pages qui composent l'interface de conception d'un nouveau SIW, la validation des données saisies se fait à l'aide de scripts Javascript. De plus, au niveau des servlets qui sont derrière ces pages, une validation des données est opérée afin d'empêcher le traitement des requêtes qui ne sont plus valides (suite à l'expiration de la session par exemple).

Suite au traitement du formulaire par la servlet « *ISServlet* », qui se charge de la gestion globale des SIW installés au sein de KIWIS, un répertoire portant le nom du système est créé. Dans ce répertoire, nous retrouvons la structure suivante (cf. Figure 8.7) :



Figure 8.7 – Structure du répertoire créé pour tout nouveau SIW

Le répertoire *xml*, contient les bases de données du SIW décrites dans la section 8.1.1.3. Le répertoire *xsd* contient les définitions de ces bases, selon la syntaxe XML Schema. Les répertoires *xsl* et *css* correspondent à la bibliothèque des présentations. Les deux autres répertoires sont : *images* qui stocke les images liées à ce SIW (par exemple le logo) et *work* qui stocke les données intermédiaires obtenues suite aux requêtes effectuées sur les bases de données.

Si toutes ces opérations aboutissent, une page « *Wizard* » est retournée au concepteur afin de le guider tout au long de la démarche de conception. En cliquant sur « *OK* », le concepteur accède à la page de définition du Modèle du Domaine.

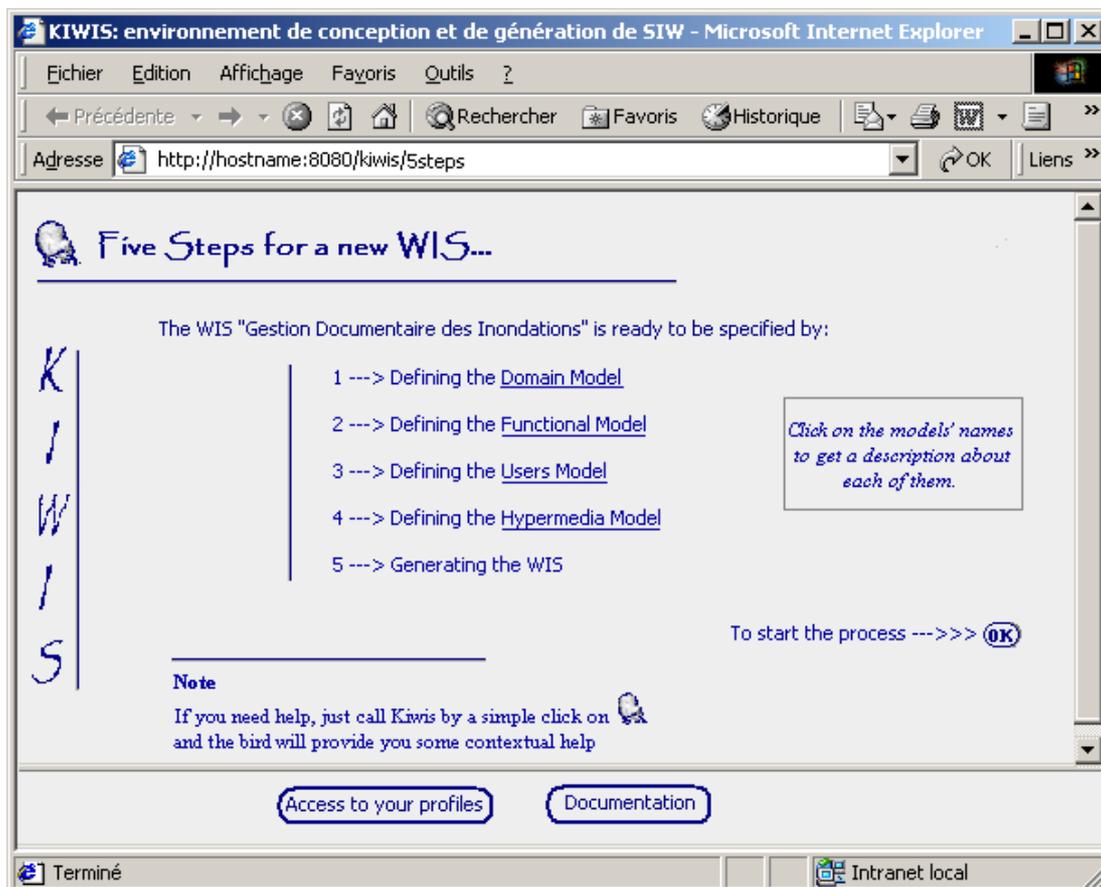


Figure 8.8 – Page « *Wizard* » présentant la démarche de conception

8.2.3 Interface de définition du Modèle du Domaine

Comme nous l'avons suggéré lors de la présentation de l'architecture de l'environnement KIWIS, le SRCO AROM peut être utilisé pour définir le Modèle du Domaine. Le concepteur dispose alors d'un environnement de modélisation graphique. Nous présentons ici l'interface à base de formulaires qui constitue un mode alternatif de définition du Modèle du Domaine. En premier lieu, le concepteur décrit les classes du système grâce à la page montrée par la Figure 8.9.

Il saisit le nom de la classe, et éventuellement, il donne une description du rôle de celle-ci au sein du système. Ensuite, il ajoute les attributs de la classe en saisissant leur nom, leur type, et leur description si nécessaire. L'ajout d'un attribut se fait en cliquant sur le lien « *Add Attribut* ». A tout moment, le concepteur peut supprimer des attributs, en les sélectionnant puis en cliquant sur « *Remove Selection* ».

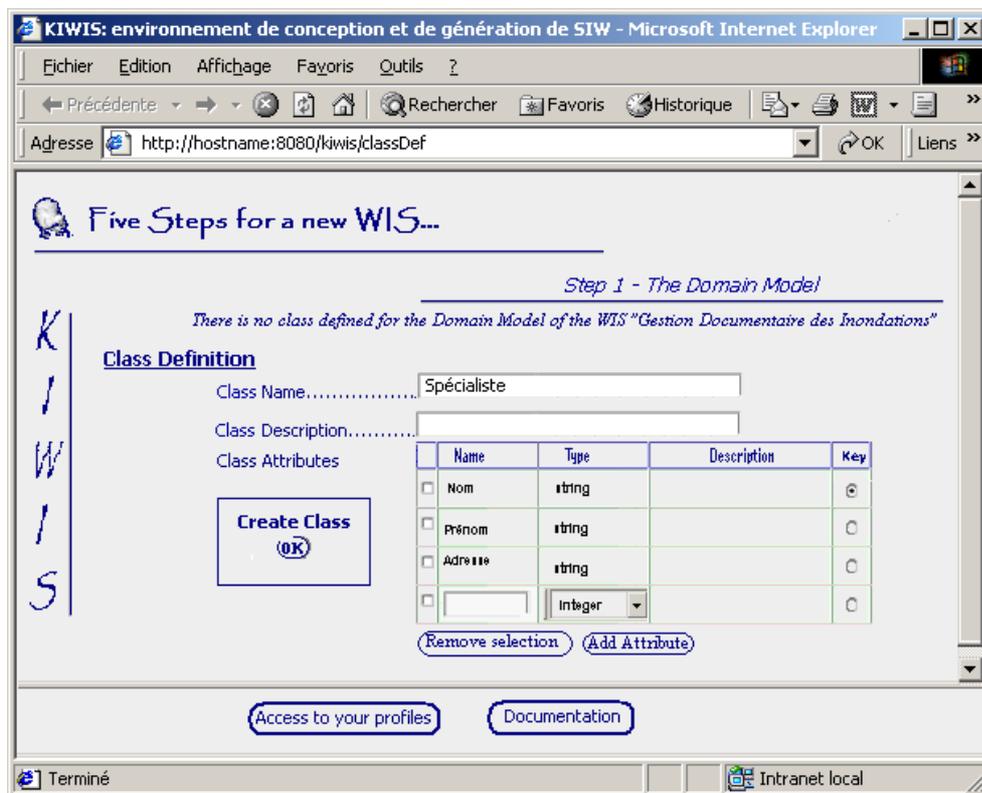


Figure 8.9 – Interface de création d'une classe du Modèle du Domaine

Une interface de type formulaire permet également de définir les associations entre les classes et de poursuivre ainsi la définition du Modèle du Domaine. La page présentée dans la Figure 8.10 est proposée au concepteur. Les liens de cette association avec les classes qu'elle implique se font à travers la spécification du type des rôles d'association (qui sont ajoutés comme cela est décrit précédemment pour l'ajout d'attributs de classes). Les classes doivent avoir été créées auparavant.

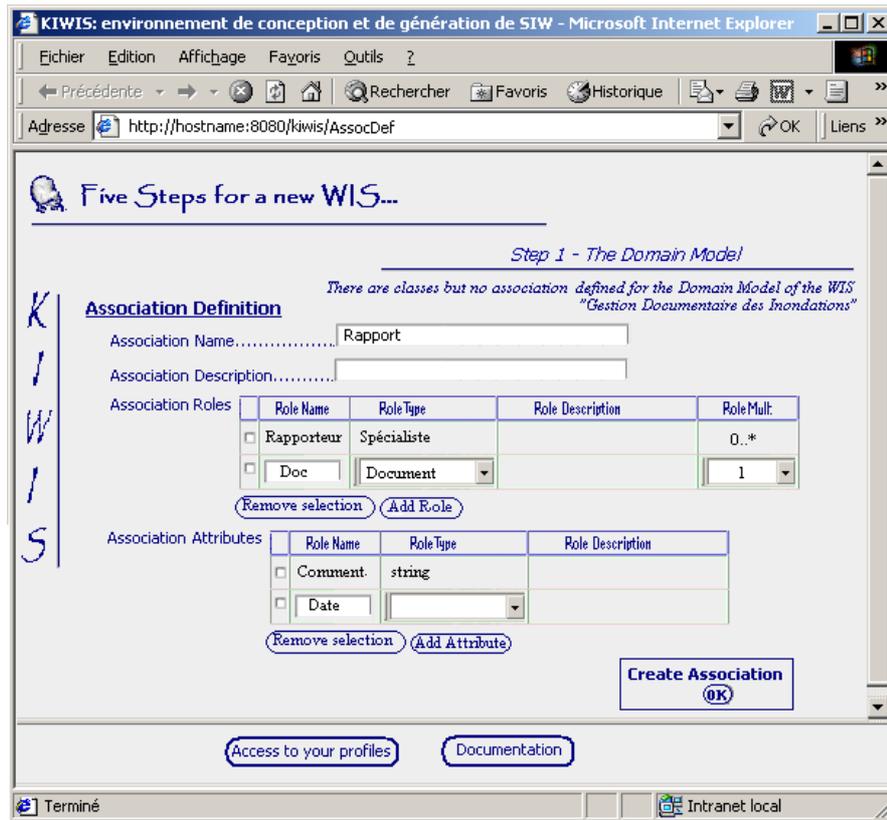


Figure 8.10 – Interface de création d'une association du Modèle du Domaine.

Lorsque toutes les classes et les associations du Modèle du Domaine ont été spécifiées, le concepteur passe à l'étape suivante. Celle-ci consiste en la création du Modèle des Fonctionnalités.

8.2.4 Interface de définition du Modèle des Fonctionnalités

Nous rappelons que la définition du Modèle des Fonctionnalités consiste pour le concepteur à décrire les différentes Fonctionnalités de chacun des Rôles Fonctionnels identifiés à partir des cas d'utilisation. Le concepteur commence par créer un Rôle Fonctionnel (nom, description de ce qu'il permet de faire, etc.). Puis, pour ce rôle, il s'engage dans la création d'une Fonctionnalité que nous décrivons ici.

La création d'une Fonctionnalité se fait grâce au formulaire illustré par la Figure 8.11. Le concepteur définit le nom de cette fonctionnalité et peut en donner une description. L'interface rappelle le Rôle Fonctionnel pour lequel cette fonctionnalité est définie. Le concepteur sélectionne ensuite la classe ou l'association sur laquelle va porter la requête associée à cette Fonctionnalité ; En d'autres termes, il spécifie la clause « From » de cette requête. Notons que l'exemple présenté ici concerne le cas d'une requête très simple, basée sur une seule association qui va être stratifiée en intension (*i.e.* masquage de la structure du résultat)³⁸. Néanmoins, le

³⁸ Notons que le prototype n'implémente actuellement que la stratification en intension. Dans une prochaine version, des commandes seront ajoutées pour que le concepteur puisse choisir le mode de stratification qu'il souhaite appliquer.

concepteur peut choisir plusieurs classes ou associations de base pour la fonctionnalité qu'il définit et construire ainsi des requêtes plus complexes.

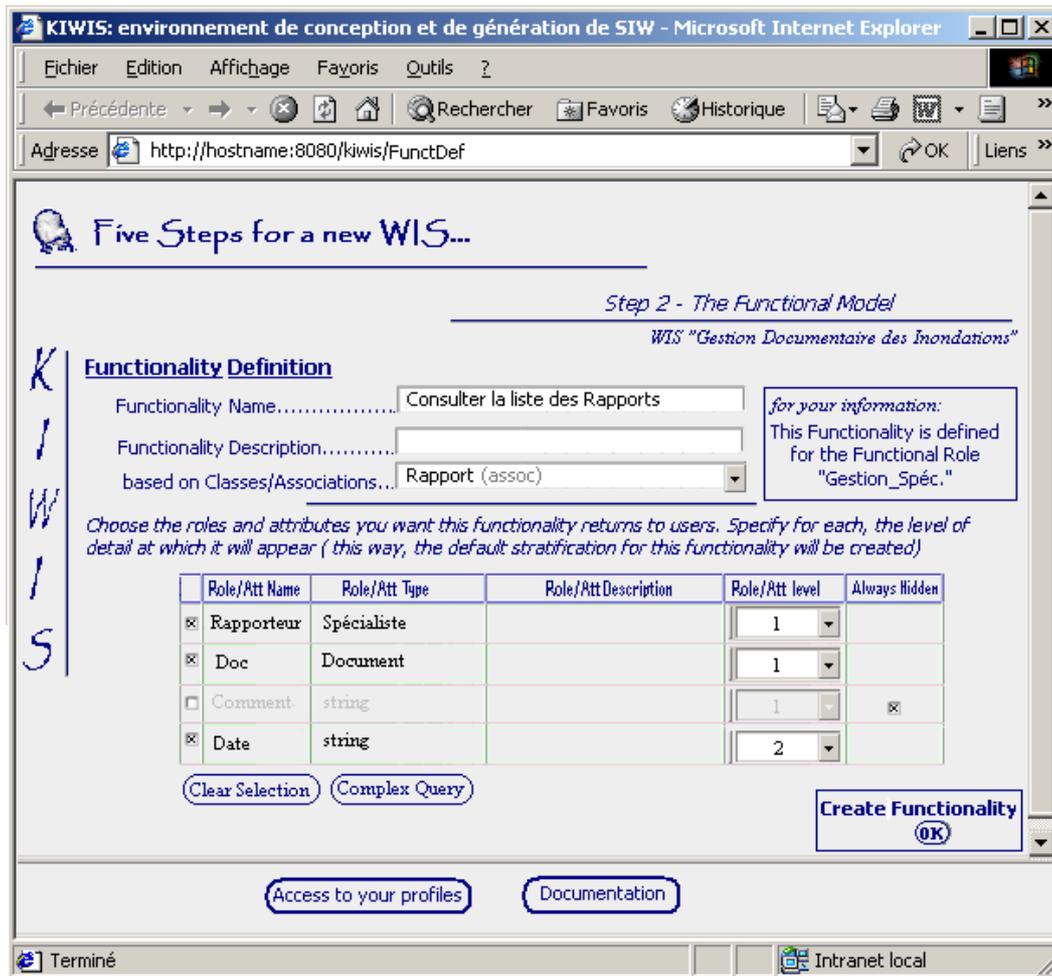


Figure 8.11 – Interface de création d'une fonctionnalité.

Le choix d'une classe ou d'une association de base pour définir la Fonctionnalité entraîne une modification dynamique de la page ; la partie en dessous du trait central s'affiche avec les informations relatives à la classe ou à l'association sélectionnée. Ici, les rôles et attributs de l'association Rapport apparaissent.

Le concepteur choisit les éléments qu'il souhaite mettre à disposition de l'utilisateur en cochant les cases associées aux rôles ou aux attributs (à gauche). En d'autres termes, il spécifie à travers cette interface les champs de la clause « Select » de la requête. Pour tout attribut (ou rôle) sélectionné, il spécifie le niveau de détail auquel celui-ci doit apparaître (à droite). Ceci a pour effet de créer la stratification par défaut du résultat de la requête. Dans l'exemple illustré par la Figure 8.11, il est spécifié que les rôles « Rapporteur » et « Doc » seront visibles au premier niveau de détail, et que l'attribut « date » le sera au second niveau (et donc atteint par dévoilement). L'attribut « comment. » n'est pas retenu dans cette stratification. Le concepteur a, de plus, spécifié que cet attribut est toujours masqué : les membres du groupe « Gestion_Spéc. » n'auront jamais accès à cette information (*i.e.* même s'ils redéfinissent la stratification, ils ne pourront jamais y inclure cet attribut confidentiel).

Le concepteur valide ses choix en cliquant sur le bouton « OK ». Une servlet contrôle la validité de la stratification. La base des Fonctionnalités contient la définition d'une nouvelle

instance de Fonctionnalité, liée au Rôle Fonctionnel adéquat. La base des stratifications contient également une nouvelle instance, attachée à la Fonctionnalité, qui donne la stratification définie pour le groupe.

Nous présentons dans la suite l'étape de définition du Modèle des Utilisateurs.

8.2.5 Interface de définition du Modèle des Utilisateurs

Lors de l'étape précédente, le concepteur a décrit des Rôles Fonctionnels. Un Rôle Fonctionnel étant attaché à un Groupe du Modèle des Utilisateurs, le système a automatiquement créé une instance de Groupe attachée à ce Rôle. Lorsqu'il atteint la page de définition du Modèle des Utilisateurs, le concepteur est informé des différents groupes dont KIWIS a déduit l'existence. Le concepteur complète les descriptions manquantes à l'aide de l'interface reproduite dans la Figure 8.12.

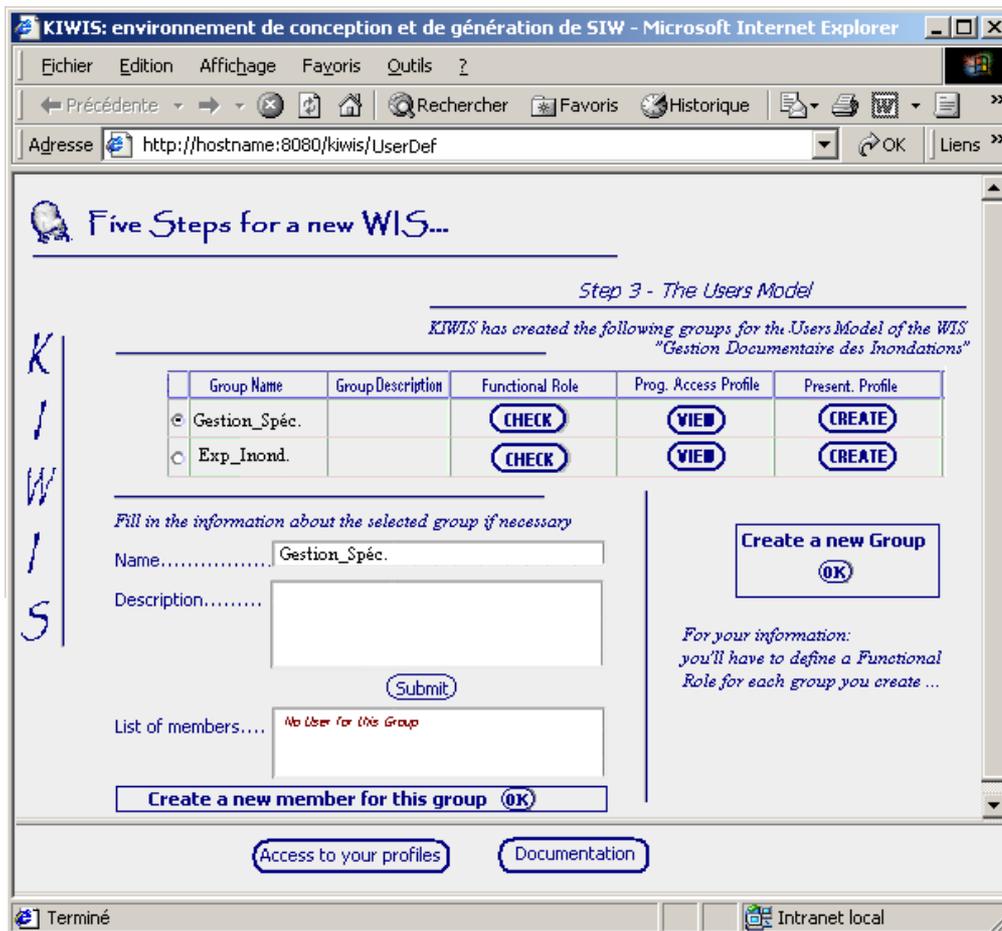


Figure 8.12 – Interface de définition des Groupes du Modèle des Utilisateurs

Il sélectionne tout d'abord le groupe pour lequel il souhaite ajouter des informations en cochant le 'radio-bouton' associé (à gauche dans le tableau). Les informations concernant le groupe sélectionné qui sont connues du système s'affichent dans les zones adéquates (ici, seul le nom du groupe est connu). Le concepteur procède aux modifications et/ou ajouts nécessaires et valide en cliquant sur « Add ». A droite, la liste des utilisateurs membres du groupe est donnée. Dans l'exemple décrit, aucun utilisateur n'a encore été créé pour le Groupe « *Gestion_Spéc.* ». Le concepteur peut procéder à l'enregistrement d'utilisateurs individuels *via* une page spécifique (atteinte en cliquant sur « OK » dans zone correspondante). Il décrit l'utilisateur en indiquant ses

nom, prénom, login, mot de passe, etc. Le lien avec le groupe sélectionné dans la page source (ici « *Gestion_Spéc* ») est automatiquement établi. Le concepteur peut également déclarer cet utilisateur en tant que membre d'autres groupes. De façon transparente, l'Espace Fonctionnel de l'utilisateur est mis à jour à chaque nouveau lien d'appartenance défini entre l'utilisateur et un groupe. Notons que dans cette page, nous donnons la possibilité au concepteur de créer une instance de groupe (cf. la commande « *create* » dans le panel intitulé « *Create a new Group* » à droite tableau). S'il procède ainsi, un nouveau groupe existe bien qu'aucun Rôle Fonctionnel n'y soit associé. Dans ce cas, dans le tableau récapitulatif des groupes (Figure 8.12), la case correspondant au Rôle Fonctionnel accueille un bouton de commande particulier (cf. Figure 8.13). Ce bouton permet d'accéder à la page de création du Rôle Fonctionnel, et par suite de créer les Fonctionnalités qui le composent (cf. section 8.2.4). Les profils d'accès progressif et de présentation ne sont pas accessibles.

| | Group Name | Group Description | Functional Role | Prog. Access Profile | Present. Profile |
|---|------------|-------------------|--------------------------|----------------------|------------------|
| ○ | GRH | | CHECK | VIEW | CREATE |
| ○ | Exp_Inond. | | CHECK | VIEW | CREATE |
| ○ | Public | | CREATION REQUIRED | | |

Figure 8.13 – Extrait de l'interface générée lorsqu'un qu'un groupe n'est pas associé à un Rôle Fonctionnel

Les pages de création/modification des groupes et des utilisateurs intègrent des liens vers les pages concernant leurs profils respectifs. Nous illustrons dans la section suivante l'accès à la page relative au profil d'Accès Progressif. Cette page permet de créer, réorganiser ou supprimer des stratifications.

8.2.6 Interface de définition des profils d'Accès Progressif

A ce stade, le concepteur a défini le Modèle du Domaine, le Modèle des Fonctionnalités et le Modèle des Utilisateurs. Nous avons vu qu'au moment de la description de toute nouvelle fonctionnalité, le concepteur peut définir la stratification par défaut à lui associer. Néanmoins, à travers l'interface de définition des profils d'Accès Progressif, nous lui donnons la possibilité de retravailler cette stratification. Cette interface permet également de définir des stratifications qui n'ont pu l'être avant. C'est notamment le cas de la stratification de l'Espace Fonctionnel d'un utilisateur particulier.

Nous illustrons le cas de la réorganisation d'une stratification en nous appuyant sur l'exemple de la fonctionnalité permettant de *consulter les rapports*, dont nous avons donné la description dans la Figure 8.11. La Figure 8.14 montre l'interface proposée au concepteur après qu'il a cliqué sur « *View* » dans la case correspondant au Profil d'Accès Progressif d'un groupe du tableau récapitulatif des Groupes (cf. Figure 8.12 ou Figure 8.13).

Le concepteur choisit la fonctionnalité qu'il souhaite réorganiser. Ceci a pour effet de présenter les différents niveaux de détail décrits pour la stratification existante. Les rôles ou attributs non utilisés dans cette stratification sont également présentés, ainsi que ceux déclarés comme confidentiels (i.e. toujours masqués). Le concepteur peut alors apporter les modifications qu'il souhaite en utilisant un ensemble de commandes associées aux opérations de réorganisation d'une stratification telles que définies par le protocole d'exploitation du MAP (cf. section 5.2.3).

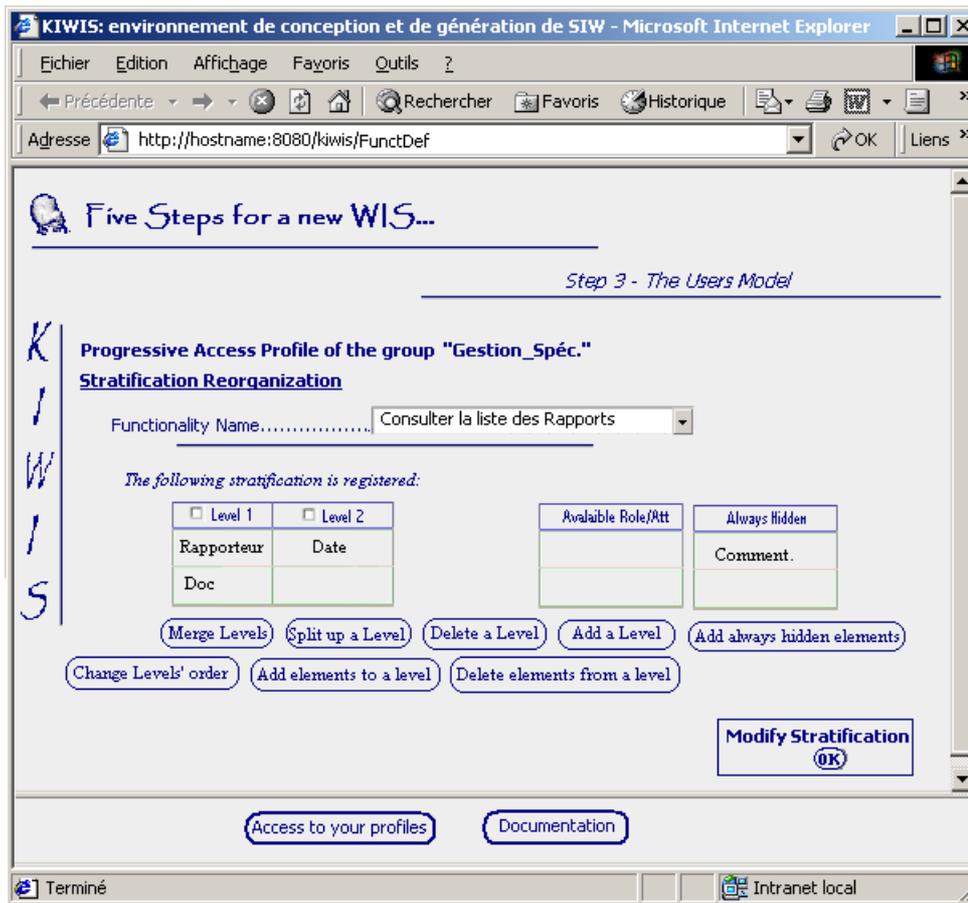


Figure 8.14 – Interface de gestion des Profils d'Accès Progressifs

L'utilisation de ces commandes entraîne une modification dynamique de la page. Par exemple, le concepteur peut choisir de finalement mettre à disposition l'information sur les commentaires établis par les spécialistes. Il peut placer l'attribut « comment. » dans la rubrique « Available Role/Att » et laisser ainsi le soin aux utilisateurs d'intégrer ou non cette information dans leurs stratifications personnelles. Une autre possibilité est de créer un niveau de détail supplémentaire qui ajouterait l'attribut « comment. ». Une colonne « Level 3 » dans laquelle apparaît cet attribut serait alors créée. Dans les deux cas, la colonne « Always Hidden » reste visible, mais contient le texte « none ».

L'étape suivante dans la conception consiste à définir le Modèle de l'Hypermédia.

8.2.7 Interface de définition du Modèle de l'Hypermédia

Dans sa version actuelle, KIWIS offre au concepteur une interface comprenant trois pages qui correspondent aux trois étapes de conception du Modèle de l'Hypermédia :

- la première page est associée à la spécification du Modèle de Structuration & Navigation ;
- la deuxième page concerne la définition des chartes de composition ;
- la troisième page est relative à la description des chartes graphiques.

À travers la description de chacune de ces pages de conception, nous expliquons comment le concepteur définit le Modèle de l'Hypermédia. Nous supposons ici que le concepteur souhaite définir ce modèle pour Madame Pad. Le cas d'un utilisateur individuel nécessitant la description

de plus d'éléments pour le Modèle de l'Hypermédia, cette supposition nous permet de décrire de façon plus complète l'interface correspondante.

8.2.7.1 Page de définition du Modèle de Structuration & Navigation.

Lors de la présentation des modèles proposés dans notre approche (*cf.* chapitre 3), nous avons expliqué que le Modèle de Structuration et Navigation est dérivé à partir du Modèle des Fonctionnalités. Conformément à ce principe, KIWIS propose au concepteur une page qui présente le résultat de cette dérivation pour un groupe ou un utilisateur donné.

La Figure 8.15 illustre le Modèle de Structuration & Navigation dérivé à partir du Modèle des Fonctionnalités défini pour Madame Pad. Dans cette page, les contextes sont représentés par des cadres contenant d'autres cadres qui donnent la stratification du concept associé à ce contexte. Ce cadre de stratification contient plusieurs zones, séparées par une ligne en pointillés, et qui correspondent aux différents niveaux de détail de la stratification. Dans chacune de ces zones, les éléments montrés au niveau de détail associé sont représentés par un bouton.

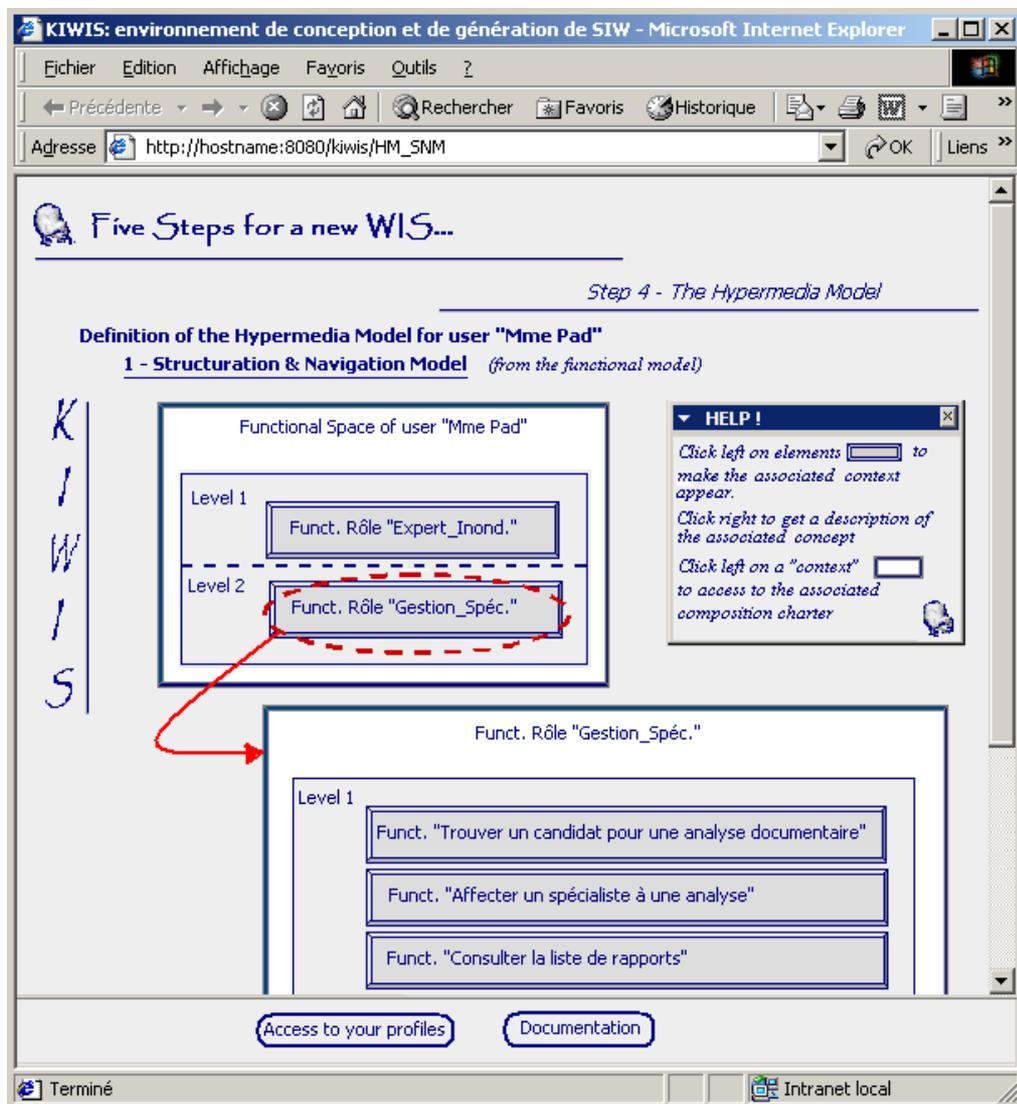


Figure 8.15 – Page de présentation du Modèle de Structuration & Navigation

Ainsi, le cadre du contexte « *Functional Space of user « Mme Pad »* » comporte un cadre de stratification à deux zones. La zone « *Level 1* » présente le Rôle Fonctionnel associé au premier niveau de détail (« *Funct. Role « Expert_Inond »* »). La zone « *Level 2* » présente le Rôle Fonctionnel ajouté au deuxième niveau de détail (« *Funct. Role « Gestion_Spéc. »* »).

À partir de cette page, le concepteur a plusieurs possibilités d'action. D'une part, lorsqu'il clique sur le bouton gauche de la souris sur un élément d'une zone de la stratification, le contexte associé apparaît dans la page. Par exemple, la Figure 8.15 illustre le résultat de cette action sur le bouton associé au Rôle Fonctionnel « *Funct. Role « Gestion_Spéc. »* » : le contexte de ce rôle est donné en bas de la page.

Lorsque le concepteur clique dans un contexte, il accède à la page de définition de la charte de composition. Plus précisément, cette page affiche le formulaire de sélection/création de la page type ou de la présentation type à associer.

8.2.7.2 Page de définition de la charte de composition

La page de définition de la charte de composition offre au concepteur la possibilité d'associer aux contextes les pages types par défaut que nous avons décrites dans la section 6.3.2.1. A travers un formulaire, il peut choisir d'apporter des modifications à ces pages types. Nous précisons toute fois que le corps de la page, qui contient l'essentiel des informations et les mécanismes de navigation pour l'accès progressif, est construit automatiquement. Les données (contenu issu des différents modèles) sont attachées de façon transparente aux composants du corps chargés de les présenter à l'utilisateur. Par exemple, pour une fonctionnalité donnée, le code permettant de la réaliser (*i.e.* principalement la requête qu'elle utilise, donnée par le Modèle des Fonctionnalités) est généré dans un composant spécifique (*i.e.* le *composant requête*, cf. section 6.3.2.1).

Nous donnons la possibilité au concepteur d'ajouter ou de supprimer les régions optionnelles d'une page type que sont les entête, pied de page, barre d'outils, et barre de navigation (*cf.* Figure 8.16). Le concepteur peut accéder à la composition interne de chacune des régions et la modifier s'il le souhaite.

Lorsque le concepteur valide ses choix, une nouvelle page type est créée. Le profil de Préférence de l'utilisatrice Mme Pad fait référence à celle-ci. Pour toutes les pages présentant un Rôle Fonctionnel auquel à accès Mme Pad, cette page type sera utilisée pour en définir la composition.

Le concepteur procède de façon similaire pour décrire les pages types des autres contextes, c'est-à-dire le contexte d'Espace Fonctionnel et celui de Fonctionnalités. Notons que s'il n'apporte aucune modification aux pages types par défaut, alors le profil de Préférence de l'utilisatrice n'établit pas de lien avec celles-ci. Ces pages-types sont référencées par les profils de Préférence associés aux groupes. Lors de la génération des pages pour un utilisateur, les spécifications données par le profil du groupe dont cet utilisateur est membre seront utilisées, conformément à la relation de substitution existant entre le profil de Préférence du groupe et celui d'un utilisateur membre de ce groupe (*cf.* section 6.4.4.2).

Pour tout élément qui entre dans la composition d'une page type, et pour la page type elle-même, le concepteur peut définir les présentations types à appliquer. Il accède pour cela aux pages de définition de la charte graphique.

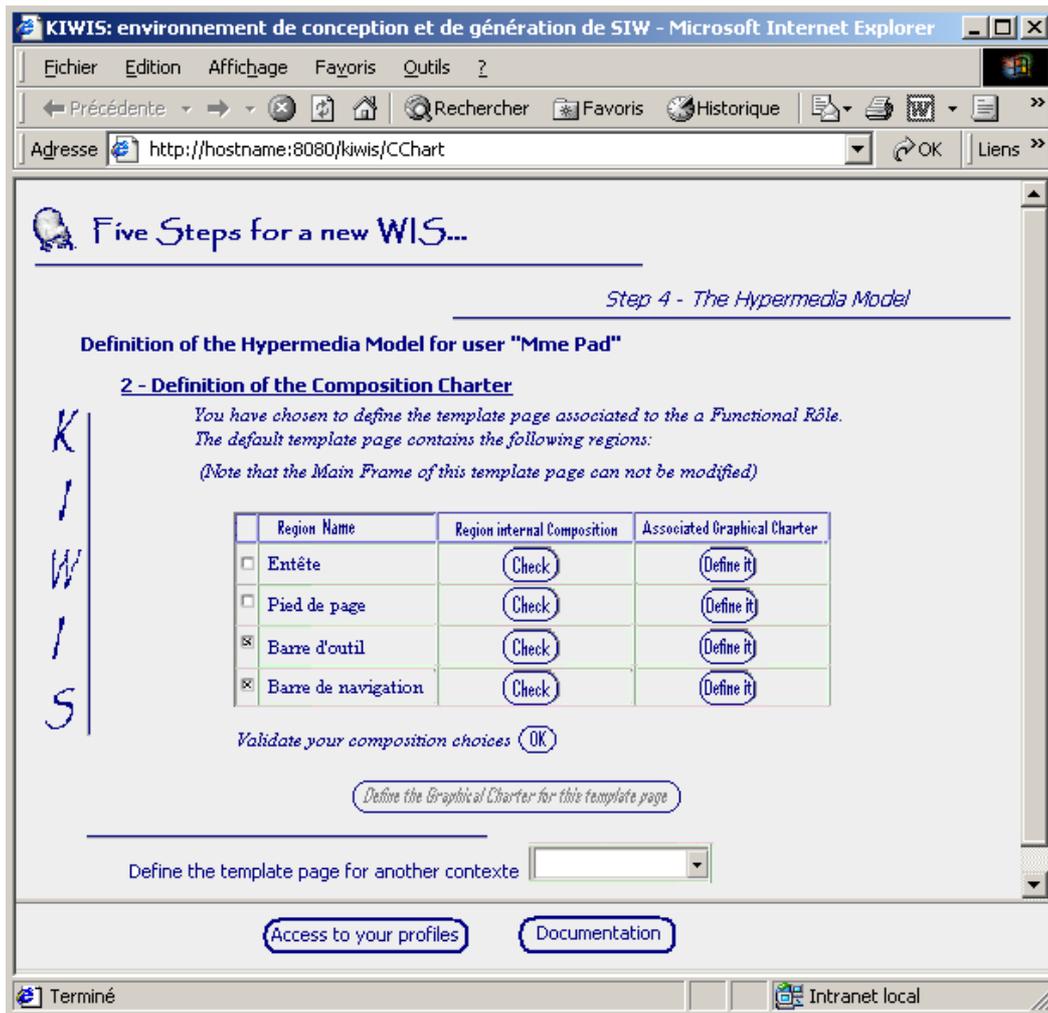


Figure 8.16 – Page de définition des chartes de composition - Définition d'une page type

8.2.7.3 Page de définition de la charte graphique

La définition de la charte graphique suit le même principe que celui décrit pour la charte de composition. Les présentations types sont associées aux pages types ou aux éléments de pages types et sont référencées par les profils de Préférence de groupe ou d'utilisateur.

Dans la version actuelle du prototype le concepteur ne peut que choisir d'appliquer les présentations types fournies par KIWIS. L'interface de création de nouvelles présentations types n'est pas implémentée à ce jour. Le concepteur peut néanmoins utiliser un éditeur de textes pour définir les feuilles de styles spécifiques dont il a besoin, en se basant sur ceux correspondant aux présentations par défaut. Dans ce but, nous proposons un formulaire permettant au concepteur d'associer, à un élément de la charte de composition (page-type, région, composant de région), la feuille de style adéquate dont il est l'auteur.

8.2.8 Génération du SIW

Lorsque toutes les étapes de la démarche ont été suivies, le concepteur peut procéder à la génération du SIW. Cette étape de génération consiste en fait à rendre le SIW accessible aux utilisateurs. Dans la version actuelle de KIWIS, tous les SIW conçus sont hébergés sur une même machine. Lors de la génération d'un SIW, le concepteur ajoute celui-ci à la liste des SIW disponibles sur KIWIS.

La Figure 8.17 illustre la page d'accueil de l'environnement KIWIS telle qu'elle est présentée à l'utilisateur, après que celui-ci a entré son login et son mot de passe (cf. Figure 8.4). Cette page présente la liste des SIW développés en utilisant le prototype et auxquels l'utilisateur a accès en fonction de son identité. KIWIS contrôle en effet les droits des utilisateurs sur les SIW et construit, dès cette page d'accueil, une réponse adaptée à l'utilisateur.

L'utilisateur sélectionne le SIW qu'il souhaite visiter en cliquant sur l'image associée. La construction dynamique de la page d'accueil du SIW sélectionné est engagée. Envoyée à l'utilisateur, cette page marque le début d'une session d'utilisation. Dans la section suivante, nous décrivons comment l'utilisateur évolue dans cet environnement.



Figure 8.17 – Page d'accueil de KIWIS : présentation des SIW accessibles à un utilisateur

8.3. Utilisation d'un SIW généré par KIWIS

Dans cette section, nous mettons en évidence le fonctionnement d'un SIW conçu et généré avec KIWIS. Cette présentation se divise en deux parties. Dans la première partie, nous décrivons l'Espace Fonctionnel d'un utilisateur et les possibilités de navigation offertes en termes d'Accès Progressif. Nous utilisons principalement le cas de Madame Pad, utilisatrice du SIW « *Gestion Documentaire des Inondations* » en guise d'illustration (cf. section 6.3.1.3). Dans la seconde partie, nous expliquons comment cette utilisatrice accède à son profil d'Accès Progressif et contrôle ainsi les stratifications qui la concerne.

8.3.1 Navigation dans un SIW à Accès Progressif

Dans cette section, nous décrivons le SIW tel qu'il est construit pour un utilisateur donné. Nous insistons plus particulièrement sur les mécanismes liés à l'Accès Progressif, permettant :

- de dévoiler / masquer un contexte. Par contexte, nous faisons référence à l'Espace Fonctionnel, au(x) Rôle(s) Fonctionnel(s) et aux Fonctionnalités accessibles pour un utilisateur. Dévoiler et/ou masquer ces contextes signifie accéder à des représentations plus ou moins détaillées de ceux-ci.
- de passer d'un contexte à l'autre, c'est-à-dire naviguer dans la structure arborescente de l'hypermédia.

Nous insistons sur le fait que les différents contextes sont stratifiés de telle façon que l'utilisateur dispose, dès le premier niveau, des informations les plus pertinentes pour lui. En conséquence, les liens de dévoilement et de masquage doivent être compris comme des possibilités de navigation supplémentaires offertes à l'utilisateur qui, ponctuellement souhaite étendre la quantité d'informations disponibles.

8.3.1.1 Dévoilement et Masquage de l'Espace Fonctionnel

Lorsque Madame Pad se connecte au SIW « *Gestion Documentaire des Inondations* », une page d'accueil lui présente son Espace Fonctionnel (cf. Figure 8.18, en haut). Cette page est adaptée aux besoins de l'utilisatrice capturés lors de l'analyse et de la conception du système. La stratification de l'Espace Fonctionnel est, en effet, telle que spécifiée dans son profil d'Accès Progressif. La page est de plus adaptée en terme de présentation : sa composition et son apparence sont conformes au profil de Préférence défini pour Madame Pad.

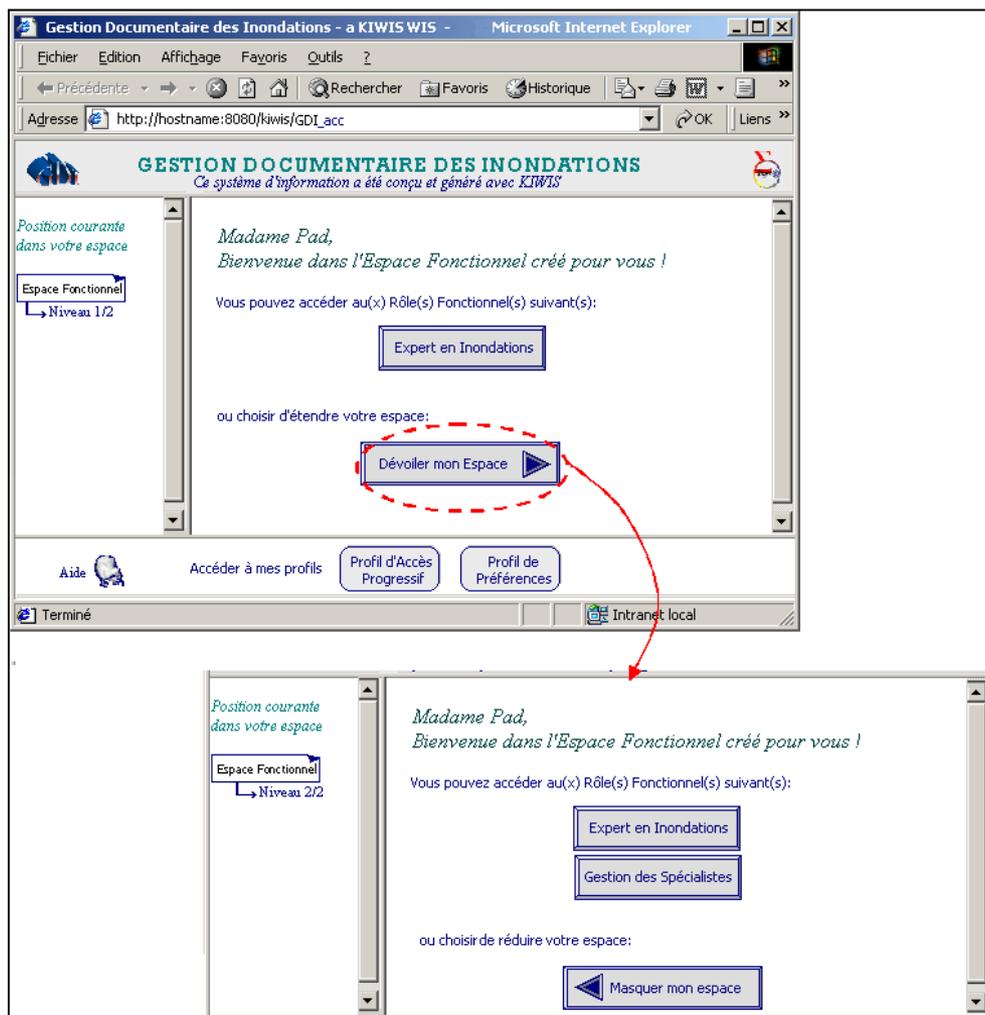


Figure 8.18 – Page d'accueil d'un utilisateur : Présentation de son espace fonctionnel

Nous décrivons la composition de cette page. Nous rappelons que, conformément à la notion de charte graphique, la composition de celle-ci est donnée par la page type correspondant au contexte d'Espace Fonctionnel du Modèle de Structuration et Navigation. Les pages des autres contextes (i.e. de Rôle Fonctionnel et de Fonctionnalités) correspondent à une composition similaire (ceci n'est pas obligatoire, mais nous permet de simplifier la présentation³⁹).

Cette page est composée d'une barre de titre horizontale, placée en haut de la page. Une barre d'outils, également horizontale est positionnée en bas de cette page. Elle donne accès à l'aide en ligne ainsi qu'aux profils d'Accès Progressif et de Préférences. A gauche, un *frame* vertical est utilisé pour informer Madame Pad de sa position dans l'hypermédia. Les informations apparaissent au fur et à mesure que l'utilisatrice descend dans l'arborescence des pages. Les informations contextuelles présentées dans les cadres (e.g. *Espace Fonctionnel*) sont des hyperliens permettant de se replacer directement dans le contexte associé, présenté conformément au dernier niveau de détail courant. Enfin, le cadre principal contient les informations relatives à l'Espace Fonctionnel et à sa stratification.

8.3.1.2 De l'Espace Fonctionnel à un Rôle Fonctionnel

Cet Espace Fonctionnel permet à Madame Pad d'accéder au Rôle Fonctionnel « *Expert en inondations* ». Actionner le bouton de commande associé lui permet d'atteindre les fonctionnalités correspondant à ce rôle. Elle peut également choisir de dévoiler son espace pour accéder à plus de Rôles Fonctionnels. Imaginons qu'elle choisisse cette option : il s'agit donc ici d'utiliser la fonction de dévoilement d'une stratification. L'extrait d'interface, reproduit en bas de la Figure 8.18, illustre le résultat de cette action :

- le rôle « *Gestion des Spécialistes* » devient accessible,
- la stratification de l'Espace Fonctionnel de Madame Pad ne comportant que deux niveaux de détail, il devient seulement possible de masquer cette stratification (le bouton « *Masquer mon espace* » remplace « *Dévoiler mon espace* »).
- le *frame* de navigation indique que la position de Madame Pad dans l'arborescence est passée du 1^{er} niveau de détail au 2^{ème} et dernier niveau de détail de l'Espace Fonctionnel.

Nous supposons à présent que Madame Pad décide d'activer le Rôle Fonctionnel « *Gestion des Spécialistes* ». Elle atteint alors la page correspondant à ce contexte, en ayant utilisé la fonction de changement de contexte associée.

8.3.1.3 Dévoilement et Masquage d'un Rôle Fonctionnel à une Fonctionnalité

La Figure 8.19 illustre deux pages correspondant à la présentation de ce Rôle Fonctionnel.

En haut, la page atteinte lorsque le rôle « *Gestion des Spécialistes* » a été activée depuis l'Espace Fonctionnel (cf. Figure 8.18). La représentation au plus bas niveau de détail est donnée. En bas, la même page mise à jour suite à une demande de dévoilement de ce rôle. La représentation de niveau de détail supérieur (i.e. le deuxième sur quatre possibles) est utilisée. Depuis cette configuration, une demande de masquage permet de retourner à la page précédente (i.e. au niveau 1 de détail).

³⁹ Dans les chartes de composition par défaut que nous proposons, la page type de l'Espace Fonctionnel et celles des Rôles Fonctionnels et Fonctionnalités diffèrent (cf. section 6.3.2.1). Nous supposons ici que le profil de présentation de Madame Pad a été redéfini spécifiquement.

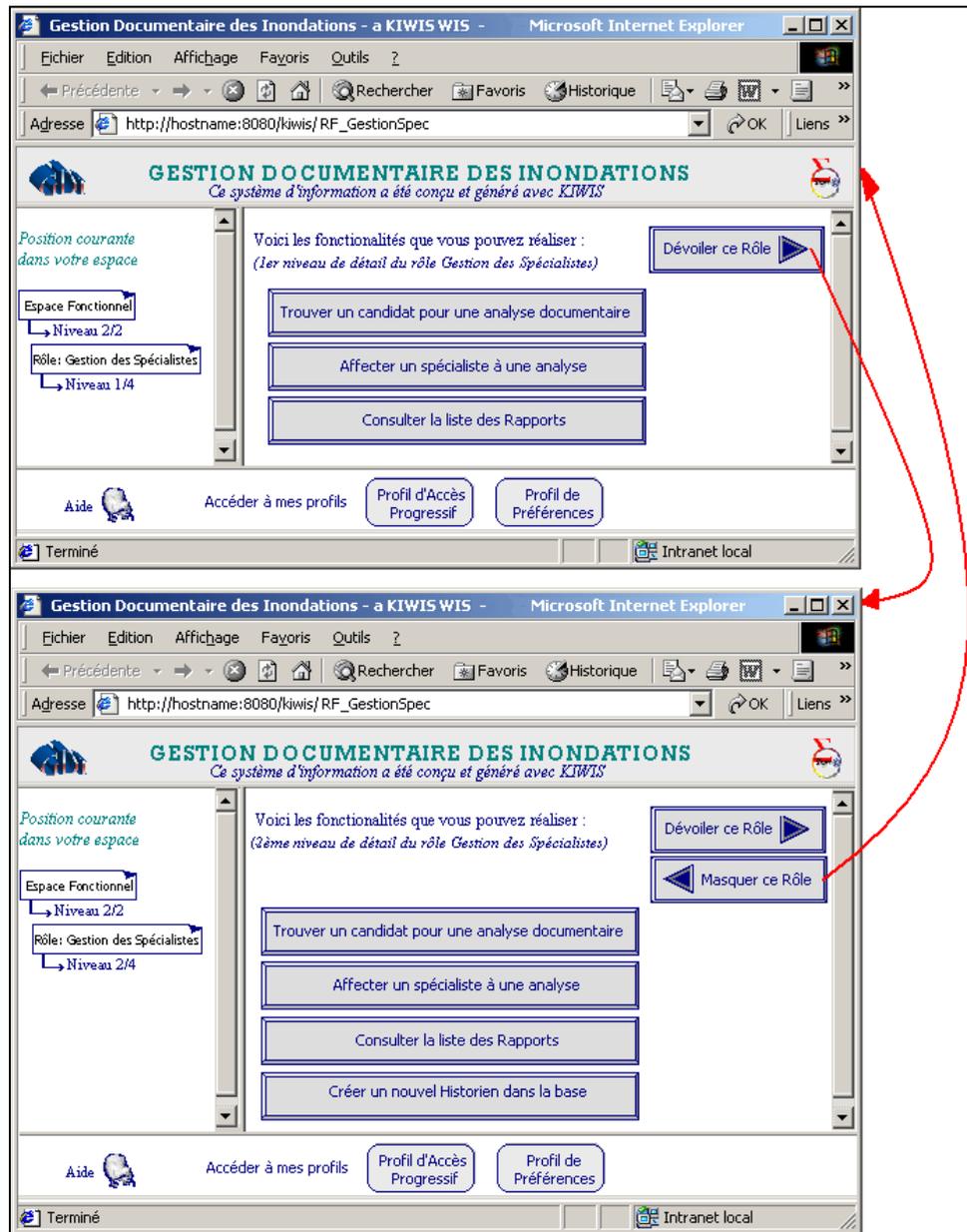


Figure 8.19 – Pages d'accès à un rôle fonctionnel : exemple de navigation entre deux niveaux de détail de ce rôle

8.3.1.4 Du Rôle Fonctionnel à une Fonctionnalité

De façon similaire à ce que nous avons décrit pour le passage d'un Espace Fonctionnel à un Rôle Fonctionnel, l'utilisateur procède au passage d'un Rôle Fonctionnel à une Fonctionnalité en cliquant sur le bouton associé à celle-ci. Il atteint alors la page consacrée à la Fonctionnalité choisie. Nous supposons à présent que Madame Pad choisit de réaliser la fonctionnalité « Consulter la liste des Rapports ». La

Figure 8.20 illustre l'interface correspondante.

Notons que la fonctionnalité de consultation présentée met en évidence une combinaison de stratifications en intension⁴⁰. D'une part la stratification relative aux informations sur les Rapports est telle qu'en actionnant le bouton « Dévoiler », le deuxième niveau atteint permet de rendre visible l'attribut « date » (date à laquelle a été établi le rapport). L'extrait de page montré dans la

Figure 8.21 illustre le résultat de cette action. Une colonne ajoutée au tableau fournie cette information pour chaque item de la liste.

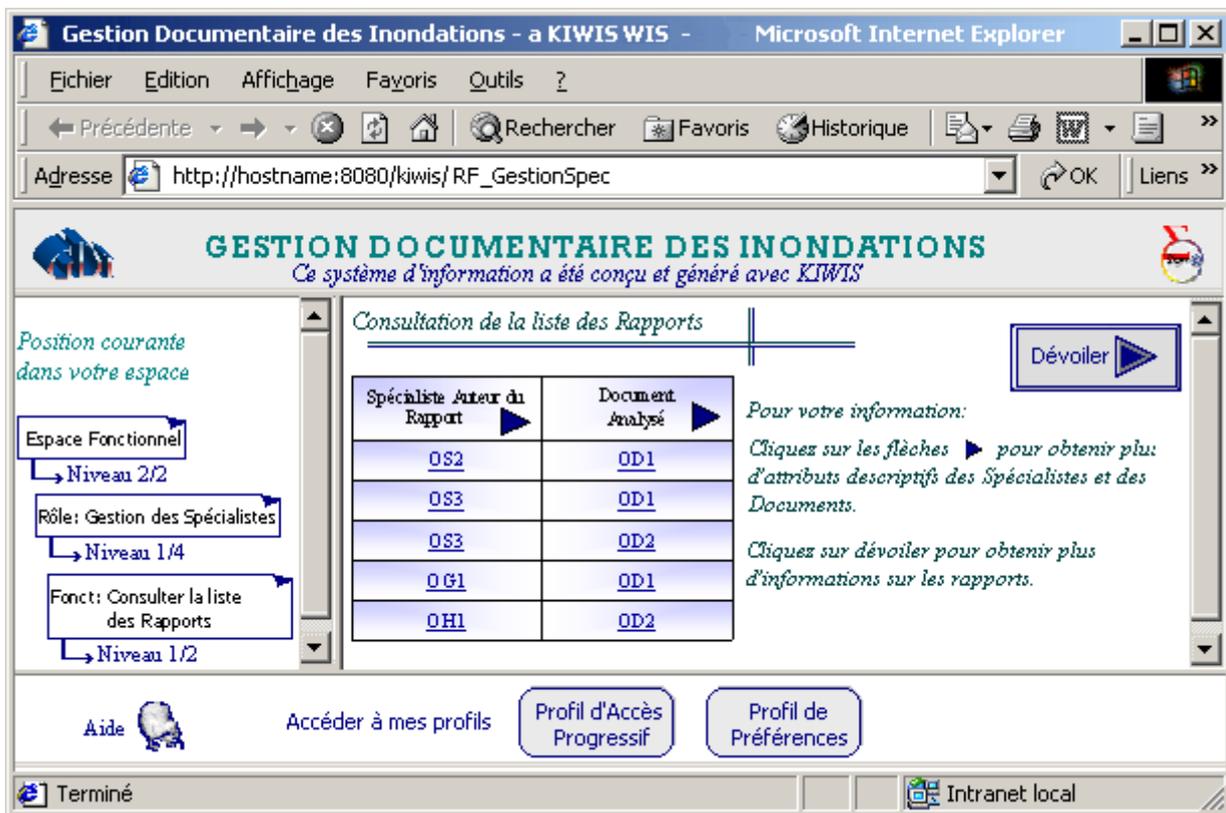
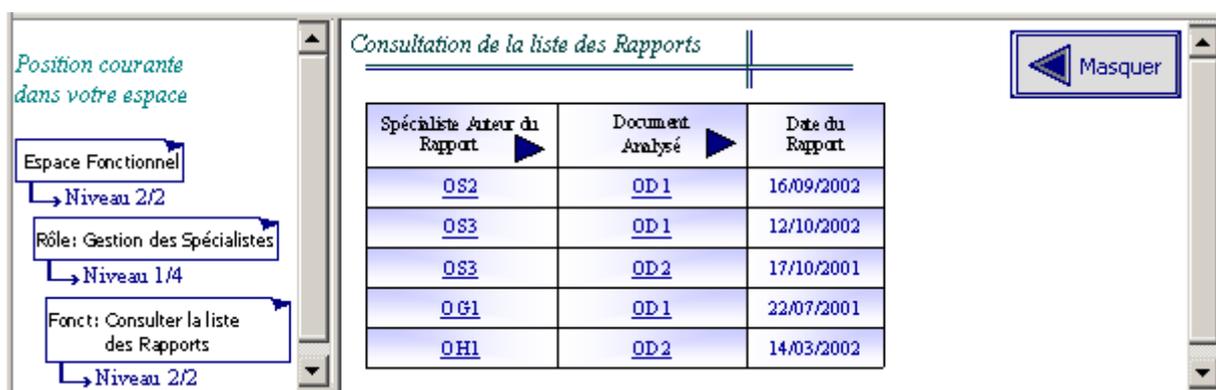


Figure 8.20 – Page dédiée à la Fonctionnalité de consultation des Rapports.



⁴⁰ Nous soulignons que la stratification par défaut définie pour la Fonctionnalité « Consulter la liste des Rapports » pour le Rôle Fonctionnel « Gestion_Spéc. » ne met pas en œuvre cette combinaison de stratifications. Une redéfinition de la stratification dans le profil d'Accès Progressif de Madame Pad est à l'origine de l'exemple illustré ici. Cette réorganisation est décrite dans la section 8.3.2.

Figure 8.21 – Extrait de la page obtenue suite au dévoilement de la Fonctionnalité de consultation des Rapports

D'autre part, deux autres stratifications décrites pour la fonctionnalité permettent d'accéder à plus d'informations à propos des Spécialistes et des Documents pour chacun des items de la liste. Ainsi, les boutons en forme de flèche (*i.e.* la forme associée au dévoilement) placés dans les deux colonnes concernées permettent de voir de façon plus détaillée les instances de Spécialistes et de Documents. Les stratifications relatives à ces deux classes sont indépendantes (*i.e.* elles peuvent être dévoilées indépendamment l'une de l'autre). L'extrait de page reproduit dans la

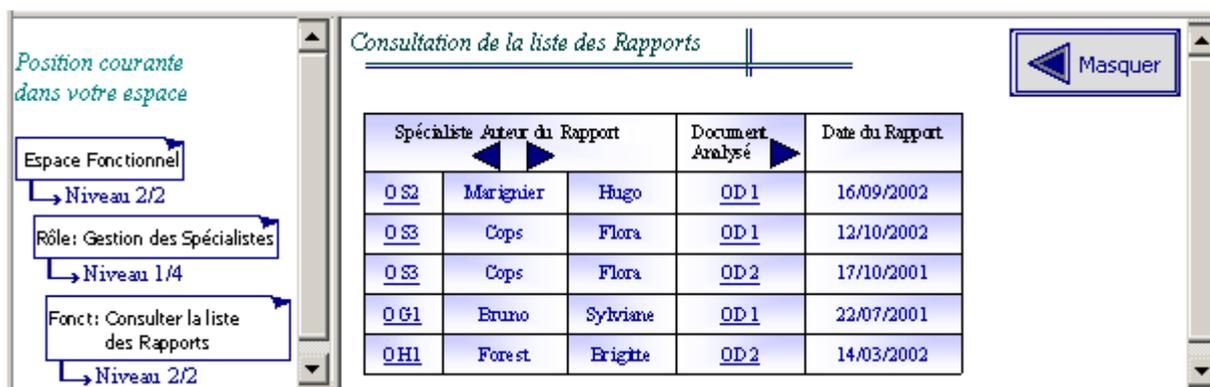


Figure 8.22 montre le résultat d'un dévoilement opéré sur la stratification de définie pour les Spécialistes. Leurs noms et leurs prénoms apparaissent.

Figure 8.22 – Extrait de la page obtenue suite au dévoilement d'une stratification combinée

8.3.2 Gestion du profil d'Accès Progressif

L'utilisateur dispose d'un accès à une page lui permettant de gérer les modalités de l'accès progressif qui lui est proposé dans le SIW. Cet accès est porté par le bouton « Profil d'Accès Progressif » situé dans la barre d'outil placée en bas de toutes les pages présentées à l'utilisateur (*cf.* par exemple

Figure 8.20).

Un utilisateur peut définir (ou redéfinir) les stratifications associées :

- à son Espace Fonctionnel. Il intervient donc ici sur les REM dont les éléments sont les Rôles Fonctionnels.
- à chacun de ses Rôles Fonctionnels. Il décrit ou modifie les REM dont les éléments sont des Fonctionnalités.
- à chacune des Fonctionnalités. Il agit ici sur les REM qui sont associées aux résultats de requêtes utilisées par chaque Fonctionnalité.

Des pages correspondent à chacune de ces possibilités. Nous illustrons la définition (ou la réorganisation) d'une stratification en reprenant l'exemple de la Fonctionnalité qui permet de « consulter la liste des rapports ». Cet exemple nous permet de mettre en évidence la différence existant entre une réorganisation de stratification effectuée par le concepteur (présentée pour cette même fonctionnalité dans la section 8.2.6) et la réorganisation telle qu'elle est pratiquée par un utilisateur individuel.

La Figure 8.23 présente la page dédiée à la réorganisation de la stratification d'une fonctionnalité (il s'agit, en l'occurrence, de la Fonctionnalité « consulter la liste des rapports » que l'utilisateur aura sélectionnée au préalable dans une page prévue à cet effet).

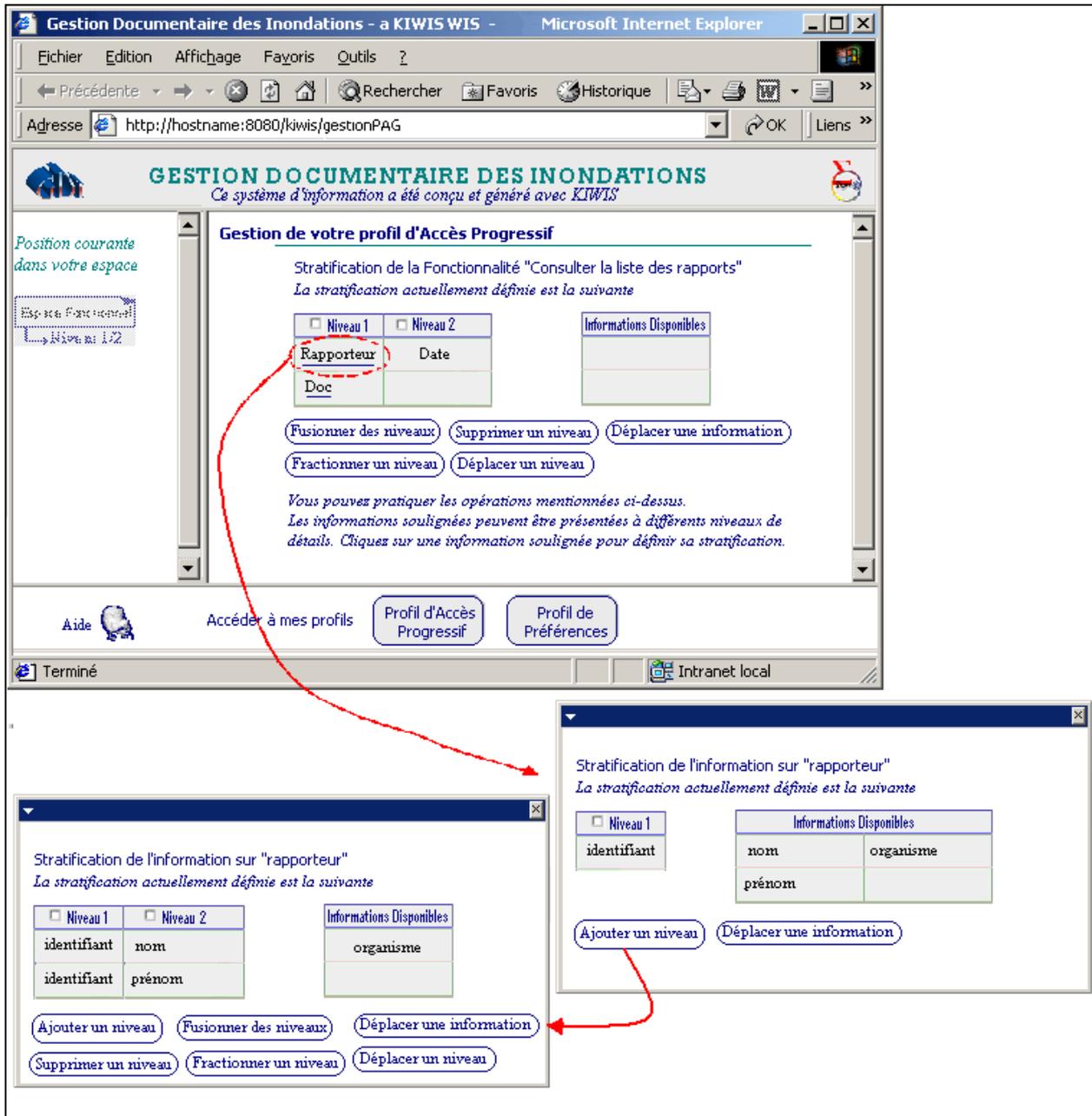


Figure 8.23 – Gestion de son profil d'Accès Progressif par l'utilisateur. Exemple de réorganisation d'une stratification existante.

8.4. Conclusion sur l'environnement KIWIS

Dans ce chapitre, nous avons décrit l'environnement KIWIS que nous développons afin de montrer la faisabilité de l'approche par Accès Progressif. Différentes versions du prototype, et notamment la plus récente présentée ici, nous ont permis de valider notre proposition et de l'opérationnaliser. Nous avons, dans un premier temps, décrit l'architecture de cet

environnement d'un point de vue logiciel et fonctionnel. Nous avons ensuite présenté les deux facettes de cet environnement, répondant à deux objectifs distincts.

D'une part, nous avons développé KIWIS dans le but d'offrir aux concepteurs de SIW un outil supportant la démarche de conception proposée. Dans cette optique, les différents modules de l'environnement, appelés gestionnaires, intègrent les modèles que nous avons décrits au chapitre 6. Ces gestionnaires contrôlent la production des modèles tout au long du processus de développement du SIW. L'interface destinée aux concepteurs se présente sous la forme de pages Web correspondant aux différentes étapes du processus.

D'autre part, KIWIS est également l'environnement sous-jacent à l'utilisation des SIW conçus et générés avec notre approche. Les gestionnaires contrôlent alors l'exploitation des modèles décrits par le concepteur. Cette exploitation est fortement guidée par les principes de l'Accès Progressif et d'adaptabilité décrits au niveau conceptuel et qui sont mis en œuvre dans tout SIW généré par KIWIS.

CONCLUSIONS

9. BILAN ET PERSPECTIVES

Le domaine d'étude de cette thèse est celui des Systèmes d'Information basés sur le Web (SIW) et de leur adaptabilité aux utilisateurs. En tant que Système d'Information, le SIW d'une organisation se présente comme un ensemble de ressources humaines, de moyens techniques, de règles de fonctionnement, de collections de données et de procédés pour traiter ces données. En tant que système basé sur le Web, les traitements sont opérés à travers une interface Web qui exploite les principes d'une structuration hypermédia de l'information. Si l'on considère ces deux visions du SIW, les capacités d'adaptation que l'on peut en attendre concernent potentiellement différentes dimensions relatives d'une part, aux aspects fonctionnels et, d'autre part, aux caractéristiques hypermédias.

Bilan du travail réalisé

L'étude des méthodes de conception d'applications basées sur le Web nous a permis d'observer que les différentes approches reposent sur des modèles qui décrivent sensiblement les mêmes dimensions. Ainsi, on retrouve généralement :

- Un modèle de données qui décrit les concepts du domaine manipulés par l'application.
- Un modèle de l'hypermédia, constitué d'un modèle de navigation qui met en évidence la structure et les possibilités de navigation, et d'un modèle de présentation destiné à décrire l'apparence et la mise en forme des pages.
- Un modèle relatif aux aspects fonctionnels, souvent basé sur des cas d'utilisations, et, plus rarement, donnant une représentation des tâches disponibles pour les utilisateurs (dans AWIS-M [Gnah00] notamment).
- Un modèle des utilisateurs décrivant, en fonction des approches leurs caractéristiques, préférences, buts, etc.

En matière d'adaptation, ces méthodes se basent sur les caractéristiques et les préférences de l'utilisateur, décrites par le quatrième modèle cité, pour proposer des adaptations portant sur les dimensions associées aux trois autres modèles. Ainsi, ces méthodes autorisent, à des degrés divers, des adaptations du contenu délivré, de la présentation et/ou des la structure de l'hypermédia et des possibilités de navigation dans celui-ci.

Néanmoins, quelles que soient ces adaptations, il s'avère qu'aucune proposition ne cherche à apporter de réponse pour contrer les risques de perte dans l'hyperespace et de surcharge cognitive auxquels peut être confronté l'utilisateur.

Notre travail repose plus particulièrement sur la proposition et la mise en œuvre d'une notion contribuant à l'adaptabilité dans les SIW d'une part, et contribuant à limiter les risques de désorientation et de surcharge cognitive d'autre part. Nous avons ainsi montré comment la notion d'*Accès Progressif*, intégrée à tous les niveaux de la conception, autorise la mise œuvre de systèmes capables de s'adapter à chaque utilisateur :

- d'un point de vue *fonctionnel* en donnant accès aux fonctionnalités qui sont associées au(x) rôle(s) joué(s) par l'utilisateur dans le système (au sens d'acteur dans UML). Ces fonctionnalités mettent en œuvre l'accès progressif en délivrant une plus ou moins grande quantité d'informations. Elles sont par ailleurs elles-mêmes accessibles de façon progressive.
- d'un point de vue *hypermédia* en construisant automatiquement un hypermédia dont la structure, les possibilités de navigation et l'apparence des pages correspondent aux attentes de l'utilisateur. La structure de l'hypermédia se présente sous une forme arborescente extensible en largeur et en profondeur par le jeu des mécanismes d'accès progressif.

L'accès progressif constitue en lui-même une dimension d'adaptation. D'une part, il l'est dans la réponse qu'il apporte aux besoins d'accès à l'information. Il permet aux utilisateurs de gérer leur propre espace d'information en distinguant ce à quoi ils souhaitent accéder de façon immédiate, et ce qui peut être atteint si besoin, de façon progressive. D'autre part, l'accès progressif est proposé dans une logique d'adaptation puisque chaque stratification peut être personnelle.

Nous résumons les différentes contributions de notre travail.

➤ *La définition, la modélisation et la mise en œuvre de la notion d'Accès Progressif.*

La notion d'Accès Progressif repose sur l'hypothèse selon laquelle l'utilisateur n'a pas *tout le temps* besoin d'accéder à *toutes* les informations et à *toutes* les fonctionnalités du SIW qui constituent les données de l'hypermédia. Nous proposons une organisation des informations et des fonctionnalités du SIW en adéquation avec cette hypothèse. Cette organisation, appelée *stratification*, repose sur une vision ensembliste des données (informations ou fonctionnalités) telle que :

- Au premier niveau de détail de la stratification, l'utilisateur accède de façon immédiate aux données les plus pertinentes pour lui ;
- chacun des niveaux de détail suivants de la stratification est associé à des données supplémentaires ;
- l'utilisateur contrôle à tout moment la quantité de données qu'il souhaite voir grâce à des mécanismes de *masquage* et de *dévoilement* permettant la navigation entre les niveaux de détails. Plus il accède à un niveau de détail élevé, plus un grand nombre de données lui est proposé.

Nous avons formalisé cette proposition dans le *Modèle d'Accès Progressif* (MAP). Ce modèle décrit les concepts et les mécanismes nécessaires à la mise en œuvre de l'Accès Progressif. La formalisation que nous avons présentée, parce qu'indépendante à la fois du domaine d'application et du paradigme utilisé pour représenter ce domaine, fait du MAP un modèle générique réutilisable et applicable à divers domaines exprimés selon différents formalismes (E/R, objet, XML, etc.)

Le MAP définit le cadre de référence dans lequel s'inscrit notre approche pour la conception de SIW.

➤ *La description de quatre modèles pour la conception de SIW à Accès Progressif.*

Les modèles que nous décrivons correspondent à ceux proposés dans la plupart des approches de conception d'applications Web. Néanmoins, la particularité de nos modèles est qu'ils appliquent tous les concepts et principes de l'Accès Progressif à des niveaux différents qui correspondent aux principales dimensions d'un SIW :

- Le *Modèle du Domaine* décrit les objets du monde réel qui constituent le domaine d'application manipulé par le SIW. Ce modèle fait l'objet d'une application des principes du MAP qui permet de proposer un Accès Progressif aux données qu'il contient.
- Le *Modèle des Fonctionnalités* décrit les tâches offertes par le SIW (i.e. consultation, modification, ajout d'informations, etc.) qui portent sur les données du Modèle du Domaine. Il met en œuvre une organisation de la dimension fonctionnelle du SIW qui repose sur trois niveaux de granularités: l'Espace Fonctionnel est composé de Rôles Fonctionnels, eux-mêmes étant composés de Fonctionnalités. Le Modèle des Fonctionnalités applique également les principes du MAP. A chaque niveau de granularité, les composants les plus pertinents sont accessibles immédiatement, les autres le sont de façon progressive grâce aux mécanismes de l'Accès Progressif.
- Le *Modèle de l'Hypermédia* permet de construire l'hypermédia du SIW à partir des spécifications du Modèle des Fonctionnalités. Une représentation logique de sa structure et de ses liens de navigation est proposée par le sous-modèle de Structuration & Navigation qui rend compte de l'organisation fonctionnelle du modèle précédent. Le Modèle Hypermédia est complété par le second sous-modèle de Présentation qui définit la composition des pages de l'hypermédia et leur apparence au moyen de chartes (dites de composition et graphiques) assurant la cohésion des pages générées dynamiquement.
- Le *Modèle des Utilisateurs* entretient des relations avec les autres modèles proposés, garantissant ainsi la conception de SIW capables de s'adapter à leurs utilisateurs de différentes façons. Ainsi, sa connexion avec le Modèle des Fonctionnalités assure un *Accès Progressif personnalisé* à chacun des niveaux de granularité de l'organisation fonctionnelle. Ces relations avec le Modèle de l'Hypermédia garantissent la personnalisation à la fois de la structure, des possibilités de navigation et de l'apparence des pages de l'hypermédia.

➤ *La proposition d'une démarche pour la conception de SIW à Accès Progressif.*

Nous avons décrit une démarche de conception qui guide le concepteur, étape par étape, dans la définition des modèles ci-dessus. Cette démarche est majoritairement incrémentale, dans la mesure où les spécifications d'un modèle reposent souvent sur celles du modèle précédent. Une étape particulière dans cette démarche est proposée pour la spécification des attentes des utilisateurs en matière d'Accès Progressif. Reposant sur des cas d'utilisation dont nous avons étendu les notations, cette étape permet aux utilisateurs d'exprimer ce à quoi qu'il souhaite accéder de façon immédiate d'une part, et de façon progressive d'autre part. Ces spécifications permettent aux concepteurs de SIW adaptables de décrire les modalités de l'Accès Progressif.

➤ *La réalisation d'un prototype intégrant et validant l'approche par Accès Progressif.*

L'environnement KIWIS (acronyme de *Knowledge for Improving Web Information Systems*) a été présenté. Il consiste en une plate-forme dédiée au concepteur de SIW. KIWIS implémente

la démarche de conception et prend en charge la génération des SIW à Accès Progressif conçus. De plus, KIWIS constitue l'environnement d'exécution de tels SIW.

Perspectives

Les fondements de l'Accès Progressif dans les SIW adaptables à l'utilisateur étant posés et validés par notre proposition, plusieurs voies de recherche ou de développement méritent, selon nous, d'être à présent poursuivies. Nous identifions plusieurs perspectives, à court et à plus long termes.

Amélioration du prototype KIWIS

La version du prototype KIWIS présentée dans ce mémoire a permis de démontrer la faisabilité de notre approche. Cet environnement, et notamment la plate-forme de conception, est en cours d'amélioration.

Nous avons mentionné le couplage de KIWIS avec le Système de Représentation de Connaissances par Objet AROM. L'éditeur fourni par AROM permet à ce jour de procéder à la modélisation et à l'instanciation du Modèle du Domaine dans un environnement graphique. Nous souhaitons proposer au concepteur la possibilité de définir, avec la même facilité, les stratifications en vue de l'accès progressif. Dans ce but, il nous paraît utile de proposer des notations graphiques spécifiques qui puissent être utilisées *via* l'IME d'AROM.

De plus des améliorations sont en cours en ce qui concerne les interfaces de définition du Modèle de l'Hypermédia. Ces améliorations portent essentiellement sur la mise à disposition du concepteur d'outils permettant de définir les pages types et les présentations types plus finement que cela est possible actuellement.

Enfin, l'amélioration du prototype passe également par le développement d'un gestionnaire dédié à la sécurité, et de façon corrélée, par une réflexion menée au niveau conceptuel sur cet aspect. KIWIS garantit, dans une certaine mesure, la sécurité sur les données et les fonctionnalités accessibles aux utilisateurs (concepteurs de SIW et utilisateurs finaux de ces systèmes). Le Modèle des Utilisateurs fournit les informations minimales permettant d'authentifier les utilisateurs. *Via* les relations qu'entretient ce modèle avec les autres modèles de l'approche, les données et les fonctionnalités délivrées par le système correspondent aux droits des utilisateurs. La possibilité de masquer complètement certaines informations est, de plus, offerte par l'Accès Progressif puisqu'il est possible de masquer totalement certaines informations. Néanmoins, actuellement, rien ne permet d'assurer la cohérence en termes de confidentialité des données. Ainsi, il est possible que l'utilisateur accède, par le biais d'une fonctionnalité, à une information spécifiée par le concepteur comme confidentielle dans le cadre d'une autre fonctionnalité (masquage total de cette information dans la stratification associée à cette seconde fonctionnalité). Ces possibilités de contournement de ces aspects relatifs à la sécurité doivent pouvoir être éliminées par l'intégration, dans le Modèle des Fonctionnalités, des possibilités d'expressions offertes par un modèle tel que RBAC (*Role Based Control Access*) [Ferr01]. La gestion des droits sur les données est gérée à un bas niveau dans RBAC grâce à la notion de *permission* qui associe une opération (lire, écrire, etc.) à un objet dont la granularité peut être très fine (un attribut par exemple). En considérant une fonctionnalité comme un ensemble de permissions, il devient possible de maintenir une représentation de toute permission définie, dans le cadre de cette fonctionnalité, pour un utilisateur et une information particulière. Une telle représentation peut alors être exploitée pour contrôler la cohérence des permissions définies par les différentes fonctionnalités accessibles par un même utilisateur. Une première

étude dans ce sens a été menée et appliquée aux fichiers XML qui contiennent les données stratifiées en vue de l'accès progressif dans KIWIS [Adda02].

Capacités d'Adaptation avancées pour l'Accès Progressif

Une perspective vers laquelle nous souhaitons orienter nos travaux concerne l'étude de capacités d'adaptation avancées pour l'Accès Progressif. Deux axes nous paraissent intéressants dans ce domaine.

Le premier axe consiste à poursuivre nos travaux sur la détection du comportement telle que nous l'avons introduite dans le chapitre 6. Nous rappelons que l'objectif visé est de recueillir des informations à partir des actions de l'utilisateur (notamment concernant le masquage et le dévoilement) afin de procéder à une réorganisation automatique des stratifications. Le protocole d'exploitation du MAP offre un ensemble d'opérations permettant de procéder à la réorganisation manuelle des stratifications. L'adaptativité du système, en tant que processus d'adaptation automatique et contrôlée par le système, nécessite :

- d'une part, de s'intéresser à la définition d'un formalisme pour la représentation des comportements des utilisateurs. Nous pensons baser celui-ci sur des propositions proches de celles décrites dans UWE [Koch00] qui reposent sur un profil UML et seraient ainsi compatibles avec les termes du MAP.
- d'autre part, de définir un ensemble de règles d'adaptabilité permettant de décider des modifications à apporter aux stratifications en fonction des comportements observés. Des seuils de décision doivent être spécifiés afin de maintenir une relative stabilité du système : l'observation ponctuelle d'un comportement ne doit pas induire une modification systématique des stratifications (ce qui rendrait le système inutilisable). Les travaux de De Bra & al. autour du modèle AHAM (Adaptive Hypermedia Model) abordent par exemple ces aspects [DeBr99].

Le deuxième axe concerne la prise en compte des conditions d'exploitation du SIW qui doit permettre d'exploiter pleinement les principes de l'Accès Progressif. Par *conditions d'exploitation*, nous faisons référence à deux aspects.

1. Les *conditions matérielles* d'utilisation du SIW méritent, selon nous, d'être considérées. Les principes de l'Accès Progressif peuvent, dans certaines situations, réduire les inconvénients liés aux contraintes du dispositif utilisé pour accéder au système. La détection de caractéristiques matérielles peut fournir une information à prendre en compte dans l'exploitation des stratifications. Par exemple, dans le cas d'un utilisateur accédant au SIW *via* son organisateur de poche (*Personal Digital Assistant – PDA*), il peut être intéressant de limiter l'accès aux premiers, voire uniquement au premier niveau de détail. Des solutions commerciales existent pour adapter le contenu du Web au format PDA (*cf.* par exemple www.avantgo.com). Néanmoins, elles sont basées sur la création *ad hoc* de pages Web conçues spécialement pour les PDA. Ceci oblige une redéfinition de la même page pour un dispositif d'accès différent utilisé par le même utilisateur. Nous pensons que l'Accès Progressif peut être à la base d'un processus plus générique.

2. Les conditions d'exploitation peuvent être comprises comme les *conditions non-matérielles* dans lesquelles l'utilisateur accède au SIW. Nous pensons par exemple à des contraintes d'ordre temporel que l'utilisateur pourrait avoir. La connaissance du temps que l'utilisateur souhaite consacrer à l'utilisation du SIW, peut également constituer une information pertinente pour l'Accès Progressif. Un travail que nous avons initié sur l'Accès Progressif aux données multimédias prend ici tout son sens. L'Accès Progressif à ce type de données repose sur la notion de *format multiple de données multimédia* que nous envisageons comme l'équivalent dans ce domaine des Représentations d'Entité Masquable proposées par le MAP. Ainsi,

l'expression d'un accès progressif à une donnée telle qu'une vidéo s'exprime par la possibilité de voir soit les méta-données (auteur, date, sujet, etc.), soit un extrait, soit un résumé de cette vidéo, déterminés et calculés sur la base du temps que souhaite consacrer l'utilisateur.

BIBLIOGRAPHIE

- [Abit91] Abiteboul S. & Bonner A., Objects and Views. ACM SIGMOD Conference on Management of Data, 1991, pp. 238-247.
- [Abit99] Abiteboul S., On Views and XML, Proceedings of ACM Symposium on Principles of Database Systems, 1999, pp. 1-9.
- [Adda02] Adda M., Modèle de sécurité pour les documents XML : Adaptation au système KIWIS , Mémoire de DEA Informatique : Systèmes et Communications, Université Joseph Fourier, Grenoble, 19 juin 2002.
- [Bard98] Bardou D., Étude des langages à prototypes, du mécanisme de délégation, et de son rapport à la notion de point de vue, Thèse de Doctorat, Université Montpellier II, avril 1998.
- [Alle83] Allen J.F., Maintaining Knowledge about Temporal Intervals, Communications of the ACM, 26(11), 1983, pp. 832-843.
- [Bala95] Balabanovic, Shoham, Yun, An adaptative agent for automated Web Browsing, Technical Report CS-TN 97-52, Stanford University, Palo Alto, CA, 1995.
- [Bare00] Baresi L., Garzotto F. & Paolini P., From Web Sites to Web Applications: New Issues for Conceptual Modeling. In Proceedings of the International Workshop on The World Wide Web and Conceptual Modeling, co-located with the 19th International Conference on Conceptual Modeling, Salt Lake City (USA), October 2000.
- [Baum99] Baumeister H., Koch N. & Mandel L., Towards a UML extension for hypermedia design. In France R. & Rumpe B. (eds), UML'99 – The Unified Modeling Language - Beyond the Standard, LNCS 1723, Springer Verlag, pp. 614-629, 1999.
- [BenA97] Ben Abdallah, Nabil. Analyse et structuration de documents scientifiques pour un accès personnalisé à l'information utile : vers un système d'information évolué, Thèse de Doctorat, Université Claude-Bernard, Lyon 1, 7 juillet 1997
- [Bieb97] Bieber M. & Vitali F., Toward Support for Hypermedia on the World Wide Web, *IEEE Computer*, 30(1), January 1997.
- [Boch99] Bochicchio M., Paiano R. & Paolini P., JWeb: An HDM Environment for Fast Development of Web Applications, IEEE International Conference on Multimedia Computing and Systems, Volume II, June 7-11, Florence, Italy, 1999, pp. 453-460.
- [Boda83] Bodart F. & Pigneur Y., Conception assistée des applications informatiques, Masson, Paris, 1983.
- [Boll99] Bollacker K.D., Lawrence S., Giles C.L., A system for Automatic Personalized Tracking of Scientific Literature on the Web, in Proceeding of the 4th ACM Conference on Digital Library, Berkeley, CA, August 11-14, 1999.
- [Booc99] Booch G., Rumbaugh J., Jacobson I., The UML user guide, Addison-Wesley, 1999.
- [Brow90] Browne D., Totterdell P. & Norman M., (eds.), Adaptive User Interfaces, Computer and People Series, Academic Press, 1990.
- [Brus97] Brusilovsky P., Efficient techniques for adaptive hypermedia, In Nicholas C. & Mayfield J. (Eds.), Intelligent hypertext: Advanced techniques for the World Wide Web, LNCS 1326, Berlin, Springer-Verlag, 1997, pp. 12-30.
- [Brus98] Brusilovsky P., Methods and Techniques of Adaptive Hypermedia. In Brusilovsky P., Kobsa A. & Vassileva J. (eds.), Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, 1998, pp. 1-43.
- Note: L'ouvrage cité édite une sélection d'articles déjà publiés dans International Journal of User Modeling and User-Adapted Interaction, Special Issue on Adaptive Hypertext and Hypermedia, 6(2-3), Kluwer Academic Publishers, 1996.*
- [Bush45] Bush V. As We May Think, in Atlantic Monthly, 176(1), July, 1945, pp. 101-108. Accessible en ligne à l'adresse : <http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm>
-

- [Card91] Cardelli L. & Mitchell J., Operations on Records, *Mathematical Structures in Computer Sciences*, Volume 1, 1991, pp3-48.
- [Cann01a] Cannataro M., Cuzzocrea A. & Pugliese A., A probabilistic adaptive hypermedia system, 1st International Workshop on Web Dynamics, In conjunction with the 8th International Conference on Database Theory, London, UK, 3 January 2001
- [Cann01b] Cannataro M., Cuzzocrea A. & Pugliese A., A Multidimensional Approach for Modelling and Supporting Adaptive Hypermedia Systems, In Bauknecht K., Madria S.K., Pernul G. (Eds.), *Proceedings of the 2nd International Conference on Electronic Commerce and Web Technologies (EC-Web 2001)*, Munich, Germany, September 4-6, 2001, LNCS 2115, Springer, 2001, pp. 132-141.
- [Cann02a] Cannataro M. & Pugliese A., XAHM: An XML-Based Adaptive Hypermedia Model and Its Implementation, In Reich S., Tzagarakis M. & De Bra P. (Eds.): *Hypermedia: Openness, Structural Awareness, and Adaptivity*, International Workshops OHS-7, SC-3, and AH-3, Aarhus, Denmark, August 14-18, 2001, Revised Papers, LNCS 2266, Springer, 2002, pp. 252-263.
- [Cann02b] Cannataro M., Cuzzocrea A., Mastroianni C., Ortale R. & Pugliese A., Modeling Adaptive Hypermedia with an Object-Oriented Approach and XML, 2nd International Workshop on Web Dynamics, in conjunction with WWW 2002, Honolulu, Hawaii, 7th May 2002.
- [Carn94] Carneiro L.M.F., Coffin M.H., Cowan D.D. & Lucena C.J.P.L., ADVCharts: a Visual Formalism for Highly Interactive Systems, in Harrison M.D. & Johnson C. (eds), *Software Engineering in Human-Computer Interaction*, Cambridge University Press, 1994.
- [Caro95] Carroll J.M, *Scenario-based Design: Envisioning Work and Technology in System Development*, John Wiley & Sons, 1995.
- [Cast01] Casteleyn, S. & De Troyer, O., Structuring Web Sites Using Audience Class Hierarchies, in *Proceedings of DASWIS 2001 Workshop*, in conjunction with 20th International Conference on Conceptual Modeling ER 2001, Yokohama, Japan, November 29-31, 2001.
- [Ceri00a] Ceri S., Fraternali P. & Bongio A., Web Modeling Language (WebML): a modeling language for designing Web sites, *Proceedings of the 9th International World Wide Web Conference (WWW9)*, Amsterdam, Netherlands, May 15 - 19, 2000.
- [Ceri00b] Ceri S., Fraternali P., Bongio A. & Maurino A., Modeling data entry and operations in WebML WebDB 2000. Dallas, 2000.
- [Ceri01] Ceri S., Fraternali P., Bongio A. & Comai S., WebML User Guide, Politecnico di Milano, 2001. disponible à l'adresse : http://Webml.elet.polimi.it/Webml/manuals/WebML_User_Guide.pdf
- [Ceri02] Ceri S., Fraternali P. & Matera M., Conceptual modeling of data-intensive Web applications. *IEEE Internet Computing*, à paraître en 2002.
- [Chan99] Chan P.K., A non-invasive approach to building Web user profiles, *Workshop on Web Usage Analysis and User Profiling*, 5th International Conference on Knowledge Discovery & Data Mining, San Diego, CA, August 15-18, 1999.
- [Chen76] Chen P.P., The Entity-Relationship Model - Towards a Unified View of Data, *ACM Transactions on Database Systems*, 1(1), March, 1976, pp. 471-522.
- [Chen00] Chen Q., Norcio A.F. & Wang J., Neural Network Based Stereotyping for User Profiles, *Neural Computing and Applications*, 9, Springer Verlag, London, 2000, pp. 259-265.
- [Cing00] Cingil I., Dogac A. & Azgin A., A broader approach to personalization, *Communication of the ACM*, 43(8), August 2000, pp. 136-141.
- [Cona00] Conallen J., *Concevoir des applications Web avec UML*, Editions Eyrolles, 2000.
- [Conk87] Conklin J., Hypertext: An introduction and survey. *IEEE Computer* 20(9), 1987, pp. 17-41.
- [Cowa95] Cowan D.D. & Lucena C.J.P., Abstract Data Views, an Interface Specification Concept to Enhance Design for Reuse, *IEEE Transactions on Software Engineering*, 21(3), March 1995.
- [Dani86] Daniels J.P., Cognitive Models in Information Retrieval – An Evaluation Review, *Journal of Documentation*, 42(4), December 1986, pp. 272-304.

- [DCMI99] Dublin Core Metadata Initiative, Dublin Core Metadata Element Set, Version 1.1: Reference Description, DCMI Recommendations, <http://dublincore.org/documents/1999/07/02/dces/>, July 02, 1999.
- [DeBr99] De Bra P., Houben G.J. & Wu H., AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In Proceedings of the 10th ACM Conference on Hypertext and Hypermedia, ACM Press, 1999, pp. 147–156.
- [Denn98] Dennis A.R., Lessons from Three Years of Web Development, *Communications of the ACM*, 41(7), July 1998, pp. 112-113.
- [DeTr98] De Troyer O.M.F., Leune C.J., WSDM : a User-Centered Design Method for Web Sites, Proceedings of the 7th International World Wide Web Conference (WWW7), *Computer Networks*, 30(1-7), April, 1998, pp. 85-94.
- [DeTr01] De Troyer, O. & Casteleyn, S., The Conference Review System with WSDM, 1st International Workshop on Web-Oriented Software Technology (IWWOST'01), Valencia, Spain, June, 18-20, 2001. <http://www.dsic.upv.es/~west2001/iwwost01/>
- [Diaz95] Diaz A. & Isakowitz T., 'RMCASE: Computer-Aided Support for Hypermedia Design and Development', International Workshop on Hypermedia Design, 1995.
- [Diet93] Dieterich H., Malinowski U., Khme T. & Schneider-Hufschmidt M., State of the Art in Adaptive User Interfaces, In Schneider-Hufschmidt M., Khme T. & Malinowski U. (Eds.): *Adaptive User Interfaces: Principle and Practice*, Amsterdam, North Holland, 1993.
- [Eyme02] Eymery C., Génération de systèmes d'information multimédia adaptables et basés sur le Web, Mémoire de Diplôme d'Ingénieur, en Informatique, Conservatoire National des Arts et Métiers, Centre d'enseignement de Grenoble, 16 janvier 2002.
- [Ferr01] Ferraiolo D.F., Sandhu R., Gavrila S., Kuhn D.R. & Chandramouli R., Proposed NIST Standard for Role-Based Access Control, *ACM Transactions on Information and Systems Security*, 4(3), August 2001
- [Fink00] Fink J. & Kobsa A., A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web, *International Journal of User Modeling and User-Adapted Interaction*, 10, Kluwer Academic Publishers, 2000, pp. 209-249.
- [Fras01] Frasinca F., Houben G-J. & Vdovjak R., An RMM-Based Methodology for Hypermedia Presentation Design, Proceedings of the 5th East European Conference on Advances in Databases and Information Systems (ADBIS 2001), LNCS 2151, Vilnius, Lithuania, September 25-28, 2001, pp. 323-337.
- [Fras02] Frasinca F. & Houben G-J, Hypermedia Presentation Adaptation on the Semantic Web, in de Bra P., Brusilovsky P. & Conejo R., (Eds.), Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002), Malaga, Spain, May 29 - 31, 2002, LNCS 2347, 2002, pp. 133-142.
- [Frat98] Fraternali P., Paolini P., A Conceptual Model and a Tool Environment for Developing More Scalable and Dynamic Web Applications, in Proceedings of EDBT'98, Valencia, Spain, March 1998, pp. 421-435.
- [Frat99] Fraternali P., Tools and Approaches for Developing Data-Intensive Web Applications: A Survey, *ACM Computing Surveys*, 31(3), September 1999, pp. 227-263.
- [Gabb98] Gabbrielli M., Marchiori M. & de Boer F.S., Dynamic Web sites, The Query Languages Workshop (QL'98), Boston, Massachusetts, December 3-4, 1998.
- [Gamm95] Gamma E., Helm R., Johnson R. & Vlissides J., *Design Patterns: Elements of reusable object-oriented software*, Addison-Wesley, 1995.
- [Garz93a] Garzotto F., Paolini P. & Schwabe D., HDM: A model-based approach to hypertext application design, *ACM Transactions on Information Systems*, 11(1), pp. 1-26, January 1993.
- [Garz93b] Garzotto F., Mainetti L., & Paolini P. HDM2: Extending the E-R Approach to Hypermedia Application Design, in Elmasri R., Kouramajian V., & Thalheim B. (Eds), Proceedings of the 12th International Conference on the Entity-Relation Approach (ER'93) , Arlington, TX, December 1993, pp. 178-189.

- [Garz01] Garzotto F., Paolini P., & Baresi L., Supporting Reusable Web Design with HDM-Edit. In Proceedings of the 34th Hawaii International Conference on System Sciences, Internet and the Digital Economy track, Web Engineering minitrack, Maui (USA), January 2001.
- [Gnah99] Gnaho C. & Larcher F., A goal Driven Approach to Web Navigation Modeling, in Proceedings of the Workshop on Internet Data Management, Florence, Italy, 1999.
- [Gnah00] Gnaho C., Définition d'un cadre méthodologique pour l'ingénierie des Systèmes d'Information, Web Adaptatifs, Thèse de Doctorat, Université Paris I, soutenue le 17 octobre 2000.
- [Gnah01] Gnaho C., Web-based Informations Systems Development – A User Centered Engineering Approach, in Murugesan S. & Deshpande Y. (Eds.), Web Engineering: Managing Diversity and Complexity of Web Application Development, LNCS 2016, 2001, pp. 105-118.
- [Goed98] Goedefroy W., Meersman R. & De Troyer O., UR-WSDM: Adding User Requirement Granularity to Model Web Based Information Systems, in Proceedings of the 1st Workshop on Hypermedia Development, Pittsburgh, USA, June 20-24, 1998.
- [Gonc00] Goncalves G. & Hémerly F., Des cas d'utilisation en UML à la gestion de rôles dans un système d'information, Actes du Congrès Inforsid, Lyon, 16-19 mai, 2000, pp. 367-379.
- [Gonc01] Goncalves G. & Hémerly F., Une plate-forme UML-XML pour la gestion des rôles d'un système d'information, Actes du Congrès Inforsid, Genève, Suisse, 30 mai - 1 juin, 2001, pp. 111-122.
- [Gros97] Grosz G., Rolland C., Schwer S., Souveyet C., Plihon V., Si-said S., Ben Achour C. & Gnaho C., Modelling and Engineering the Requirements Engineering Process : An Overview of the NATURE Approach, in Requirements Engineering Journal, 2, 1997, pp. 115-131.
- [Güel00] Güell N., Schwabe D. & Vilain P., Modeling Interactions and Navigation in Web Applications, in Proceedings of the World Wild Web and Conceptual Modeling Workshop, ER'00 Conference, LNCS1921, Springer, Salt Lake City, 2000.
- [Hans01] Hansen S., Deshpande Y. & Murugusan S., A Skills Hierarchy for Web-Based Systems Development, in Murugesan S. & Deshpande Y. (Eds.), Web Engineering: Managing Diversity and Complexity of Web Application Development, LNCS 2016, 2001, pp. 223-235.
- [Henn00] Hennicker R. & Koch N., A UML-based Methodology for Hypermedia Design. In Evans A., Stuart S. & Selic B. (eds), UML'2000 - The Unified Modeling Language - Advancing the Standard, LNCS 1939, Springer Verlag, York, England, October 2000.
- [Henz99] Henze N. & Nejd X., Bayesian Modeling for Adaptive Hypermedia Systems. Technical Report, University of Hannover, July 1999.
- [Houb00] Houben G.J., Hera: Automatically Generating Hypermedia Front Ends for Ad Hoc Data from Heterogeneous and Legacy Information Systems, in Proceedings of the 3rd International Workshop on Engineering Federated Information Systems, Aka and IOS Press, 2000, pp. 81–88.
- [IFIP96] IFIP WG2.7, Design Principles for Interactive Software, Chapman & Hall, 1996.
- [Isak95] Isakowitz T., Stohr A. & Balasubramanian E., RMM: A methodology for structured hypermedia design. Communications of the ACM 38(8), pp. 34-44, August 1995.
- [Isak97a] Isakowitz T., Kamis A. & Koufaris M., Extending the capabilities of RMM: Russian Dolls and Hypertext, Proceedings of the 30th Hawaii International Conference on System Sciences (HICSS-30), 1997.
- [Isak97b] Isakowitz T., Kamis A. & Koufaris M., Reconciling Top-Down and Bottom-Up Design Approaches in RMM, Proceedings of WITS-97, Atlanta, GA, USA, December 1997.
- [Isak98a] Isakovitz T., Bieber M. & Vitali F., Web Information Systems, Communications of the ACM, 41(7), July 1998, pp. 78-80.
- [Isak98b] Isakowitz T., Kamis A. & Koufaris M., The Extended RMM Methodology for Web Publishing, Working Paper IS-98-18, Center for Research on Information Systems, 1998.
- [Jaco92] Jacobson I., Christerson M., Jonsson P. & Övergaard G., Object-Oriented Software Engineering: a use case driven approach, Addison-Wesley, 1992.
- [Jaco99] Jacobson I., Booch G. & Rumbaugh J., The Unified Software Development Process, Addison-Wesley, 1999.

- [Kapl01] Kaplan A. & Lunn J., FlexXML: Engineering a More Flexible and Adaptable Web, in Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'01), 2-4 April, Las Vegas, NV, 2001, pp. 405-410.
- [Kapp00] Kappel G., Retschitzegger W., & Schwinger W., Modeling Customizable Web Applications - A Requirement's Perspective, in Proceedings of the International Conference on Digital Libraries, (ICDL 2000), Kyoto, Japan, November 13-16, 2000.
- [Klyn01] Klyne G., Reynolds F., Woodrow C. & Ohto H., Composite Capability/ Preference Profile (CC/PP): Structure and Vocabularies, W3C, 2001, <http://www.w3.org/TR/CCPP-struct-vocab>
- [Kobs01a] Kobsa A., Koenemann J. & Pohl W., Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships, The Knowledge Engineering Review, 16(2), Cambridge University Press, UK, 2001, pp. 111-155.
- [Kobs01b] Kobsa A., Generic User Modeling Systems, International Journal of User Modeling and User-Adapted Interaction, 11, Kluwer Academic Publishers, 2001, pp. 49-63.
- [Koch99] Koch N., A Comparative Study of Methods for Hypermedia Development, Technical Report 9905, Ludwig-Maximilians-Universität München, November 1999.
- [Koch00] Koch N., Software Engineering for Adaptive Hypermedia Systems – Reference Model, Modelling Techniques and Development Process, Ph.D Thesis, Fakultät der Mathematik und Informatik, Ludwig-Maximilians-Universität München, December 2000.
- [Kram00] Kramer J., Noronha S. & Vergo J., A user-centered design approach to personalization, Communication of the ACM, 43(8), August 2000, pp. 44 – 48.
- [Kruc00] Kruchten P., The Rational Unified Process: An Introduction (Second Edition), Addison-Wesley, 2000.
- [Lain99] Lainé-Cruzel S., ProfilDoc : filtrer une information exploitable, Bulletin des bibliothèques de France, 44 (5), 1999, p 60-65.
- [Lang01a] Lang M., A Study of Practice in Hypermedia Systems Design, Doctoral Consortium, European Conference on Information Systems, Bled, Slovenia, 24-26 June, 2001.
- [Lang01b] Lang M., Issues and Challenges in the Development of Hypermedia Information Systems, in Hackney R. (ed.), Proceedings of the 11th Annual Business Information Technology Conference (BIT2001), Manchester, England, 30-31 October, 2001.
- [Lieb95] Lieberman H., Letizia: An agent that assists Web Browsing. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 20-25, 1995, pp. 924-929.
- [LeMo77] Le Moigne J.L., La théorie du système général, Presse Universitaire de France, 1977.
- [LMWG00] Learner Model Working Group (IEEE Learning Technology Standards Committee), PAPI Learner Draft 7 Specification, November 28, 2000. <http://ltsc.ieee.org/wg2/>
- [LTSC02] IEEE Learning Technology Standards Committee, Learning Object Metadata Standard, June 13, 2002. <http://ltsc.ieee.org/wg12/>
- [Mart00] Martin H. & Lozano R., Dynamic Generation of video abstracts using an object-oriented video DBMS, Networking and Information Systems Journal, Special Issue on Video Data 3(1), Hermes, 2000, pp. 53-75.
- [Mecc99] Mecca G., Merialdo P., Atzeni P. & Crescenzi V., The ARANEUS Guide to Web-Site Development, ARANEUS Project Working Report, AWR-1-99, March 1999.
- [Moba00] Mobasher B., Cooley R. & Srivastava J., Automatic Personalization Based on Web Usage Mining, Communications of the ACM, 43(8), August 2000, pp. 142-151.
- [Mots99] Motschnig-Pitrik R., Contexts and Views in Object-Oriented Languages, in P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, F. Castellan (Eds.): Modeling and Using Context, Proc of the 2nd International and Interdisciplinary Conference, CONTEXT'99, Trento, Italy, 1999.
- [Mouk97] Moukas A., Amalthea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem', in Applied Artificial Intelligence: An International Journal, 11(5), 1997, pp. 437-457.
- [Mull99] Muller P-A., Modélisation objet avec UML, Editions Eyrolles, 1999.

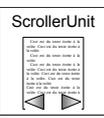
- [Muru01] Murugesan S., Deshpande Y., Hansen S., Ginige A., Web Engineering: A New Discipline for Development of Web-Based Systems, in Murugesan S. & Deshpande Y. (Eds.), Web Engineering: Managing Diversity and Complexity of Web Application Development, LNCS 2016, 2001, pp. 3-13.
- [Nels65] Nelson T., A File Structure for the Complex, the Changing, and the Indeterminate, ACM 20th National Conference, New York, 1965, pp. 84-100.
- [OMG01] Object Management Group, Unified Modeling Language Specifications, Version 1.4, September 2001. <http://www.omg.org/cgi-bin/doc?formal/01-09-67.pdf>
- [Oppe94] Oppermann R., Introduction, in Oppermann R. (Ed), Adaptive User Support, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1994, pp. 1-13.
- [Oppe97] Oppermann R., Rashev R. & Kinshuk, Adaptability and Adaptivity in Learning Systems, in Behrooz A. (Ed), Knowledge Transfer (Volume II), Proceedings on Knowledge Transfer, July 14 - 16, 1997, London UK, Pace, London, 1997, pp. 173-179.
- [Page00] Page M., Gensel J., Capponi C., Bruley C., Genoud P. & Ziébelin D., Représentation de connaissances au moyen de classes et d'associations : le système AROM, Actes Langages et Modèles à Objets (LMO'00), Mont Saint-Hilaire, Canada, 25-28 janvier, 2000, pp. 91-106.
- [Pate99] Paterno P. & Mancini C., Designing Web User Interfaces for Museum Applications to Support different Types of Users, in Proceedings of International Conference about Museums and the Web, March 12-14, New Orleans, LA, 1999, pp.75-86.
- [Pazz96] Pazzani M., Muramatsu J. & Billsus D., Syskill & Webert : Identifying Interesting Web Sites. In Proceedings of the National Conference on Artificial Intelligence, Portland, OR, 1996, pp. 54-61.
- [Prat97] Prat N., Une approche linguistique pour la formalisation et la classification des buts en ingénierie des processus, 1st International Workshop on the Many Facets of Process Engineering (MFPE'97), Gammarth, Tunisie, Septembre 1997.
- [Pret99] Pretschner A. & Gauch S., Ontology Based Personalized Search, in Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, Chicago, November 1999, pp. 391-398.
- [Raad02] Raad H. & Causse B., Modelling of an Adaptive Hypermedia System Based on Active Rules, in Cerri S.A., Gouardères G. & Paraguaçu F. (Eds.), Proceedings of the 6th International Conference on Intelligent Tutoring Systems (ITS 2002), Biarritz, France and San Sebastian, Spain, June 2-7, 2002, LNCS 2363, 2002, pp. 149-157.
- [Rich99] Rich E., Users are individuals: individualizing user models, Int Journal on Human-Computer Studies, 51, 1999 (Ré-édition d'un article de 1981), pp. 323-338.
- [Roll88] Rolland C., Foucault O. & Benci G., Conception des Systèmes d'Information – La méthode REMORA, Eyrolles, 1988.
- [Roll00] Rolland C., Information Systems and Web Information Systems : A Methodological Perspective, Proceedings International Forum cum Conference on Information Technology and Communication at the Dawn of the New Millennium, Bangkok, Thailand, August 2000.
- [Rose01] Rosenberg M., The personalization story, ITworld.com, 11th may 2001, disponible à l'adresse : <http://www.itworld.com/Man/2676/ITW010511rosenberg/>
- [Ross99] Rossi G., Schwabe D. & Lyardet F., Web application models are more than conceptual models, in Proceedings of the World Wide Web and Conceptual Modeling'99 Workshop, ER'99 Conference, LNCS 1727, Springer, Paris, 1999, pp. 239-252
- [Ross01] Rossi G., Schwabe D. & Guimarães R., Designing Personalized Web Applications, 10th International World Wide Web Conference (WWW10), May 1-5, Hong Kong, 2001.
- [Rumb99] Rumbaugh J., Jacobson I., & Booch G., The UML reference manual, Object Technology Series, Addison-Wesley, 1999.
- [Schw96] Schwabe D., Rossi G., Barbosa S., Systematic Hypermedia Application Design with OOHD, Proceedings of ACM Hypertext, 1996.
- [Schw98] Schwabe D. & Rossi G., An Object Oriented Approach to Web-Based Application Design, Theory and Practice of Object Systems, 4(4), Wiley and Sons, New York, 1998.

- [Schw99] Schwabe D., Pontes R.A. & Moura I., OOHDM-Web: An Environment for Implementation of Hypermedia Applications in the WWW, SigWEB Newsletter, 8(2), June, 1999.
- [Step98a] Stephanidis C., Paramythis A., Sfyraakis M., Stergiou A., Maou N., Leventis A., Paparoulis G. & Karagiandidis C., Adaptable and adaptive user interfaces for disabled users in AVANTI Project, in Triglia S., Mullery A., Campolargo M., Vanderstraeten H. & Mampaey M. (Eds), Proceedings of the 5th International Conference on Intelligence in Services and Networks (IS&N'98), Technology for Ubiquitous Telecom Services, Antwerp, Belgium, 25-28 May 1998, LNCS 1430, Springer-Verlag, Germany, 1998, pp. 153–166.
- [Step98b] Stephanidis C., Paramythis A., Akoumianakis D. & Sfyraakis M., Self-Adapting Web-based Systems: Towards Universal Accessibility, Proceeding of the 4th Workshop on User Interface For All, Stockholm, Sweden, 19-21 October, 1998.
- [Taka97] Takahashi K., Liang E., Analysis and Design of Web Based Information Systems, Proceedings of the 6th International World Wide Web Conference (WWW6), Santa Clara, California, USA, April 1997.
- [Theo99] Theodorakis M., Analyti A., Constantopoulos P. & Spyrtos N., Contextualization as an Abstraction Mechanism for Conceptual Modelling, Proceedings of the 18th International Conference on Conceptual Modeling, ER'99, Paris, France, November 15-18, 1999.
- [Tong95] Tong- Tong J.-R., La logique floue, Hermès, Paris, 1995.
- [Turk01] Turki S., Introduction aux hyperclasses, Actes du XIXe Congrès Inforsid, Martigny, Suisse, 29 Mai – 1^{er} Juin, 2001, pp. 281-299.
- [Turk02] Turki S. & Léonard M., Hyperclasses: towards a new kind of independence of the methods from the schema, in proceedings of the 4th International Conference on Enterprise Information Systems (ICEIS'2002), Volume 2, Ciudad Real, Spain, April 3-6, 2002, pp. 788-794.
- [Vila00a] Vilain P., Schwabe D. & De Souza C.S., Use Cases and Scenarios in the Conceptual Design of Web Applications, Technical Report MCC 12/00, Departamento de Informática, PUC-Rio, 2000.
- [Vila00b] Vilain P., Schwabe D. & De Souza C.S., A Diagrammatic Tool for Representing User Interaction in UML, in Proceedings of the 3rd International Conference UML'2000, The Unified Modeling Language, Advancing the Standard, York, UK, October 2-6, LNCS 1939, Springer, 2000, pp. 133-147.
- [Vill02a] Villanova M., Belkhatir N. & Martin H., A Process Environment Supporting Web Information Systems, in International Journal of Computers and Applications (IJCA), 24(2), Acta Press, 2002, pp. 70-76.
- [Vill02b] Villanova-Oliver M., Gensel J. & Martin H., Une approche objet pour l'accès progressif à l'information, Actes de la conférence Langages et Modèles à Objets (LMO'02), Montpellier, 23-25 Janvier 2002, Revue des Sciences et Technologies de l'Information (RSTI), série L'objet, Hermès-Lavoisier, 8(1-2), 2002, pp. 61-78.
- [Vill02c] Villanova-Oliver M., Gensel J. & Martin H. Stratification of the Information Space in Web-based Information Systems, Proceedings of the 10th European Conference on Information Systems, Gdańsk, Poland, June 6-8, 2002
- [W3C98] World Wide Web Consortium, Cascading Style Sheets, level 2, CSS2 Specification, W3C Recommendation, May 12, 1998. <http://www.w3.org/TR/REC-CSS2>
- [W3C99a] World Wide Web Consortium, Resource Description Framework (RDF), Model and Syntax Specification, W3C Recommendation, February 22, 1999. <http://www.w3.org/TR/REC-rdf-syntax/>
- [W3C99b] World Wide Web Consortium, HyperText Markup Language, HTML Specification version 4.01, W3C Recommendation, December 24, 1999. <http://www.w3.org/TR/html4/>
- [W3C00] World Wide Web Consortium, Extensible Markup Language, XML Specification – version 1.0 (Second Edition), W3C Recommendation, October 6, 2000. <http://www.w3.org/TR/2000/REC-xml-20001006>
- [W3C01] World Wide Web Consortium, eXtensible Stylesheet Language, XSL Specification – version 1.0, W3C Recommendation, October 15, 2001. <http://www.w3.org/TR/xsl/>
- [W3C02] World Wide Web Consortium, RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft, April 30, 2002. <http://www.w3.org/TR/rdf-schema>

- [Zade75] Zadeh L.A., The concept of a linguistic variable and its application to approximate reasoning, *Information Science*, 1, 1975, pp. 199-250.
- [Zeln98] Zelnick N., Nifty Technology and Nonconformance : the Web in Crisis, *Computer*, October 1998, pp. 115-119.

ANNEXES

Annexe 1 – Les unités de contenus (Content Units) du modèle de composition de WebML

| | |
|---|--|
|  | <p>Description : Ces unités montrent l'information pour une instance d'entité du modèle structurel. Plusieurs unités peuvent être définies pour une entité donnée offrant ainsi des alternatives pour la présentation de l'information (e.g. version longue, courte, textuelle ou multimédia de l'instance). Définir une Data Unit nécessite d'indiquer l'entité du modèle structurel à laquelle elle fait référence et de sélectionner les attributs à inclure dans l'unité.</p> <p>Syntaxe XML</p> <pre><DATAUNIT id="identifiant_data_unit" entity="entité_de_référence"> <INCLUDE attribute="nom_atribut1"/> <INCLUDE attribute="nom_atribut2"/> ... </DATAUNIT></pre> <p>rmq: le tag <code><INCLUDEALL/></code> permet d'inclure tous les attributs de l'entité dans la Data Unit</p> |
|  | <p>Description : Ces unités montrent un ensemble d'instances correspondant à une entité du modèle structurel. La présentation de cet ensemble s'appuie sur la répétition de Data Units identiques. La spécification d'une Multi-Data Unit requiert d'indiquer à quel élément du modèle structurel font référence les instances à présenter (entité, relation) et de spécifier la Data Unit à utiliser pour présenter chaque instance.</p> <p>Syntaxe XML</p> <pre><MULTIDATAUNIT id="identifiant_multidata_unit" entity="entité_référence"> <DATAUNIT id="identifiant_data_unit" entity="entité_référence"> spécification de la Data Unit </DATAUNIT> </MULTIDATAUNIT></pre> |
|  | <p>Description : Ces unités montrent une liste d'instances correspondant à une entité du modèle structurel. Chaque instance est une entrée de la liste et représentée de façon succincte. La spécification d'une Index Unit passe par l'indication de l'élément du modèle structurel auquel font référence les instances à présenter (entité, relation) et par le choix des attributs qui représentent une instance dans l'index (clé d'entrée).</p> <p>Syntaxe XML</p> <pre><INDEXUNIT id="identifiant_index_unit" entity="entité_de_référence"> <DESCRIPTION Key="attribut_clé_de_l'index"/> </INDEXUNIT></pre> |
|  | <p>Description : Ces unités décrivent des commandes d'accès aux éléments d'un ensemble ordonné d'instances. Les accès définis permettent d'atteindre le premier, le dernier ou le <i>i</i>^{ème} élément de l'ensemble, ou d'atteindre l'élément précédant ou suivant l'élément courant. Une Scroller Unit est utilisée conjointement à une Data Unit qui représente l'élément courant. La spécification d'une Scroller Unit nécessite d'indiquer l'élément du modèle structurel qui fournit les instances à présenter et de choisir les attributs qui représentent une instance dans l'index</p> <p>Syntaxe XML</p> <pre><SCROLLERUNIT id="identifiant_scroller_unit" entity="entité_de_référence" first="valeur" last="valeur" previous="valeur" next="valeur"/></pre> <p>rmq: les attributs <code>first</code>, <code>last</code>, <code>previous</code> et <code>next</code> sont de type booléen.</p> |
|  | <p>Description : Ces unités intègrent des champs de saisie dont les valeurs sont utilisés pour rechercher des instances dans une entité du modèle structurel. Les instances qui respectent les termes de la requête sont montrées dans une Multidata Unit ou une Index Unit. La description d'une Filter Unit comprend la spécification de l'élément du modèle structurel sur lequel va porter la requête et l'expression d'un prédicat de recherche sur des valeurs d'attributs de cet élément. Les attributs à tester sont indiqués ainsi que l'opérateur de comparaison à utiliser.</p> <p>Syntaxe XML</p> <pre><FILTERUNIT id="identifiant_scroller_unit" entity="entité_de_référence" /> <SEARCHATTRIBUTE name="nom_atribut1" predicate="like between ..."/> ... </FILTERUNIT></pre> |
|  | <p>Description : Ce type d'unité ne définit pas un affichage de l'information mais traduit le fait qu'une instance est connectée à une autre par un lien sémantique. Il s'agit d'une facilité syntaxique pour exprimer une sorte d'index qui contient un unique objet et l'associe à un autre via une relation de cardinalité (1:1).</p> <p>Syntaxe XML</p> <pre><DIRECTUNIT id="identifiant_scroller_unit" relation="nom_relation"/></pre> |

Annexe 2 – Stratifications envisageables en fonction du type de projection utilisé par la requête

| TYPE DE PROJECTION | STRATIFICATION |
|--|--|
| <p>Projection Totale sur une collection Unique</p> <div style="border: 1px solid black; padding: 2px;"> Select * From C </div> <p style="text-align: center;"><i>ou</i></p> <div style="border: 1px solid black; padding: 2px;"> Select * From A </div> | <p>- Stratification similaire à la stratification en intension d'une EM 'Classe' pour stratifier C (avec * donne l'intension de C, i.e. ses attributs) <i>ou</i></p> <p>- Stratification similaire à la stratification en intension d'une EM 'Classe Association' pour stratifier A (avec * donne l'intension de A, i.e. ses rôles et ses attributs)</p> |
| <p>Projection Totale sur une collection Multiple</p> <div style="border: 1px solid black; padding: 2px;"> Select C1.*, C2.*, A1.* From C1, C2, A1 </div> | <p>- pour chaque constituant C1, C2, A1, stratification similaire à la stratification en intension d'une EM classe ou association. (avec C1.* donne l'intension de C1, i.e. ses attributs, avec C2.* donne l'intension de C2, i.e. ses attributs, et avec A1.* donne l'intension de A, i.e. ses rôles et ses attributs)</p> |
| <p>Projection Partielle sur une collection Unique</p> <div style="border: 1px solid black; padding: 2px;"> Select C.Att1, C.Att2, C.Att3 From C </div> <p style="text-align: center;"><i>ou</i></p> <div style="border: 1px solid black; padding: 2px;"> Select A.R1, A.R2, A.Att1 From A </div> | <p>- Stratification similaire à la stratification en intension d'une EM 'Classe' dont on considère une partie de l'intension (ici réduite aux trois attributs retenus par la projection) <i>ou</i></p> <p>- Stratification similaire à la stratification en intension d'une EM 'Classe Association' dont on considère une partie de l'intension (ici réduite aux deux rôles et à l'attribut retenus par la projection)</p> |
| <p>Projection Partielle sur une collection Multiple</p> <div style="border: 1px solid black; padding: 2px;"> Select C₁.Att₁, C₁.Att₂, C₂.Att₁, C₂.Att₂, A₁.R₂, A₁.R₁, A₁.Att₁, A₁.Att₂ From C₁, C₂, A </div> | <p style="text-align: center;">Deux possibilités :</p> <p style="text-align: center;">1 – Plusieurs Stratifications</p> <p>- pour chaque constituant C1, C2, A1, stratification similaire à la stratification en intension d'une EM 'Classe' ou 'Classe Association' dont on considère une partie de l'intension.</p> <p style="text-align: center;">2 – une Seule Stratification</p> <p>- stratification similaire à la stratification en intension d'une EM 'Classe Association' dont l'intension serait donnée par tous les attributs et les rôles retenus par la projection.</p> |

Annexe 3 - Expression des besoins en matière d'accès progressif

Lors de la conception d'un SIW, les modalités de l'accès progressif à l'information sont définies une fois le modèle de données établi et les utilisateurs identifiés. L'idée phare de l'accès progressif est d'éviter à l'utilisateur d'être submergé par l'information. Pour cela, le premier niveau de la stratification lui donne accès à l'essentiel de l'information, et en fonction de ses besoins, il accèdera à plus de détails. Une stratégie possible pour établir la stratification est alors d'estimer *a priori* le degré de pertinence des éléments d'information et de le corrélérer avec les niveaux de détail. Ainsi, plus un élément d'information est jugé pertinent, plus son accès est immédiat (i.e. plus il apparaît à un bas niveau de complétude).

Un dialogue entre le concepteur et l'utilisateur est nécessaire pour déterminer une stratification reflétant les besoins de l'utilisateur en matière d'accès progressif. Afin de faciliter cet échange, nous proposons au concepteur de SIW des notations spécifiques. Ces notations sont basées sur les cas d'utilisation et diagrammes de séquence UML, et exploitent également les CEMAP (Cas d'Exploitation du MAP) définis au chapitre 5 (cf. section 5.2).

Les *cas d'accès progressif* (CAP) montrés dans l'Illustration 1 constituent le point d'entrée du dialogue avec l'utilisateur. A ce niveau d'abstraction, ils visent à l'informer des possibilités offertes par le SIW en termes de stratification de l'espace fonctionnel, d'un rôle fonctionnel et d'une fonctionnalité.

Nous illustrons les principes d'une démarche utilisant ces notations en les appliquant à la spécification de la stratification attendue pour une fonctionnalité avec accès progressif.

Dans l'Illustration 1b, le cas d'utilisation qui correspond à la fonctionnalité étudiée '*Consulter les informations sur les spécialistes*' est traité ici comme un CAP. Il étend le cas général '*accéder progressivement au résultat retourné par une fonctionnalité*'.

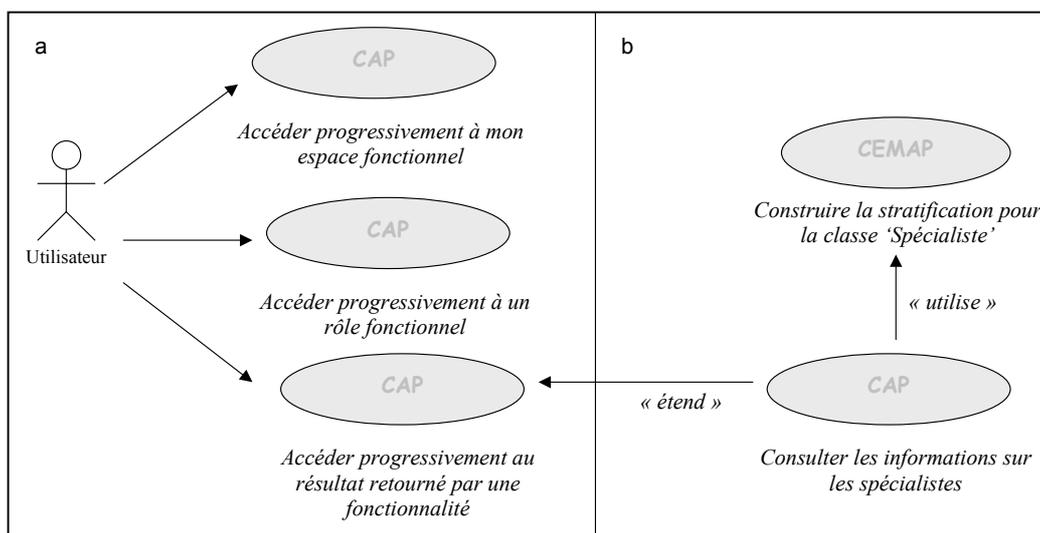


Illustration 1 - Cas d'utilisation pour l'accès progressif (CAP).

Basés sur la notation des cas d'utilisation, on les distingue grâce au label **CAP** – a. les CAP de premier niveau destinés à initier le dialogue avec l'utilisateur – b. (en bas) le cas d'utilisation associé à une fonctionnalité du SIW traité comme un CAP qui étend le CAP général. (en haut) le CAP utilise un CEMAP de construction d'une stratification défini par le protocole d'exploitation du MAP.

Le cas d'accès progressif nécessite de construire une stratification de la classe Spécialiste : il utilise donc le CEMAP de création d'une stratification appliqué à cette classe. L'étape suivante de conception est documentée par les *diagrammes de séquence pour l'accès progressif* (DSAP).

Également inspirés des notations UML, les DSAP traduisent le comportement du système, et à, plus bas niveau, des objets du système, correspondant aux CAP. La dimension temporelle introduite par les diagrammes de séquence permet d'exprimer le caractère progressif de l'accès. L'Illustration 2 présente le DSAP qui documente le CAP 'consulter les informations sur les Spécialistes'.

Elle exprime ce que l'utilisateur attend du système en termes d'accès progressif à la classe Spécialiste. Suite à la demande émanant de l'utilisateur, le système retourne l'information sur les rapporteurs en donnant accès aux attributs nom et prénom de la classe Spécialiste pour en visualiser les instances. L'utilisateur peut ensuite demander à voir plus de détail : les attributs code et organisme complètent alors la description des instances de la classe. Dans ce cas, l'attribut adresse restera masqué pour cet utilisateur.

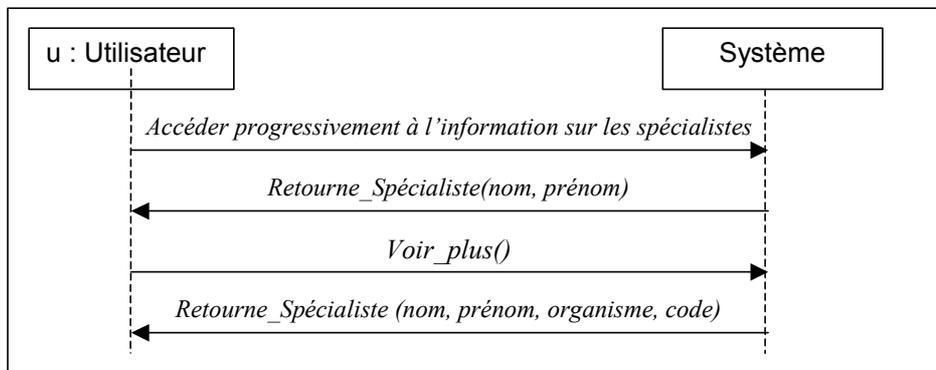


Illustration 2 – Diagramme de Séquence pour l'Accès Progressif
Ce diagramme correspond au CAP 'consulter les informations sur les spécialistes'.

Le concepteur poursuit la conception en utilisant le diagramme de collaboration que nous avons présenté dans le protocole d'exploitation du MAP. Il est adapté au cas spécifique de la fonctionnalité à stratifier. L'Illustration 3 reproduit ce diagramme. Les opérations (paramètres, résultats) sont précisées en respect des spécifications données par le DSAP précédent.

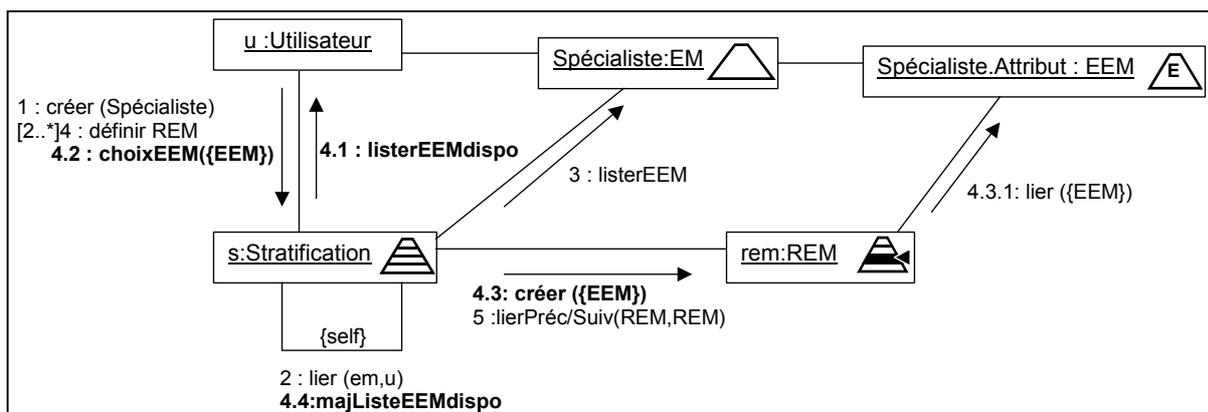


Illustration 3 – Diagramme de Séquence pour l'Accès Progressif à l'information d'une classe
Deux instances de la classe REM sont créées pour définir la stratification de la classe Rapporteur selon les besoins de l'utilisateur u.

Enfin, dans l'optique de proposer des modalités d'accès progressif qui répondent aux attentes des utilisateurs en la matière, nous avons proposé une extension spécifique aux cas d'utilisation. Ces notations s'appuient sur les propositions décrites dans le cadre du protocole d'exploitation du MAP. A travers des cas d'utilisation et diagrammes de séquence *pour l'accès progressif*, nous avons proposé des notations permettant de spécifier les besoins des utilisateurs en matière d'accès progressif. Les attentes des utilisateurs quant aux modalités d'accès progressif qu'ils souhaitent voir mises en œuvre peuvent être exprimées aux trois niveaux du Modèle des Fonctionnalités : Espace Fonctionnel, Rôle Fonctionnel et Fonctionnalité. De façon évidente, un concepteur n'a pas la possibilité de définir, pour et avec chaque utilisateur potentiel, les stratifications souhaitées. Son rôle consiste donc à définir les stratifications partagées par les membres d'un même groupe. Ainsi, le concepteur décrit les stratifications des Rôles Fonctionnels (qui sont directement liés aux groupes) et des Résultats de Requêtes pour les Fonctionnalités avec accès progressif. La stratification de l'Espace Fonctionnel sera définie par l'utilisateur lui-même, lors de l'utilisation du SIW.

Cette approche présente l'inconvénient de ne pouvoir être légitimement menée à bien lors de la phase de conception pour tous les utilisateurs potentiels du SIW. Nous pallions cet inconvénient en laissant la possibilité à l'utilisateur de modifier les stratifications établies une fois le système en place.

Annexe 4 – Modèle et principes de publication utilisé par Cocoon.

Le modèle de publication utilisé par Cocoon est fondé sur la séparation en trois vues d'un document XML. Les deux premières vues correspondent, dans la technologie XML, au *contenu* du document (données au format XML) et à la feuille de style au format XSL pour la *visualisation* de ce document. La *logique* d'un document, qui constitue la troisième vue, est définie dans le fichier de configuration de Cocoon « sitemap.xmap ». Ce fichier permet d'établir la liste des opérations qui seront réalisées lors de la requête d'affichage d'un fichier XML.

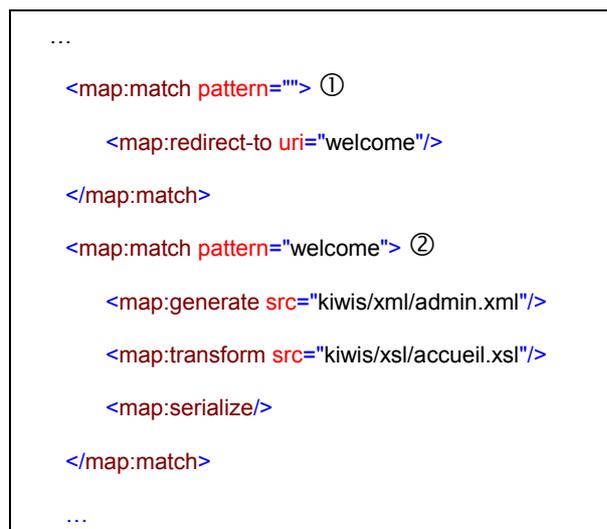


Illustration 4 – Extrait du fichier de configuration de Cocoon « sitemap.xmap »

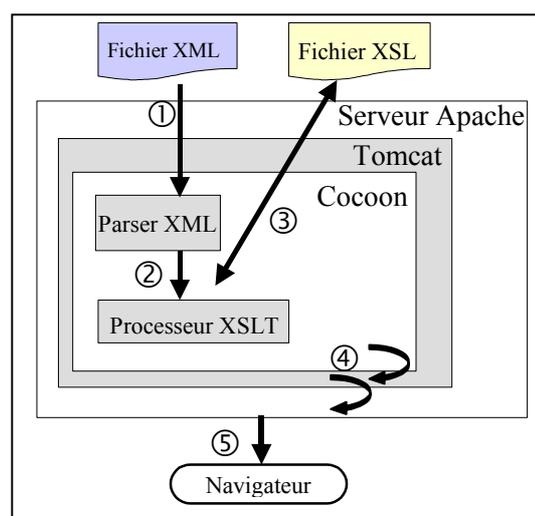


Illustration 5 – Publication XML dans un navigateur HTML

Dans l'extrait de « sitemap.xmap » donné dans l'illustration 4, la logique est la suivante (on considère que le site Web est <http://www-kiwis:8080>) : lorsque l'internaute demande la page <http://www-kiwis:8080/> (« pattern = "" ») ① Cocoon se reporte au bloc ②, récupère le document « admin.xml » et lui associe la feuille de style « accueil.xsl ». La demande de la page <http://www-kiwis:8080/welcome> aurait conduit au même résultat (en atteignant directement le bloc ②).

En distinguant ces trois vues, Cocoon permet de les concevoir et de les administrer séparément. Cette approche favorise aussi la réutilisation des styles de présentation sur une même famille de documents et procure un gain de temps important dans la mise en œuvre de la publication.

La Cocoon s'appuie ensuite sur les fonctions de transformation d'XSL pour fusionner ces documents en une information publiable sur le Web.

L'apport majeur de cet outil est qu'il permet de délivrer, sans la dupliquer, une même information sous plusieurs formats de présentation et cela indépendamment du type de navigateur Web. Un des formats de sortie de Cocoon, HTML 4.0, est d'autant plus utile que tous les navigateurs ne supportent pas encore XML et XSL.

Le principe de publication XML dans des navigateurs HTML est le suivant (cf. Illustration 5) :

Lorsque l'utilisateur appelle la page <http://www-kiwis:8080/welcome>, le serveur Apache appelle Tomcat. Tomcat lance la servlet Cocoon. Cocoon vérifie dans un premier temps que le résultat du traitement n'est pas déjà dans le cache (on suppose dans cet exemple, qu'il n'est pas dans le cache) et

récupère les fichiers à l'aide de « sitemap.xmap ». Cocoon ouvre ensuite le fichier « admin.xml » ①. Le parseur de Cocoon récupère les instructions des traitements à effectuer (ici conversion en HTML) et appelle le processeur XSLT ②. Le processeur XSLT ouvre la feuille de style associée au fichier xml et effectue la transformation en HTML ③. Cocoon transmet le document à Tomcat, qui le transmet au serveur Apache ④. Finalement, Apache passe le document au navigateur de l'utilisateur ⑤.

RÉSUMÉ :

Cette thèse traite de l'ingénierie des Systèmes d'Information basés sur le Web (SIW) adaptables à l'utilisateur. En tant que Système d'Information qui sous-tend le fonctionnement de l'organisation, le SIW doit offrir aux différents acteurs les fonctionnalités adéquates pour traiter l'information. Dans un système basé sur le Web, les fonctionnalités sont proposées à travers une interface structurée en nœuds et en liens de navigation selon l'approche hypermédia. La quantité de fonctionnalités et d'informations disponibles, combinée à une structure hypermédia, peut amener l'utilisateur à s'égarer lors de la navigation et à subir une surcharge cognitive.

La notion d'*Accès Progressif* et son intégration dans une démarche de conception de SIW adaptables à l'utilisateur visent à répondre à ce problème. L'Accès Progressif permet à un utilisateur de disposer d'un système lui offrant un accès immédiat aux fonctionnalités/informations les plus pertinentes, et lui permettant d'atteindre graduellement plus de fonctionnalités/informations, si besoin.

Le *Modèle d'Accès Progressif* (MAP) formalise les concepts et les mécanismes d'exploitation de cette proposition. Générique, réutilisable et applicable à divers domaines exprimés selon différents formalismes (E/R, objet, XML,...), le MAP est le cadre de référence de notre approche pour la conception de SIW. Celle-ci repose sur quatre modèles appelés *Modèles du Domaine, des Fonctionnalités, de l'Hypermédia et des Utilisateurs*. Étroitement reliés entre eux et au MAP, ces modèles couvrent les différentes facettes d'un SIW. Ils permettent au concepteur de spécifier des SIW adaptables à l'utilisateur, en termes de contenu, de présentation et d'accès progressif.

KIWIS (*Knowledge for Improving Web Information Systems*) est une plate-forme dédiée aux concepteurs de SIW qui valide l'approche par accès progressif. KIWIS implémente la démarche de conception proposée en complément des modèles et prend en charge la génération dynamique du SIW conçu.

MOTS-CLÉS : Systèmes d'Information basés sur le Web, Adaptabilité, Conception, Accès Progressif

ABSTRACT :

This study deals with the engineering of Web Information Systems (WIS) which are adaptable to their users. Like every Information System supporting the organization's functioning, a WIS has to provide its users with functionalities for efficiently processing information. In such Web-based systems, functionalities are generally proposed through an hypermedia interface whose structure is a composition of nodes related by navigational links. Such a large amount of functionalities and information accessible through this hypermedia can lead to some disorientation or cognitive overload syndromes experienced by the users. It is therefore crucial to adapt to users the functionalities, the content and the presentation of information to limit these negative effects.

We address this issue by introducing and implementing the notion of *Progressive Access* into a WIS design approach. Our objective is to offer the WIS design the possibility to build a WIS which has the capacity to deliver *progressively* a personalized information to its end users. First, information considered as essential with regard to their needs is provided, and then, some additional information, if needed, is gradually available through navigation mechanisms.

The Progressive Access Model (PAM) describes the concepts and mechanisms of our proposition. The PAM is general enough to apply to different application domains expressed using different representation formalisms (E/R, object, XML,...). This model is central in our design approach. The PAM is closely related to four other models (Data Model, Functional Model, Hypermedia Model, User Model). We show how these five models are exploited during the design phase leading to the generation of WIS which dynamically adapt content, presentation and progressive access features to their users.

Design and generation of such systems are supported by a platform called KIWIS (*Knowledge for Improving Web Information Systems*) which validates the progressive access approach.

KEYWORDS : Web-based Information Systems, Adaptability, Design, Progressive Access