



HAL
open science

WISDOM : Méthode d'aide à la conception de sites web basée sur la réutilisation conceptuelle et logicielle

Etienne Cocquebert

► To cite this version:

Etienne Cocquebert. WISDOM : Méthode d'aide à la conception de sites web basée sur la réutilisation conceptuelle et logicielle. Web. Université de Valenciennes et du Hainaut-Cambresis, 2008. Français. NNT : . tel-00404523

HAL Id: tel-00404523

<https://theses.hal.science/tel-00404523>

Submitted on 16 Jul 2009

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.



UMR CNRS 8530

Numéro d'ordre : 08/18

THESE

présentée à

L'UNIVERSITE DE VALENCIENNES ET DU HAINAUT CAMBRESIS

en vue de l'obtention du titre de

DOCTEUR de l'Université de Valenciennes et du Hainaut Cambrésis

Spécialité Automatique des Systèmes Industriels et Humains

Par

Etienne COCQUEBERT

Titre :

WISDOM : Méthode d'aide à la conception de sites web basée sur la réutilisation conceptuelle et logicielle

Soutenue le 3 décembre 2008 devant la commission d'examen composée de :

M. ALDANONDO	Professeur à l'Ecole des Mines d'Albi	Rapporteur
G. HALIN	Maître de Conférences HDR à l'Université de Nancy 2	Rapporteur
C. CAUVET	Professeur à l'Université Aix-Marseille 3	Examineur
S. LAMOURI	Professeur à Supméca à Paris	Examineur
D. RIEU	Professeur à l'INP Grenoble	Examineur
C. PIERRE DE GEYER	Responsable formation à Anaska	Invité
D. TRENTESAUX	Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis	Co-directeur
C. TAHON	Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis	Directeur de thèse



UMR CNRS 8530

Numéro d'ordre : 08/18

THESE

présentée à

L'UNIVERSITE DE VALENCIENNES ET DU HAINAUT CAMBRESIS

en vue de l'obtention du titre de

DOCTEUR de l'Université de Valenciennes et du Hainaut Cambrésis

Spécialité Automatique des Systèmes Industriels et Humains

Par

Etienne COCQUEBERT

Titre :

WISDOM : Méthode d'aide à la conception de sites web basée sur la réutilisation conceptuelle et logicielle

Soutenue le 3 décembre 2008 devant la commission d'examen composée de :

M. ALDANONDO	Professeur à l'Ecole des Mines d'Albi	Rapporteur
G. HALIN	Maître de Conférences HDR à l'Université de Nancy 2	Rapporteur
C. CAUVET	Professeur à l'Université Aix-Marseille 3	Examineur
S. LAMOURI	Professeur à Supméca à Paris	Examineur
D. RIEU	Professeur à l'INP Grenoble	Examineur
C. PIERRE DE GEYER	Responsable formation à Anaska	Invité
D. TRENTESAUX	Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis	Co-directeur
C. TAHON	Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis	Directeur de thèse

Pour Michèle,
Pour Stéphane, Laura et Julie,
A la mémoire de mes parents car « il fallait avoir confiance... »

Remerciements

Je remercie Michel ALDANONDO, Professeur à l'Ecole des Mines d'Albi, d'avoir accepté d'être rapporteur de ma thèse. Expert en systèmes d'information et conception produit au sein de la communauté productique, ses remarques et ses questions judicieuses, notamment lors des JD MACS 2007, m'ont permis de prendre du recul par rapport au cadre applicatif à partir duquel j'ai construit ma thèse.

Je remercie Gilles HALIN, Maître de Conférences HDR à l'Université de Nancy 2 d'avoir également accepté d'être rapporteur de ma thèse. Spécialiste en conception hypermédia et en systèmes d'information web, il m'a montré lors de nos entretiens les points sur lesquels il fallait davantage insister ou approfondir. Ses conseils ont été précieux pour la préparation de la soutenance.

Je remercie Corine CAUVET, Professeur à l'Université Aix-Marseille 3, de m'avoir fait l'honneur de bien vouloir examiner mon travail étant donné l'étendue de son travail dans le développement des systèmes d'information. Son livre m'a été d'une grande utilité pour positionner et baser mon travail par rapport à l'ingénierie de la réutilisation.

Je remercie Dominique RIEU, Professeur à l'INPG à Saint Martin d'Hères, dont la participation est également un honneur, étant donné l'importance des travaux menés au LIG sous sa direction dans le domaine de l'approche patron. Nos contacts par mail ou lors d'INFORSID 2007 ont enrichi la manière dont mes travaux ont été menés vis-à-vis de l'utilisation des patrons et vis-à-vis de l'ingénierie dirigée par les modèles.

Je remercie Samir LAMOURI, Professeur à Supméca à Paris, d'avoir accepté d'examiner ce travail. Fervent promoteur de composants logiciels métiers qu'il a utilisé pour supporter l'organisation de CPI 2007, MOSIM 2008 et CPI 2009, ses critiques constructives ont contribué à l'élaboration du travail présenté dans ce mémoire. De plus, je lui suis reconnaissant de me donner la possibilité de tester mon travail auprès d'étudiants dans l'environnement de travail de Supméca.

Je remercie Cyril PIERRE DE GEYER, responsable formation à Anaska et co-auteur du livre « PHP5 Avancé » de participer à ce jury de thèse : sa participation est un honneur compte tenu de la notoriété de son livre dans le domaine. D'autre part, il est le co-initiateur du groupe de travail « Briques Logicielles » de l'AFUP¹ au sein duquel les réflexions que nous avons menées m'ont permis d'avancer dans la partie « identification et choix de composants ». A ce propos, j'en profite pour remercier Pascal Blacher : autre co-initiateur du groupe de travail, les composants référencés dans mon catalogue ont été caractérisés grâce à son implémentation des fiches enquêtes.

Je remercie Damien TRENTESAUX, Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis, d'avoir été à l'origine de ce travail en me montrant l'intérêt scientifique de la manière dont je menais mon travail de webmaster et le bénéfice de sa diffusion pour la communauté sous la forme d'une thèse. Sans lui, la formalisation n'aurait pas été réalisée. Je le remercie également pour m'avoir appris la rigueur nécessaire pour analyser, prendre de la hauteur et proposer « quelque chose qui tient la route ». Le travail n'aurait pas abouti sans cela.

Je remercie Christian TAHON, Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis, de m'avoir offert la structure nécessaire à la réalisation d'une thèse, mais également d'avoir largement contribué à la mise en place de la réflexion scientifique qui a mené au résultat qui est présenté dans ce mémoire. Malgré ses nombreuses occupations, un nombre important de réunions ont dû être tenues avant de trouver la bonne direction. Je tiens également à le remercier pour l'autonomie totale qu'il m'a accordé dans le partage de mes journées de travail entre la conduite de cette thèse et mes activités d'ingénieur d'études au sein de l'équipe « système de production » du LAMIH.

Je remercie également les directions successives du GRP (de 1998 à 2002) et du GDR MACS (depuis 2002) : en me faisant confiance pour m'occuper d'un site web d'envergure (pour un groupe de recherche),

¹ <http://www.afup.org/>

elles ont largement contribuées à l'émergence et à l'élaboration de WISDOM et à l'implémentation du WISDOM Tool.

Pour m'avoir permis d'acquérir l'expertise en développement web en me proposant de concevoir leurs sites web et/ou de développer des composants logiciels spécifiques², je remercie également :

- J.-P. Bourrières, porteur du RTP-PCM de 2002 à 2004 ;
- J. Oudin et P. Millot, porteurs du pôle ST2 de 2001 à 2004 ;
- les organisateurs des journées STP à Valenciennes (STP '06) et des journées prospectives STP à Lyon (Lyon '07),
- les organisateurs des conférences suivantes :
 - éditions 2005, 2007 et 2009 des conférences LaPlagne,
 - journées techniques AIP-PRIMECA « JTSAGT '04 » et « CAO-Calculs '08 »,
 - éditions 2003, 2005, 2007 des conférences PENTOM,
 - MOSIM '08,
 - CPI '07 et CPI '09,
 - EAM '06 ;

En me faisant confiance, ils ont permis la proposition de WISDOM et l'implémentation du WISDOM Tool.

Je remercie Eric Markiewicz, directeur du LAMIH, d'avoir lancé le groupe « communications » du LAMIH : les réflexions que nous y avons menées ont alimenté la définition de WISDOM et m'ont fournies un exemple réel d'application sous la forme du nouveau site web.

Je remercie les collègues de l'équipe et entre autre Alain, Philippe et Thierry : leurs remarques pertinentes ont montré que mes idées étaient bonnes, mais qu'il fallait creuser ! Une partie de ce travail n'aurait pas vu le jour sans Sondès : c'est grâce à elle que son frère est venu travailler dans l'équipe et qu'il a développé le site web de GISEH en utilisant mon travail ! Cela a apporté de l'eau au moulin de la validation. Concernant la préparation de la soutenance, je remercie Marlène car sa prise en main efficace des détails de l'organisation logistique ont permis de la préparer au mieux ; Thérèse, Thierry et Bruno pour leurs remarques judicieuses concernant le fond ; ceux dont je n'ai pas eu connaissance de leur implication.

Je remercie ma famille, ma belle-famille et les amis pour leur soutien actif mais discret dans la réalisation de ce travail de longue haleine : les questions sur l'avancement, les propositions et les initiatives d'organisation ont apporté la détente et/ou remis en selle la motivation pour aller au bout.

Je remercie la musique de m'avoir aidé à rester éveillé au plus fort de la rédaction et du développement.

Je remercie Michèle : sans la motivation qu'elle m'a donnée, sans sa présence et sans nos enfants : il n'y aurait rien eu.

Enfin, je remercie mes parents pour leur soutien posthume : j'ai rarement autant vérifié qu'« il faut avoir confiance ». Je ne doute pas de leur fierté.

² LAMIHSubmission (gestion de relecture), LAMIHRegister (inscription à un événement), LAMIHDoc (gestion de documents, avec la participation de Paul Dubuc et Khalid Roudani, étudiants à l'ENSIAME en 2007-2008),

Table des matières

Introduction générale	19
Chapitre I : Contexte de l'étude	21
1 Conception d'une application logicielle	23
1.1 Etapes de conception	23
1.2 Processus de conception	23
2 Domaines de recherche concernés	24
2.1 Ingénierie de la réutilisation	24
2.1.1 Types d'approches	24
2.1.2 Types d'entité utilisées	24
2.1.3 Les composants de conception : les patrons	25
2.1.3.1 Définition	25
2.1.3.2 Types de patrons	26
2.1.3.3 Processus d'utilisation d'un patron	27
2.1.4 Les composants logiciels	27
2.1.4.1 Définition	27
2.1.4.2 Types de composants	27
2.1.4.3 Processus d'utilisation d'un composant	28
2.2 Ingénierie dirigée par les modèles	29
2.3 Ingénierie du web	29
3 Cadre applicatif de notre étude	30
3.1 Les applications web	30
3.2 Typologie des sites web	31
4 Fonctionnalités à mettre en œuvre pour la conception d'un site web	31
4.1 Assister la réalisation du processus de conception	31
4.2 Identifier et faciliter la réutilisation de solutions de conception	32
4.3 Identifier et faciliter le choix de composants logiciels	32
4.4 Accélérer la mise en place d'un site web	32
5 Objectifs	32
Chapitre II : Etat de l'art	35
1 Fonctionnalité « Assister la réalisation du processus de conception »	37
1.1 Guider le concepteur	37
1.2 Apporter une aide à la réalisation des étapes du processus	39
2 Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »	40
2.1 Identifier des solutions de conception	41
2.2 Faciliter la réutilisation des solutions de conception	42
3 Fonctionnalité « Identifier et faciliter le choix de composants logiciels »	43
3.1 Identifier des composants logiciels	43
3.1.1 Approche orientée « modèle »	43
3.1.2 Approche orientée « processus »	45
3.2 Faciliter le choix d'un composant	47
4 Fonctionnalité « Accélérer la mise en place d'un site web »	47
4.1 Génération de données	47
4.2 Accélérer la conception d'un site web	50
5 Conclusion	52
Chapitre III : Démarche	55
1 Fonctionnalité « Assister la réalisation du processus de conception »	57
1.1 Guider le concepteur dans le processus	57

1.1.1	Définir les étapes	57
1.1.2	Définir le processus	57
1.1.3	Définir le modèle de données	58
1.1.3.1	Définir le modèle pour la navigation	58
1.1.3.2	Définir le modèle pour la présentation	59
1.1.3.3	Définir le modèle pour le contenu	60
1.1.3.4	Définir le modèle pour l'implémentation	60
1.2	Aider le concepteur à réaliser les étapes du processus	61
2	Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »	61
2.1	Identifier et proposer des solutions de conception	61
2.1.1	Définir le modèle d'analyse	62
2.1.2	Définir l'utilisation du modèle d'analyse	62
2.1.3	Définir l'exploitation des analyses	62
2.2	Faciliter la réutilisation des solutions de conception	62
2.2.1	Définir la sélection d'une solution de conception	63
2.2.2	Définir l'instanciation d'une solution de conception	63
2.2.3	Définir l'adaptation d'une solution de conception	63
3	Fonctionnalité « Identifier et faciliter le choix de composants logiciels »	63
3.1	Identifier des composants logiciels	63
3.1.1	Définir un modèle de composant	64
3.1.2	Définir l'utilisation du modèle de composant	64
3.1.3	Définir l'exploitation des analyses de composants	64
3.2	Faciliter le choix de composants	64
4	Fonctionnalité « Accélérer la mise en place d'un site web »	65
4.1	Génération de structures de données	65
4.2	Accélérer la conception d'un site web	66
4.2.1	Définir un modèle d'architecture d'accueil	66
4.2.2	Définir l'instanciation du modèle de l'architecture	66
4.2.3	Définir l'utilisation de l'architecture instanciée	66
5	Conclusion	67
Chapitre IV : Proposition		69
1	Fonctionnalité « Assister la réalisation du processus de conception »	71
1.1	Etapas et processus de conception	72
1.2	Modèle de site web	75
1.3	Patrons de domaine	79
1.3.1	Patron « navigation »	79
1.3.2	Patron « présentation »	80
1.3.3	Patron « contenu »	83
1.3.4	Patron « implémentation »	85
2	Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »	86
2.1	Proposer des solutions de conception	86
2.1.1	Modèle de fiche d'analyse	86
2.1.2	Utilisation des fiches d'analyses	88
2.1.3	Mécanisme d'exploitation des fiches d'analyses	90
2.2	Faciliter la réutilisation des solutions de conception	90
3	Fonctionnalité « Identifier et faciliter le choix de composants logiciels »	90
4	Fonctionnalité « Accélérer la mise en place d'un site web »	91
4.1	Génération de données	91
4.1.1	Génération pour la navigation	91
4.1.2	Génération pour le contenu	92
4.2	Génération d'une architecture logicielle d'accueil	92
5	Conclusion	93

Chapitre V : Mise en œuvre	95
1 Conception d'un site web	97
1.1 Etape 1 : « définition du contexte »	97
1.2 Etape 2 : « conception de la navigation »	98
1.3 Etape 3 : « conception de la présentation »	100
1.3.1 Identification des informations communes.	100
1.3.2 Conception du modèle de présentation.	101
1.4 Etape 4 : « définition de l'implémentation »	101
1.5 Etape 5 : « choisir un item de menu »	102
1.6 Etape 6 : « conception des contenus »	103
1.7 Aide à l'utilisateur	107
2 Analyse de sites web	107
2.1 Mise en œuvre des fiches d'analyses	107
2.2 Utilisation des fiches d'analyses	108
3 Analyse de composants	111
4 Conclusion	113
Chapitre VI : Application et validation	115
1 Présentation du cas d'application	117
2 Conception du site web	117
2.1 Etape « identification du contexte »	117
2.2 Etape « conception de la navigation »	117
2.3 Etape « conception de la présentation »	118
2.4 Etape « définition d'une implémentation » pour l'affichage du menu	119
2.5 Etape « choisir un item de menu »	120
2.5.1 Conception associée à l'item « membres »	121
2.5.1.1 Etape « conception d'un contenu »	121
2.5.1.2 Etape « définition d'une implémentation » pour l'affichage d'un contenu	123
2.5.2 Conception associée à l'item « documents administratifs »	124
2.6 Résultat	125
2.7 Conclusion sur la conception du site web du LAMIH	126
3 Validation	126
3.1 Fonctionnalité « Assister la réalisation du processus de conception »	127
3.1.1 Capacité à guider le concepteur	127
3.1.1.1 Validation par rapport à la méthode	127
3.1.1.2 Validation par rapport à l'outil	128
3.1.2 Aider à la réalisation des étapes	128
3.2 Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »	128
3.2.1 Validation par rapport à la méthode	129
3.2.2 Validation par rapport à l'outil	129
3.3 Fonctionnalité « Identifier et faciliter le choix de composants logiciels »	130
3.3.1 Validation par rapport à la méthode	130
3.3.2 Validation par rapport à l'outil	130
3.4 Fonctionnalité « Accélérer la mise en place d'un site web »	131
3.4.1 Générer une structure de données	131
3.4.2 Faciliter la conception d'un site web	131
4 Conclusion	132
Conclusion générale	133
Références bibliographiques	135
Annexe 1. Modèle des données d'un site web	139

Annexe 2.	Formalisme P-Sigma	140
4.1	Structure générale	140
4.2	La partie Interface	140
4.3	La partie Réalisation	141
4.4	La partie Relation	141
Annexe 3.	Patrons de domaine de WISDOM	143
4.5	Patron « navigation »	143
4.5.1	La partie Interface	143
4.5.2	La partie Réalisation	143
4.5.3	La partie Relation	146
4.6	Patron « présentation »	146
4.6.1	La partie Interface	146
4.6.2	La partie Réalisation	146
4.6.3	La partie Relation	150
4.7	Patron « contenu »	150
4.7.1	La partie Interface	150
4.7.2	La partie Réalisation	150
4.7.3	La partie Relation	153
4.8	Patron « implémentation »	154
4.8.1	La partie Interface	154
4.8.2	La partie Réalisation	154
4.8.3	La partie Relation	156
5	Génération de l'architecture d'accueil	156
Annexe 4.	Norme ISO 9126	157
Annexe 5.	page d'accueil de CAO-Calculs '08	158
Annexe 6.	page d'accueil de GISEH	159
Résumé		160

Table des figures

Figure 1 : étapes concernées par notre étude	24
Figure 2 : entités pour l'approche « système d'information par réutilisation »	25
Figure 3 : classification des patrons ([Saidane, 05])	26
Figure 4 : relations entre les processus de développement d'un système et d'évaluation d'un composant	28
Figure 5 : niveaux de modélisation de l'IDM (à partir de [Favre, 06a] et [Rieu, 07])	29
Figure 6 : types d'applications web ([Gnahou, 00])	31
Figure 7 : utilisation de WISDOM	33
Figure 8 : périmètre de l'utilisation de WISDOM	33
Figure 9 : processus défini pour guider le développement ([Parson, 06])	37
Figure 10 : acteurs impliqués par le processus ([Parson, 06])	38
Figure 11 : formalisation d'un patron	39
Figure 12 : informatiser la recherche dans un catalogue de patrons	41
Figure 13 : génération de code à partir d'un traitement automatique sur l'application de patrons	42
Figure 14 : caractérisation des composants	43
Figure 15 : approche orientée « modèle »	43
Figure 16 : utilisation d'un processus analytique	44
Figure 17 : méthode de sélection basée sur le modèle de domaine	44
Figure 18 : utilisation d'une ontologie	45
Figure 19 : reconnaissance de cartes de besoins (d'après [LE 01])	46
Figure 20 : appariement structurel (d'après [Khayati, 05])	46
Figure 21 : catégorisation des composants	47
Figure 22 : génération de code à partir de patrons (d'après [Couturier, 04])	48
Figure 23 : rétro-ingénierie dirigée par les méta-modèles	48
Figure 24 : recherche et configuration de gadgets sur iGoogle	49
Figure 25 : architecture logicielle d'accueil ([Parson, 05])	50
Figure 26 : proposition d'implémentation à partir de patrons	50
Figure 27 : génération automatique du code	51
Figure 28 : utilisation d'un framework ou d'un CMS	51
Figure 29 : page de configuration de Joomla	52
Figure 30 : spécifications du processus de conception	57
Figure 31 : spécifications pour définir le modèle de données	58
Figure 32 : représentation du menu d'un site web sous la forme d'une arborescence	58
Figure 33 : exemples de types de zones dans deux sites web différents en production	59
Figure 34 : types de services	60
Figure 35 : spécifications pour aider le concepteur	61
Figure 36 : spécifications d'un modèle d'analyse de sites web	62
Figure 37 : spécifications pour la réutilisation de solutions de conception	63
Figure 38 : spécifications pour identifier des composants logiciels	64
Figure 39 : spécifications pour faciliter le choix d'un composant logiciel	65
Figure 40 : spécifications pour générer des données	65
Figure 41 : spécifications pour accélérer la mise en place d'un site web	66
Figure 42 : diagramme d'activité du processus de conception d'un site web	73
Figure 43 : modèle associé à la dimension « navigation »	76
Figure 44 : modèle associé à la dimension « présentation »	77
Figure 45 : modèle associé à la dimension « contenu »	78
Figure 46 : modèle associé à la dimension « implémentation »	78
Figure 47 : processus du patron navigation	79
Figure 48 : processus du patron « présentation »	81
Figure 49 : exemples de zones sur une page d'accueil (colloque AIP-PRINECA)	82
Figure 50 : processus du patron « contenu »	83
Figure 51 : exemple de types de données sur un extrait de la page « conférence » du GDR-MACS	84

Figure 52 : processus du patron « identification du type d'implémentation »	85
Figure 53 : modèle général de la fiche d'analyse.....	86
Figure 54 : modélisation de la fiche d'analyse de la dimension « navigation »	86
Figure 55 : modélisation de la fiche d'analyse de la dimension « présentation »	87
Figure 56 : modélisation de la fiche d'analyse de la dimension « contenu »	87
Figure 57 : modélisation de la fiche d'analyse de la dimension « service ».....	88
Figure 58 : modélisation de l'analyse d'un site web	88
Figure 59 : extrait d'une analyse de la dimension « navigation » d'un site web	89
Figure 60 : besoin d'insertion d'un item dans la fiche d'analyse	89
Figure 61 : fiche d'analyse modifiée	89
Figure 62 : modèle de composant.....	91
Figure 63 : diagramme de classe pour générer l'implémentation du menu.....	91
Figure 64 : méta-modèle pour la génération de données.....	92
Figure 65 : modèle de l'architecture.....	93
Figure 66 : page d'accueil du WISDOM Tool.....	97
Figure 67 : étape « identification du contexte »	98
Figure 68 : conception de la navigation.....	99
Figure 69 : conception de la présentation	100
Figure 70 : choix d'une implémentation pour le menu.....	101
Figure 71 : choix d'un item de menu.....	103
Figure 72 : définition d'un contenu	104
Figure 73 : données définies.....	105
Figure 74 : association d'un lien à un item de menu	106
Figure 75 : association d'un service à un item	106
Figure 76 ; processus terminé.....	107
Figure 77 : définition du contenu d'une fiche	108
Figure 78 : référencement d'un site web	109
Figure 79 : extrait de la liste de sites web référencés	109
Figure 80 : mise en œuvre de l'analyse de la navigation.....	110
Figure 81 : mise en œuvre de l'analyse de la présentation	111
Figure 82 : informations relatives à l'installation.....	112
Figure 83 : informations relatives à l'utilisation	112
Figure 84 : informations relatives au support.....	112
Figure 85 : informations relatives à l'exploitation	112
Figure 86 : informations relatives à l'évolution	113
Figure 87 : informations générales	113
Figure 88 : identification du composant.....	113
Figure 89 : contexte de la conception du site du LAMIH	117
Figure 90 : liste adaptée des items de navigation	118
Figure 91 : liste adaptée des informations communes à toutes les pages.....	118
Figure 92 : modèle de présentation du site web du LAMIH	119
Figure 93 : choix de l'implémentation du menu.....	120
Figure 94 : choisir un item de menu pour y associer un contenu	121
Figure 95 : liste adaptée des items de navigation	122
Figure 96 : traitement terminé pour l'item « membre »	122
Figure 97 : composants techniques pour la fonction « affichage de données ».....	123
Figure 98 : traitement terminé du choix de l'implémentation	123
Figure 99 : composants techniques pour implémenter le service « gestion de documents ».....	124
Figure 100 : traitement terminé du service « gestion publications »	125
Figure 101 : page d'accueil du site web du LAMIH	125
Figure 102 : page « membre » du site web du LAMIH.....	125
Figure 103 : composant « LAMIHDoc ».....	126
Figure 104 : modèle des données d'un site web	139
Figure 105 : structure générale de P-Sigma ([Conte, 01a]).....	140
Figure 106 : processus du patron navigation	143

Figure 107 : modèle associé à la dimension « navigation »	145
Figure 108 : diagramme de classe pour générer l'implémentation du menu.....	146
Figure 109 : processus du patron « présentation »	147
Figure 110 : modèle associé à la dimension « présentation ».....	149
Figure 111 : Exemples de zones sur une page d'accueil (LaPlagne2007).....	149
Figure 112 : processus du patron « contenu »	151
Figure 113 : modèle associé à la dimension « contenu »	153
Figure 114 : Exemple de types de données sur un extrait de la page « conférence » du GDR MACS	153
Figure 115 : processus du patron « identification du type d'implémentation »	154
Figure 116: modélisation selon la dimension « implémentation »	155
Figure 117 : rubriques de la norme ISO 9126	157
Figure 118 : page d'accueil du site « CAO-Calculs '08 ».....	158
Figure 119 : page d'accueil du site « GISEH »	159

Table des tableaux

Tableau 1 : légende des graphiques utilisés dans les illustrations des analyses	37
Tableau 2 : chronologie des processus des méthodes de conception de sites web	38
Tableau 3 : détails de l'étape 1 « définition du contexte »	72
Tableau 4 : détails de l'étape 2 « conception/modification de la navigation ».....	72
Tableau 5 : détails de l'étape 3 « conception/modification de la présentation »	74
Tableau 6 : détails de l'étape 4 « définition/modification d'une implémentation »	74
Tableau 7 : détails de l'étape 5 « choisir un item de menu ».....	74
Tableau 8 : détails de l'étape 6 « conception/modification d'un contenu »	75
Tableau 9 : détails pour la proposition de menu de navigation	80
Tableau 10 : détails pour la récupération d'un menu sauvegardé	80
Tableau 11 : détails pour l'adaptation du menu de navigation.....	80
Tableau 12 : détails pour générer des données associées au menu de navigation	80
Tableau 13 : détails pour proposer des données communes	81
Tableau 14 : détails pour définir le modèle de présentation	82
Tableau 15 : détails pour récupérer la liste des données communes et le modèle de présentation	82
Tableau 16 : détails pour l'adaptation de la liste	83
Tableau 17 : détails pour l'insertion de données communes	83
Tableau 18 : détails pour la proposition de la liste des contenus associés.....	84
Tableau 19 : détails pour la récupération des contenus définis	84
Tableau 20 : détails pour l'adaptation de la liste	84
Tableau 21 : détails pour la génération de la table de données	85
Tableau 22 : détails pour identifier les composants.....	85
Tableau 23 : détails pour choisir un composant	85
Tableau 24 : détails pour rechercher ou réaliser un développement spécifique	86
Tableau 25 : résultat des analyses en fonction de la dimension de modélisation.....	89
Tableau 26 : résultat de l'adaptation des propositions en fonction de la dimension de modélisations	90
Tableau 27 : partie « interface » d'un patron selon le du formalisme P-Sigma	141
Tableau 28 : partie « réalisation » d'un patron selon le du formalisme P-Sigma.....	141
Tableau 29 : partie « relations » d'un patron selon le du formalisme P-Sigma.....	142
Tableau 30 : détails pour la proposition de menu de navigation	144
Tableau 31 : détails pour la récupération d'un menu sauvegardé	144
Tableau 32 : détails pour l'adaptation du menu de navigation.....	144
Tableau 33 : détails pour générer des données associées au menu de navigation	145

Tableau 34 : détails pour proposer des données communes	147
Tableau 35 : détails pour définir le modèle de présentation	148
Tableau 36 : détails pour récupérer la liste des données communes et le modèle de présentation	148
Tableau 37 : détails pour l'adaptation de la liste	148
Tableau 38 : détails pour l'insertion de données communes	149
Tableau 39 : détails pour la proposition de la liste des contenus associés.....	151
Tableau 40 : détails pour la récupération des contenus définis	151
Tableau 41 : détails pour l'adaptation de la liste	152
Tableau 42 : détails pour la génération de la table de données	152
Tableau 43 : détails pour identifier les composants.....	155
Tableau 44 : détails pour choisir un composant	155
Tableau 45 : détails pour rechercher ou réaliser un développement spécifique	155

Introduction générale

Depuis leur apparition en 1996 jusqu'en 2002 où le domaine est devenu mature, la nature et les acteurs d'un site web ont considérablement évolués [Bordage 08]. Les premiers sites diffusaient uniquement de l'information statique et étaient réalisés de manière artisanale par peu d'acteurs. Désormais, ils sont devenus un vecteur privilégié d'informations dynamiques (sites marchands, institutionnels, de marques, etc.), de dialogue entre les utilisateurs (sites communautaires, intranet, collaboratifs, etc.) et sont réalisés par des équipes projets faisant intervenir de multiples compétences (programmeurs, ergonomes, référenceurs, chefs de projet, graphistes, etc.).

En se focalisant sur la réalisation elle-même, cette évolution a demandé aux concepteurs de sites web de faire face à un ensemble de critères de plus en plus contraignants. Il leur faut mettre en œuvre, de manière toujours plus rapide, des fonctionnalités telles que la mise à jour régulière des sites, une interface conviviale et rapide d'accès, et un ensemble étendu de services interactifs. Pour éviter une augmentation de coût liée à celle du temps de développement et des exigences de qualité, de sécurité, etc., un nombre élevé de travaux ont été menés et ont proposés des outils et méthodes d'aide à la conception à caractère académique, commercial ou « open source ».

Or, le développement de nombreux sites web et de composants logiciels nous a amené aux constats suivants :

- les méthodes et outils pour la conception de sites web ne répondent pas totalement aux attentes des concepteurs car ces derniers ne sont pas guidés explicitement dans la démarche de conception, l'expérience de la conception de sites web n'est pas réutilisée, les composants logiciels sont rarement choisis en toute connaissance de cause ;
- il existe des invariants qui se dégagent de la conception d'un site web à l'autre dans la démarche de conception elle-même, dans l'expérience de conception en fonction de familles de sites web, dans le choix des composants logiciels à partir de la prise en compte de leurs caractérisations fonctionnelle *et* non fonctionnelle ;
- les invariants réutilisables ne sont pas ou peu formalisés.

C'est pourquoi nous avons souhaité proposer une méthode d'aide à la conception de sites web dont les objectifs sont de :

- guider le concepteur de sites web ;
- réutiliser une expérience de conception ;
- réutiliser des développements existants ;
- faciliter le développement du site lui-même.

Cette activité relève du génie logiciel, thématique historique du **L**aboratoire d'**A**utomatique de **M**écanique et d'**I**nformatique industrielle et **H**umaine (LAMIH) au sein duquel cette étude a été réalisée.

L'objet de ce mémoire est de détailler le résultat de cette thèse suivant six chapitres dont nous présentons maintenant les contenus.

Le chapitre I présente le contexte selon lequel cette étude a été réalisée. Il est constitué par l'identification des étapes du processus de développement d'un site web, la présentation des trois domaines de recherche dont l'étude relève (l'ingénierie de la réutilisation, l'ingénierie dirigée par les modèles, et l'ingénierie du web), la description du cadre applicatif de notre méthode. Ce contexte nous permet de définir l'objectif de notre étude et les fonctionnalités que nous proposons de mettre en œuvre.

Les chapitres suivants, structurés par chacune de ces fonctionnalités, ont les objectifs suivants :

- le chapitre II présente l'analyse de l'état de l'art dans chacun des domaines cités précédemment ;
- le chapitre III détaille la démarche à suivre pour définir une méthode d'aide à la conception de sites web ;
- le chapitre IV propose la réalisation de ces spécifications au sein de la méthode « WISDOM » ;
- le chapitre V présente une mise en œuvre de la méthode au sein du « WISDOM Tool ».

Le chapitre VI est décomposé en deux parties : la première décrit quelques étapes de la conception d'une nouvelle version du site web du LAMIH à l'aide du WISDOM Tool ; la seconde évalue le bénéfice apporté par la méthode WISDOM et par l'utilisation du WISDOM Tool pour concevoir des sites web.

La conclusion générale fait la synthèse de ce mémoire par rapport à notre objectif d'aider la conception de sites web en réutilisant des travaux logiciels ou de conception, et présente nos perspectives pour améliorer les résultats déjà concrets de cette étude.

Chapitre I : Contexte de l'étude

L'objectif de ce chapitre est dans un premier temps de délimiter les bornes de notre étude, constituées par l'identification des étapes du processus de conception qui nous concernent, les domaines de recherche dont notre étude relève et le cadre applicatif de notre méthode. Dans un second temps et en nous basant sur notre expérience de conception de sites web, ce chapitre présente les fonctionnalités que nous proposons de mettre en œuvre. Ces deux volets permettent de définir avec précision notre objectif en conclusion de ce chapitre.

Dans ce chapitre, les phases soulignées mettent en avant les choix réalisés pour mener cette étude.

1 Conception d'une application logicielle

L'étude de la conception d'une application logicielle concerne l'identification des étapes de conception et du processus selon lesquelles leur succession est organisée.

1.1 Etapes de conception

Depuis de nombreuses années, il est établi que le cycle de développement d'une application logicielle est composé des étapes suivantes [Coulanges, 96] [Meinadier, 98] :

- *Analyse*, pour recueillir les besoins des utilisateurs et les formaliser sous la forme d'un ensemble de fonctions externes ;
- *Conception globale*, pour définir l'architecture fonctionnelle du système indépendamment de toute considération technique, en identifiant les fonctions internes nécessaires pour satisfaire les fonctions externes. Ces fonctions internes sont remplies par des modules fonctionnels auxquels sont attribués des données en entrée et en sortie et des comportements internes ;
- *Conception détaillée*, pour définir l'architecture logicielle du système, et définir pour chaque module fonctionnel les structures de données et les algorithmes qui permettront de coder les comportements associés ;
- *Implémentation*, pour développer l'architecture logicielle du système et les modules fonctionnels eux-mêmes, après avoir choisi le langage de programmation, la plateforme et l'environnement de développement, les composants logiciels à intégrer ;
- *Test*, pour tester les modules intégrés et l'ensemble du système, afin de vérifier le bon fonctionnement de l'ensemble ;
- *Evaluation*, pour vérifier que le système final répond aux besoins identifiés lors de l'analyse.

Dans le cadre de notre étude, nous nous intéressons aux trois étapes « conception globale », « conception détaillée » et « implémentation » (cf. Figure 1). Les autres ne sont pas prises en compte pour les raisons suivantes :

- l'étape d'analyse fait partie d'un domaine de recherche à part entière (l'ingénierie des besoins) que nous ne prenons pas en compte dans notre étude (cf. § 2 page suivante) ;
- l'étape de test peut faire appel à des outils logiciels spécifiques performants, et elle est parfois incluse dans l'implémentation elle-même sous la forme de tests unitaires ;
- l'étape d'évaluation ne fait plus partie de la conception elle-même car elle consiste à utiliser le système dans les conditions réelles.

1.2 Processus de conception

La définition des processus récents de conception (cf. leurs présentations dans [Lepreux, 05]) ont montré la nécessité de prendre en compte les deux aspects suivants :

- les itérations sur le déroulement du processus, nécessaires car des choix réalisés lors d'une étape peuvent remettre en cause des choix réalisés en amont et conduire à enrichir le cahier des charges avec de nouvelles spécifications (cf. Figure 1) ;
- la réutilisation de composants logiciels et des choix de conception, permettant de diminuer entre autre le temps global de conception.

Ces deux aspects sont donc également à prendre en compte pour définir notre processus de conception.

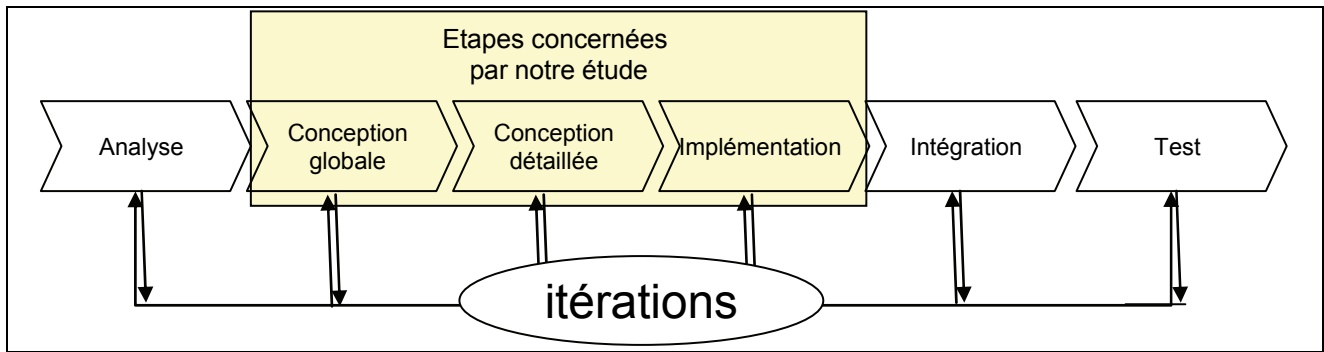


Figure 1 : étapes concernées par notre étude

2 Domaines de recherche concernés

Notre étude est à la croisée de trois domaines de recherche que nous présentons maintenant : l'ingénierie de la réutilisation, l'ingénierie dirigée par les modèles et l'ingénierie du web.

2.1 Ingénierie de la réutilisation

2.1.1 Types d'approches

Trois types d'approches sont habituellement distinguées [Oussalah, 99] [Cauvet, 01] :

- *approche « Bibliothèque de composants »*. Elle propose un catalogue de composants avec interface de recherche. Dans cette approche, la sélection, l'adaptation et l'intégration sont à la charge du développeur ;
- *approche « Système de réutilisation »*. Reliée à l'ingénierie *pour* la réutilisation ou « design for reuse », elle formalise des modèles d'entités réutilisables et leurs processus de gestion, de composition, d'adaptation et d'intégration ;
- *approche « Système d'information par réutilisation »*. Reliée à l'ingénierie *par* réutilisation ou « design by reuse », elle formalise des processus pour guider la sélection et la composition des entités réutilisées.

Dans le but d'apporter une aide à la conception de sites web par la réutilisation de composants logiciels et d'expériences de conception, notre étude relève de l'approche « système d'information par réutilisation », dont nous détaillons maintenant les entités et la manière dont elles sont utilisées.

2.1.2 Types d'entité utilisées

Les différents travaux relevant de l'approche « système d'information par réutilisation » ont conduit à la définition de différentes classifications d'entités réutilisables [Oussalah, 99] que nous avons synthétisées dans le tableau de la Figure 2.

Comme le montre ce tableau, les classifications ont toutes définies des entités qui leur sont spécifiques, mais en les rapportant à l'objectif de leur utilisation, le tableau met en évidence que seuls trois types d'entités sont à considérer :

- *les entités pour la conception et le développement de logiciels d'un même domaine d'application*. Ces entités concernent l'étape d'analyse. Elles se présentent sous la forme de méthodes d'analyse et de modélisation, de framework ou sont reliées à un domaine d'application ;
- *les entités pour la réutilisation de décisions de conception*. Ces entités concernent les étapes de conception globale et détaillée. Elles se présentent sous la forme de patrons, de frameworks et de méthodes de développement et intègrent les informations pour guider leur utilisation / adaptation / intégration au sein du logiciel.
- *les entités pour la production de logiciels*. Ces entités concernent l'étape d'implémentation. Elles se présentent sous la forme de fonctions, de sous-programmes, de classes, etc. dont l'adaptation et l'intégration sont à la charge du développeur.

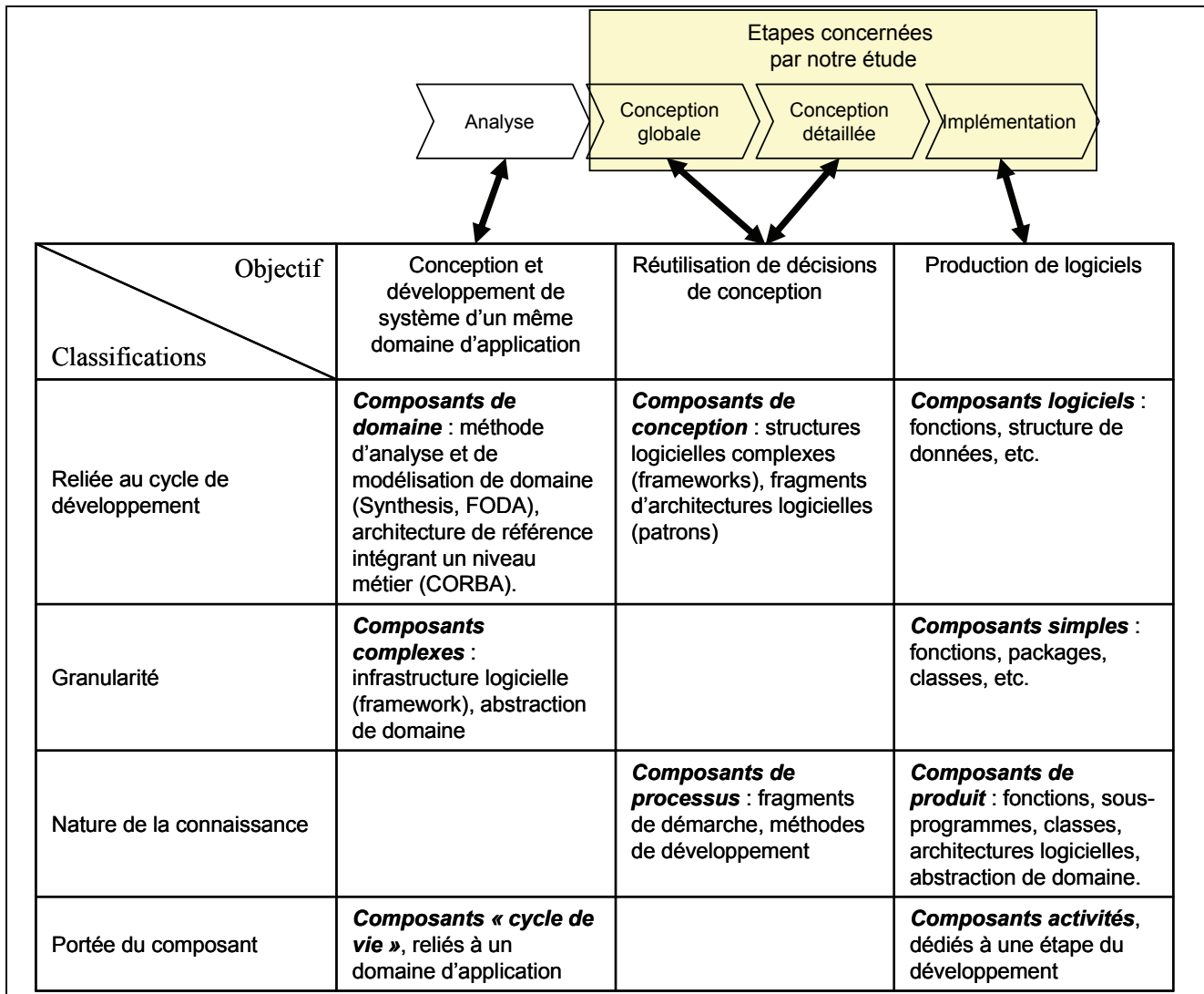


Figure 2 : entités pour l'approche « système d'information par réutilisation »

Comme nous nous focalisons sur les étapes « conception globale », « conception détaillée » et « implémentation » du cycle de développement d'une application logicielle, nous nous focalisons sur les entités concernant l'étape de conception et l'étape d'implémentation. En reprenant les concepts de la première classification (« Reliée au cycle de développement »), nous présentons maintenant plus en détails les composants de conception et les composants logiciels.

2.1.3 Les composants de conception : les patrons

Le patron est le composant de conception majoritairement utilisé dans les travaux que nous avons identifié. Par conséquent, c'est cette entité qui sera considérée dans le cadre de notre étude.

2.1.3.1 Définition

Le concept de patron permet de guider l'utilisateur dans un processus de choix et d'adaptation. Il a été initialisé par C. Alexander qui a défini les premiers patrons de conception dans le domaine de l'architecture des bâtiments pour permettre à des personnes non architectes de construire leurs propres maisons. Réutilisé et enrichi dans différents domaines applicatifs, il est devenu un vecteur privilégié de capitalisation et de partage de bonnes pratiques pour résoudre un problème récurrent. Selon la définition habituellement présentée, « un patron a pour but de décrire avec succès des solutions récurrentes à des problèmes logiciels communs dans un certain contexte, et d'aider les gens à réutiliser des pratiques vraies et résolues » [Schmit, 96]. Cependant, cette définition n'est pas assez précise par rapport à notre étude car elle ne précise ni les solutions récurrentes ni la manière dont l'aide est apportée.

Ainsi, nous précisons cette définition de la manière suivante : « un patron a pour but de décrire les concepts, les scénarios, les solutions conceptuelles et logicielles, propres à la réalisation d'une étape d'un processus »

2.1.3.2 Types de patrons

Parmi les différentes classifications de patrons qui ont été établies et présentées Figure 3 [Saidane, 05], nous retenons celle relative à la portée car elle associe les patrons aux étapes du développement par rapport auxquelles notre étude a déjà été positionnée [Cocquebert, 07a] [Cocquebert, 07b]. Cette classification distingue³ :

- les patrons d'analyse pour aider le concepteur à définir le modèle du besoin lors de l'étape d'analyse [Coad, 95] ;
- les patrons de conception générique [Gamma, 95] ou d'un domaine particulier [Gzara, 00] pour aider le concepteur à identifier les objets, leurs collaborations et la distribution de leurs responsabilités lors de l'étape de conception d'une application logicielle ;
- les patrons d'implémentations (encore appelés « idiomes »), qui décrivent comment implémenter dans un langage donné certains aspects particuliers des composants ou des relations entre eux [Coplien, 96].

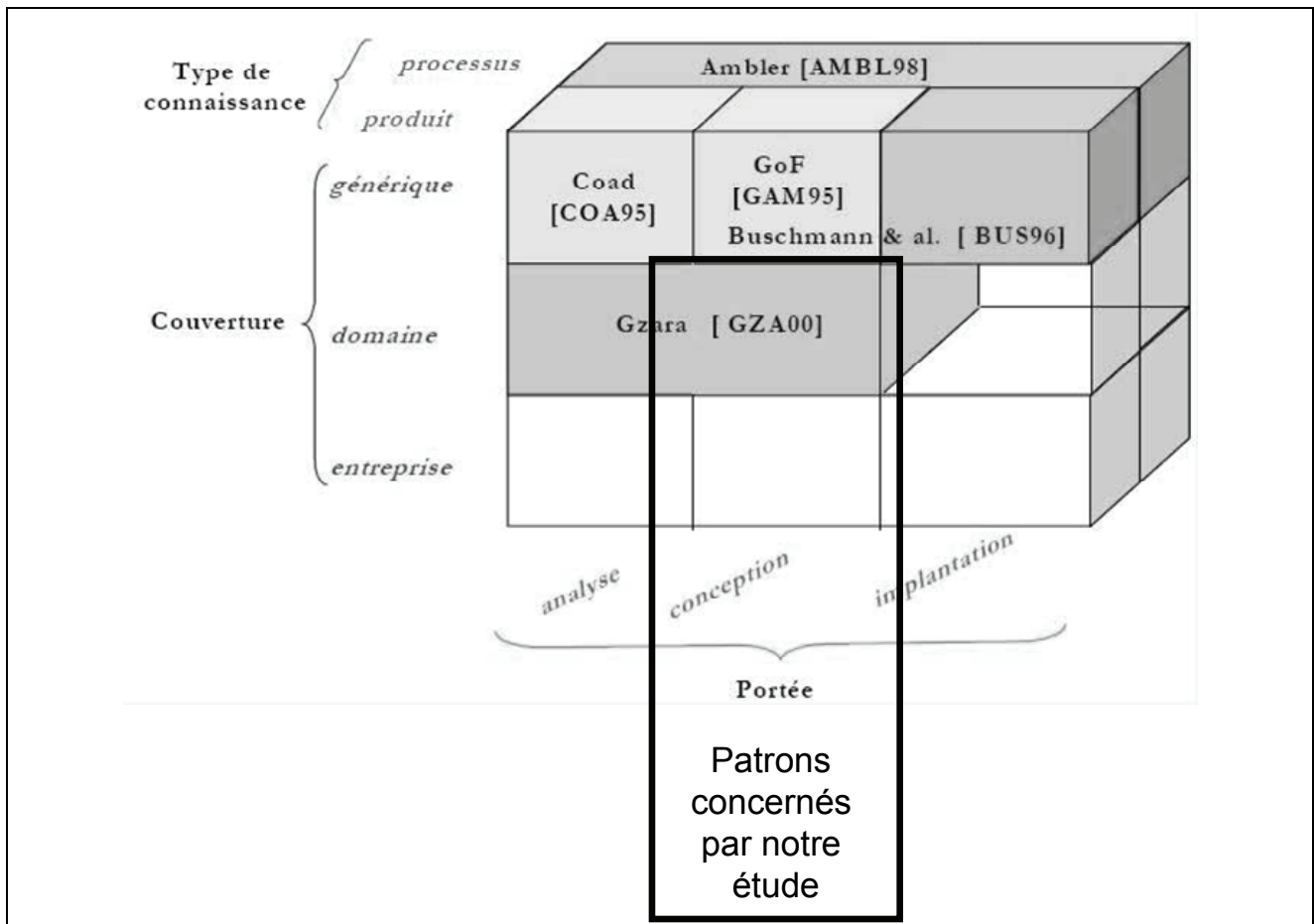


Figure 3 : classification des patrons ([Saidane, 05])

Dans le cadre de notre étude :

- nous nous intéressons aux patrons de conception et plus particulièrement aux patrons du domaine de la conception de sites web car ils sont reliés à l'étape de conception du processus de conception ;
- nous ne considérons pas les patrons d'implémentation malgré le lien avec l'étape d'implémentation car, contrairement aux composants logiciels, ils ne constituent pas une réutilisation logicielle directe en tant que telle.

³ Les références sont indiquées à titre d'exemple et ne représentent pas une liste exhaustive.

2.1.3.3 *Processus d'utilisation d'un patron*

La conception basée sur les patrons conduit généralement à "imiter" plusieurs patrons existants et à intégrer ensuite ces imitations :

- l'« imitation » comporte les différentes étapes suivantes [Oussalah, 99] :
 - identification : elle consiste à consulter un catalogue afin d'identifier le patron pour résoudre le problème à résoudre,
 - adaptation : il s'agit de renommer le patron identifié, d'ajouter, puis de modifier ou de supprimer des attributs et des méthodes sur le diagramme de classe original,
 - implémentation : c'est le développement du code associé au diagramme adapté ;
- l'intégration a pour objectif la suppression des synonymies et homonymies de classes et de propriétés par des unions (remplacement de deux classes par une seule), ou par association des classes issues d'imitations différentes mais modélisant le même concept.

Si l'intégration est réalisée par un concepteur expérimenté, elle conduit à une étape de « création » car une abstraction du résultat de cette activité peut conduire à la création d'un nouveau patron.

Dans la suite de notre travail, nous considérons uniquement les activités d'imitation et d'intégration car elles relèvent de l'ingénierie par réutilisation, contrairement à l'étape de création qui relève de l'ingénierie pour la réutilisation.

Un autre type d'entité concerné par notre étude est le composant logiciel que nous présentons dans la partie suivante.

2.1.4 *Les composants logiciels*

2.1.4.1 *Définition*

Les composants logiciels sont également dénommés COTS (« Component-Off-The-Shelf » ou « composants sur étagère »). A partir de son analyse des multiples définitions qui ont été données des COTS, R. Michel les définit de la manière suivante [Michel, 06] : « les COTS sont des produits logiciels :

- existants en de multiples copies ;
- dont le code source, parfois accessible sous forme « open source », est le plus souvent indisponible ;
- vendus, loués ou fournis gratuitement par un vendeur ;
- ayant des mises à jour périodiques s'accompagnant d'un accroissement des fonctionnalités fournies alors que certaines autres deviennent obsolètes ;
- ayant la capacité de proposer l'un de ses services (ou plus) comme partie indispensable au fonctionnement d'une autre application ».

Cette définition est suffisamment ouverte pour que beaucoup de produits logiciels puissent être considérés comme des COTS, mais elle est également trop large par rapport aux composants considérés par cette étude.

En effet, les composants considérés dans cette étude sont disponibles en « open source » et sont sélectionnés pour les fonctions qu'ils offrent à un instant T. Notre définition est donc la suivante : « un composant est un produit logiciel :

- disponible gratuitement (« open source ») ;
- qui offre des services de haut niveau et disposant de leur propre interface utilisateur (par exemple, un forum de discussion), ou de bas niveau sans interface utilisateur à intégrer au cœur de l'application (par exemple, l'interaction avec une base de données). »

2.1.4.2 *Types de composants*

Parmi les différentes classifications de composants qui ont été présentées dans [Michel, 06] pour affiner sa définition, nous retenons celles qui concernent la conception de systèmes d'informations par réutilisation par rapport à laquelle notre étude a déjà été positionnée [Cocquebert, 07a] [Cocquebert, 07b].

Classification par rapport à la disponibilité du code [Valetto, 95]. Elle distingue les composants :

- « boîtes blanches », où le code source est disponible et peut être modifié en vue de son intégration au sein d'une application logicielle ;

- « *boîtes grises* », qui ne permet pas d'accéder au code source, mais qui met à disposition des interfaces de programmation (API ou « Application Procedural Interface ») pour l'utiliser au sein d'une application logicielle ;
- « *boîtes noires* », où le code source n'est pas disponible et qui se présente uniquement sous forme binaire.

Classification selon le type de services offerts par les composants [Hassine, 02]. Elle distingue les composants :

- « *métier* », qui met en œuvre des fonctions de haut niveau afin de les intégrer au sein d'un développement logiciel (forums de discussion, service d'authentification, etc.) ;
- « *technique* », qui met en œuvre des fonctions de bas niveau afin de les intégrer au sein d'un développement logiciel (interaction avec une base de données, mise en page HTML, etc.).

En recoupant notre définition et ces deux classifications, notre étude concerne les composants « boîtes blanches », et considère les composants métiers et les composants techniques.

2.1.4.3 Processus d'utilisation d'un composant

L'utilisation des composants relève de la « Conception de système Basé sur la Réutilisation de Composants » (CBSE ou « Component-based Software Engineering » [Cnkovic, 02]), qui met plus l'accent sur la réutilisation des COTS que sur le processus de développement. Cela se traduit par une conduite parallèle du processus de développement d'un système (PES) et d'un processus d'évaluation des composants (PEC) qui s'influencent mutuellement pour définir les spécifications du système. Cette relation étroite entre les deux processus est présentée Figure 4 d'après leurs présentations dans [Cnkovic, 02].

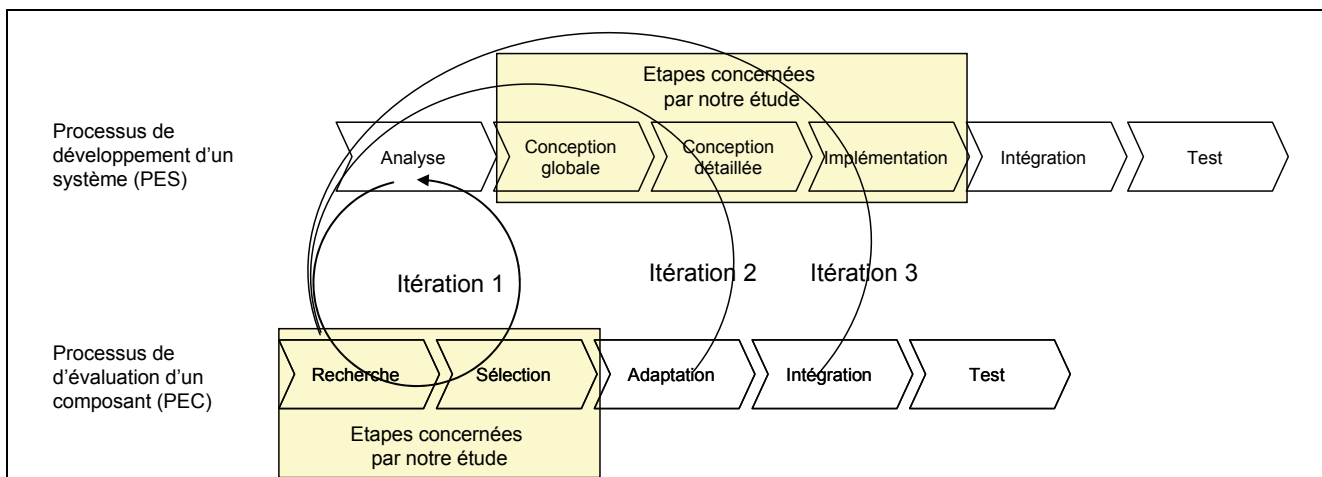


Figure 4 : relations entre les processus de développement d'un système et d'évaluation d'un composant

Au cours du PEC, les étapes de recherche et de sélection consistent à identifier un ou plusieurs composants pour satisfaire les besoins fonctionnels et non-fonctionnels identifiés dans l'étape d'analyse du PES (itération 1). Cependant, tous les besoins peuvent ne pas être mis en œuvre par les composants sélectionnés, ou ceux-ci peuvent apporter des fonctionnalités non identifiées. C'est pourquoi et comme le représente l'itération 1 de la Figure 4, le résultat de ces étapes peut influencer la définition des spécifications de l'étape d'analyse du PES.

Au cours du PEC, les étapes d'adaptation et d'intégration consistent respectivement à adapter le composant aux besoins identifiés et à l'intégrer dans l'architecture du développement de l'application. Selon l'effort d'adaptation et d'intégration du composant, ou selon la nécessité de modifier l'architecture pour recevoir le composant, la réalisation de ces deux étapes peut remettre en cause le choix du composant et peut nécessiter l'itération 2 ou l'itération 3 pour rechercher et sélectionner un ou plusieurs nouveaux composants.

Du fait de l'importance des étapes de recherche et de sélection dans le PEC et de leur relation étroite avec l'étape de conception globale du PES, nous avons concentré notre étude sur l'étape de recherche et de sélection des composants pour satisfaire un besoin identifié. En conséquence, les étapes d'adaptation et d'intégration, orientées « développement pur », ainsi que l'étape de test ne seront pas considérées.

Nous avons présenté dans cette première partie l'ingénierie de la réutilisation, premier domaine de recherche dont notre étude relève. La partie suivante décrit le deuxième domaine qui nous concerne : l'ingénierie dirigée par les modèles

2.2 Ingénierie dirigée par les modèles

Les travaux de recherche relevant de l'ingénierie de la réutilisation ont produit un nombre important de modèles technologiques et techniques, mais bien souvent incapables de communiquer entre eux. C'est la raison pour laquelle le CNRS a créé en 2003 l'action spécifique MDA (« Model-Driven Architecture ») avec pour objectif de mettre le modèle au centre des différentes étapes du processus de développement et de découpler la modélisation des considérations techniques. Le résultat des travaux de cette action spécifique a été de formaliser un cadre méthodologique et technologique appelé « Ingénierie Dirigée par les Modèles » (IDM) dont nous résumons maintenant les points essentiels [Favre, 06a] [Rieu, 07].

L'IDM est défini par une superposition de quatre niveaux de modélisation, appliqués à la dimension « produit » et à la dimension « processus » (cf. Figure 5) :

- le niveau M0 définit un système ;
- le niveau M1 définit une modélisation (abstraction) de ce système, décrivant les règles permettant de concevoir ce système ;
- le niveau M2 définit les règles qui permettent de décrire un ensemble de modèles ;
- le niveau M3 définit les règles d'écriture de méta-modèles.

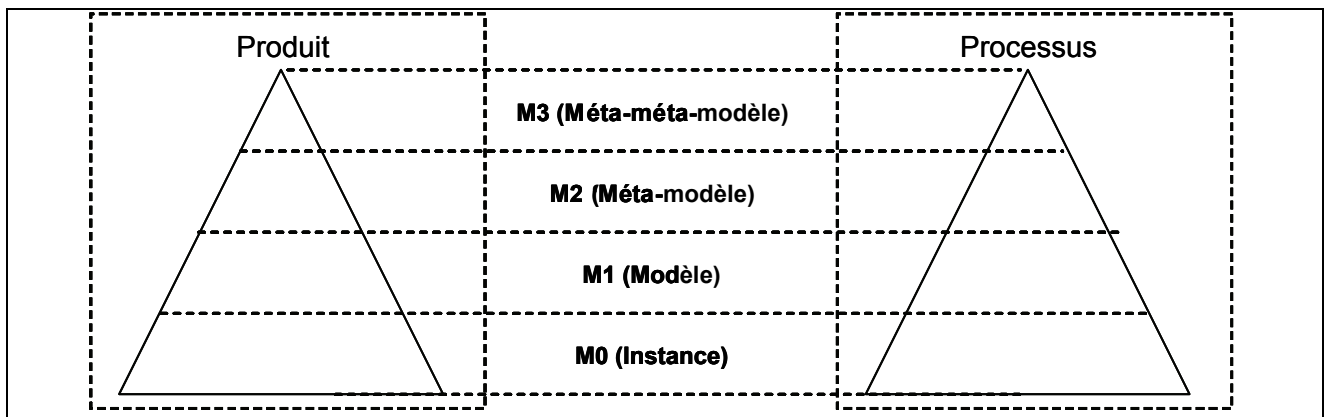


Figure 5: niveaux de modélisation de l'IDM (à partir de [Favre, 06a] et [Rieu, 07])

L'approche de l'IDM est plus une évolution qu'une révolution car elle prend en compte les travaux réalisés dans le cadre de l'ingénierie de la réutilisation et utilise à chacune des étapes d'un processus la technologie la plus adaptée. Sa spécificité est de mettre la modélisation au centre du processus, afin de permettre la génération automatique de code à partir de transformations successives et itératives (situées dans la dimension « processus ») appliquées sur les modèles (situées dans la dimension « produit »).

Dans le cadre de notre étude, ce sont les niveaux de modélisation M0 et M1 de l'IDM que nous considérons car ils constituent une zone commune avec l'ingénierie de la réutilisation. Celle-ci vise à modéliser l'exploitation des entités de réutilisations pour faciliter leur intégration au sein d'un système d'information par réutilisation (approche ascendante), alors que l'ingénierie dirigée par les modèles vise à générer les entités utilisées à partir de leur modélisation (approche descendante).

Après avoir présenté l'ingénierie de la réutilisation et l'ingénierie dirigée par les modèles, nous abordons le troisième domaine dont notre étude relève : l'ingénierie du web.

2.3 Ingénierie du web

Depuis leur apparition en 1996 jusqu'en 2002 où le domaine est devenu mature, la nature, l'objectif et les acteurs d'un site web ont considérablement évolués [Bordage 08]. Les premiers sites diffusaient uniquement de l'information statique et étaient réalisés de manière artisanale par peu d'acteurs. Désormais, ils sont devenus un vecteur privilégié d'informations dynamiques (sites marchands, institutionnels, de marques, etc.), de dialogue entre les utilisateurs (sites communautaires, intranet, collaboratifs, etc.) et sont réalisés par des

équipes projets faisant intervenir de multiples compétences (programmeurs, ergonomes, référenceurs, chefs de projet, graphiste, etc.). Désormais, un site web est caractérisé par la complexité des informations gérées et des services proposés (site de commerce électronique, intranets, ...) et par la prise en compte de l'utilisateur pour lui présenter des informations qui lui soient adaptées. Les travaux de recherche reliés à ce domaine ont conduit à l'émergence de différents axes de recherche que nous abordons ci-dessous

« *Méthodes, modèles et outils de conception de sites web* ». L'objectif est de définir méthodes, modèles et outils pour la conception de sites web en adaptant les résultats de l'ingénierie du logiciel à la nature plus ouverte et plus flexible du web d'une part et de l'hypermédia d'autre part [Gnaho, 00] [Halin, 05] [Villanova-Oliver, 02]. Dans cette catégorie et parmi les différentes méthodes qui ont été définies, on peut citer :

- RMM pour la conception d'applications hypermédia [Isakowitz, 95] ;
- OOHDM qui introduit une démarche de conception orientée objet [Schwabe, 96] ;
- WSDM qui considère en entrée les besoins utilisateurs plutôt que les données [De Troyer, 98] ;
- WebML qui définit un langage de modélisation graphique [Ceri, 00].

« *Adaptation à l'utilisateur* ». Du fait de l'ouverture du web sur le monde, les utilisateurs présentent des buts, des connaissances et des compétences divers, ont des préférences variées en termes de présentation, ou encore accèdent aux sites web dans des conditions matérielles différentes. L'objectif est ici de doter les sites web de capacités d'adaptations à leurs utilisateurs afin qu'ils soient satisfaits du site web par rapport au contenu affiché, à la navigation disponible, à la présentation des informations. Dans cette catégorie, on peut citer les travaux de :

- [Gnaho, 00] pour son cadre méthodologique pour l'ingénierie des systèmes d'informations web adaptatifs ;
- [Villanova-Oliver, 05] pour son modèle d'accès progressif mis en œuvre au sein de KIWIS ;
- [Jrad, 08] pour son modèle utilisateur dans un contexte de navigation dans le cadre du projet Eiffel.

« *Web sémantique* ». Suite à l'explosion du nombre de sites web, les utilisateurs ont souvent des difficultés à trouver une information pertinente par rapport à leurs besoins. En se basant sur les standards actuels du web, l'objectif des travaux du web sémantique est de renforcer les informations disponibles par des données formelles compréhensibles par les machines, de manière à optimiser les recherches sur le web. Les travaux ont conduit entre autres à la proposition du langage RDF par le W3C [W3C, 99], au formalisme des « Topic Maps » par l'ISO [IEC, 99], au langage de description des ontologies OWL du W3C⁴, etc., dont le XML et les flux RSS sont une de leurs applications.

Dans le cadre de cette étude, nous considérons les méthodes, modèles et outils de conception de sites web.

Nous présentons maintenant le cadre applicatif de notre étude, c'est-à-dire les types de sites web que nous avons considérés pour la réaliser.

3 Cadre applicatif de notre étude

3.1 Les applications web

Du point de vue de la terminologie, on distingue les systèmes web et les applications web :

- Les systèmes web désignent des outils et infrastructures de production et de mise en œuvre ;
- les applications web sont mises en œuvre à l'aide des systèmes web.

Les applications web sont donc ce que l'on appelle couramment les « sites web » dont la conception :

- n'est plus une activité artisanale et empirique, mais un domaine d'application à part entière de l'ingénierie logicielle [Gnaho, 00] ;
- est désormais menée sous la forme de projets web faisant intervenir de multiples acteurs aux compétences diverses : concepteurs, ergonomes, référenceurs, chefs de projet, graphistes, etc. [Bordage, 06],

Par rapport à ces différents acteurs, notre étude concerne les concepteurs de sites web.

⁴ <http://www.w3.org/2004/OWL/>

3.2 Typologie des sites web

En considérant le niveau de complexité des données et des applications, [Gnaho, 00] distingue quatre catégories d'applications web (cf. Figure 6) [Cocquebert, 08a] [Cocquebert, 08b] :

- les sites de présence, caractérisés par une relative faible complexité de données et des applications (sites de conférence) ;
- les sites orientés services, où l'accent est mis sur la mise à disposition de services évolués tels les moteurs de recherches, une messagerie gratuite, etc. ;
- les sites catalogues, où l'accent est mis sur l'organisation et la maintenance d'une structure de données complexe, tels les sites académiques ;
- les systèmes d'informations web, qui proposent aux utilisateurs d'accéder à un système d'informations à travers le web, tels les sites de commerce électronique, les intranets, etc.

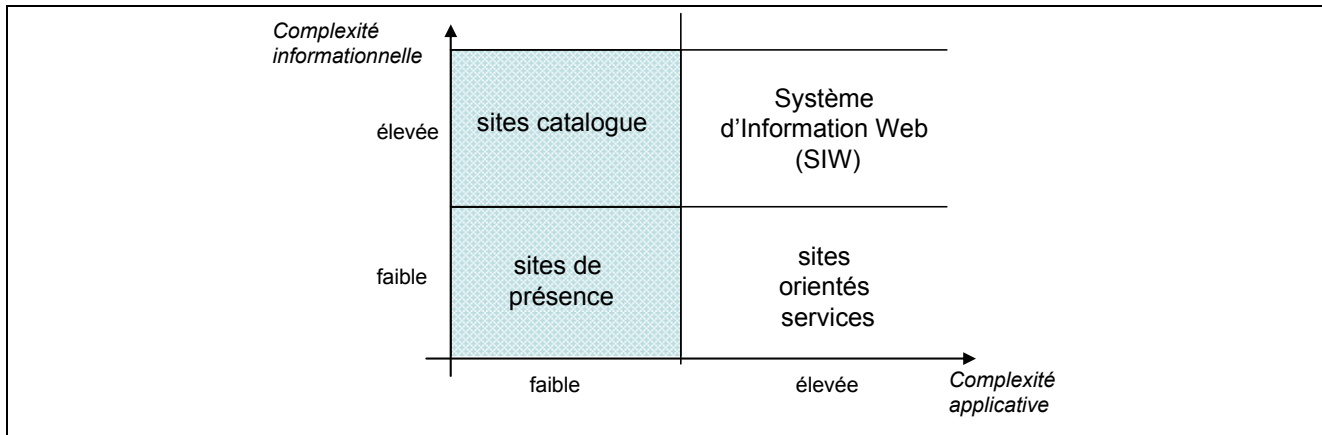


Figure 6 : types d'applications web ([Gnaho, 00])

De par notre expérience dans la conception et la réalisation de sites web⁵, cette étude a été menée en considérant les sites de présence et les sites catalogues.

Dans la suite de ce document, l'utilisation du terme « **site web** » désigne les sites de présence et les sites catalogues.

Nous avons défini les limites de notre étude en précisant les étapes du cycle de développement qui nous concerne, les domaines de recherche dont notre étude relève, et le cadre applicatif de nos résultats. L'objet de la partie suivante est de présenter les fonctionnalités à mettre en œuvre.

4 Fonctionnalités à mettre en œuvre pour la conception d'un site web

Pour aider un concepteur de sites web, notre expérience a permis d'identifier les quatre besoins suivants, que nous présentons et justifions dans la suite de cette partie en tant que fonctionnalités à mettre en œuvre [Cocquebert, 08b] :

- Assister la réalisation du processus de conception, surtout dans le cas où ce concepteur est débutant ;
- Identifier et faciliter la réutilisation de solutions de conception afin d'accélérer la définition du contenu du site web ;
- Identifier et faciliter le choix de composants logiciels de manière rationnelle et objective ;
- Accélérer la mise en place d'un site web.

4.1 Assister la réalisation du processus de conception

L'apprentissage du processus de développement d'un site web peut être basé sur la lecture de livres, la consultation de sites internet, le suivi de formation, etc. mais, comme nous le verrons au chapitre 2, il n'y a pas d'implémentation explicite du cycle de développement dans les outils de développement : ils permettent

⁵ cf. <http://www.univ-valenciennes.fr/sp/cocquebert/>

la conception avec des degrés variés d'assistance, mais le concepteur doit connaître les étapes du processus pour que l'assistance soit efficace.

Pour mettre en œuvre cette fonctionnalité, nous proposons la réalisation des points suivants :

- guider le concepteur vers les étapes valides du processus qui s'offrent à lui, compte tenu du résultat de l'étape qu'il vient de terminer ;
- aider à la réalisation des étapes en indiquant les informations qui les concernent (objectif, entrées, résultats, méthodes, outils supports, etc.).

4.2 Identifier et faciliter la réutilisation de solutions de conception

Il est habituel de constater que la réutilisation de solutions de conception dépend de l'expérience du concepteur, et qu'il est difficile de formaliser cette expérience afin de faciliter sa réutilisation par d'autres concepteurs. Or, l'expérience montre que des sites web de même nature ont des menus similaires, des contenus similaires qui sont présentés de manière similaire, et que des familles de sites web peuvent être définies. Cependant, il n'existe pas de formalisation de cette expérience de conception de sites web.

Pour mettre en œuvre cette fonctionnalité, nous proposons la réalisation des points suivants :

- identifier et proposer des solutions de conception déjà utilisées dans des sites web en production ;
- faciliter la réutilisation des solutions de conception retenues par l'utilisateur parmi les propositions.

4.3 Identifier et faciliter le choix de composants logiciels

L'identification et le choix d'un composant logiciel est une tâche fastidieuse pour les concepteurs. En effet, il n'existe pas de source d'information ou de référentiel qui permettent de connaître :

- les différents composants existants pour mettre en œuvre une fonction ;
- les retours d'expérience formalisés des utilisateurs.

Le besoin d'un composant logiciel conduit donc invariablement à un questionnement sur les forums ou des listes de discussion pour connaître l'existence de composants et pour avoir des retours d'expérience, ou à consulter des sites internet de comparaison. Au mieux, cela peut conduire assez rapidement à réutiliser toujours les mêmes composants pour ne pas consommer du temps en recherche, au pire à réaliser des développements spécifiques pour chaque besoin sans chercher à identifier des composants existants.

Pour mettre en œuvre cette fonctionnalité, nous proposons la réalisation des points suivants :

- identifier des composants logiciels existants déjà utilisés dans des sites web en production ;
- faciliter le choix d'un composant utilisateur par la mise à disposition d'informations sur les fonctions des composants et sur des retours d'expérience d'utilisateurs.

4.4 Accélérer la mise en place d'un site web

Lors de la mise en place d'un site web, le concepteur exécute invariablement les mêmes tâches répétitives : définition de dossiers pour structurer l'arborescence des fichiers, copie de fichiers de méthodes ou de classes génériques issus de conceptions antérieures, génération de base de données à partir des choix de conception, etc.

Pour mettre en œuvre cette fonctionnalité, nous proposons la réalisation des points suivants :

- générer automatiquement des structure de données à partir de choix de conception ;
- définir une architecture d'accueil pour accélérer la conception d'un nouveau site web.

En considérant ces quatre fonctionnalités à mettre en œuvre, nous présentons maintenant nos objectifs en prenant en compte les différents choix que nous avons mis en avant dans ce chapitre.

5 Objectifs

Le premier objectif de notre étude est de proposer une méthode d'aide à la conception de sites web favorisant la réutilisation d'expériences de conception et de composants logiciels, appelée « WISDOM » ; le deuxième objectif est de valider cette méthode par l'utilisation d'une mise en œuvre appelée « WISDOM Tool » pour concevoir un site web réel.

Comme le montre la Figure 7 :

- l'entrée de WISDOM est le cahier des charges du site web à concevoir ;

- le résultat est l'architecture d'accueil du nouveau site web, c'est-à-dire que l'architecture logicielle du site web est opérationnelle et que le concepteur a la possibilité de programmer facilement des comportements spécifiques :
 - en personnalisant les fichiers et les structures de données générés ;
 - en utilisant des composants métiers ou des composants techniques intégrés à l'architecture ou en intégrant des composants externes.

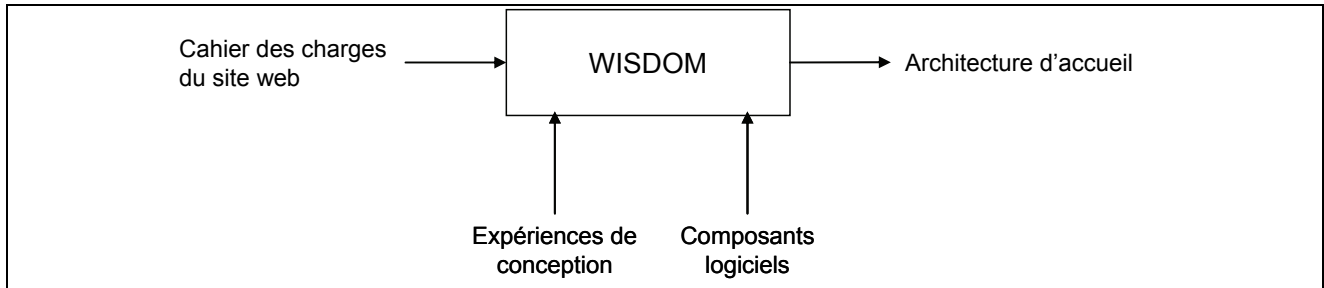


Figure 7 : utilisation de WISDOM

En se référant à notre présentation de l'étape d'implémentation (cf. page 23), celle-ci n'est pas entièrement réalisée par WISDOM (cf. Figure 8) car l'architecture logicielle est générée, mais les développements spécifiques sont laissés à la charge du concepteur.

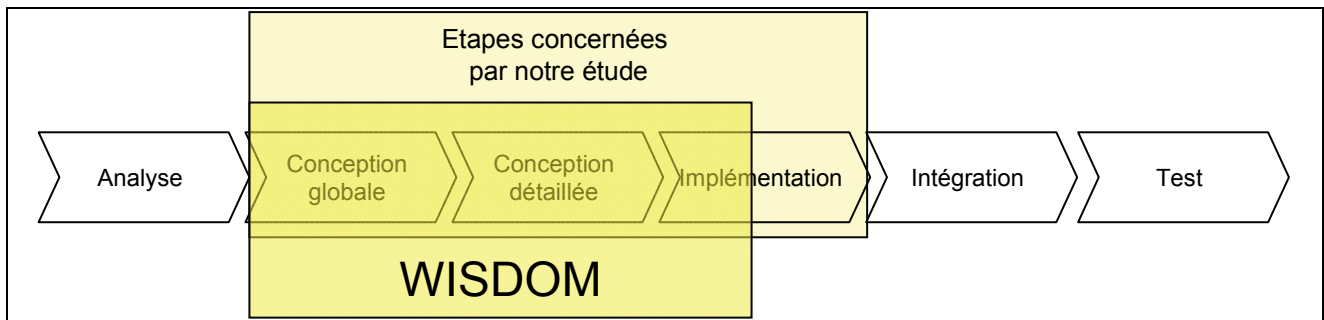


Figure 8: périmètre de l'utilisation de WISDOM

Ainsi, d'un point de vue global, notre objectif n'est pas de proposer une méthode de génération automatique du code d'un site web, mais une méthode qui facilite :

- la conception d'un site web à partir de la réutilisation de travaux de conception et de composants logiciels ;
- la mise en place de son architecture logicielle et son adaptation au cahier des charges.

L'objectif du chapitre suivant est d'analyser l'état de l'art par rapport à ce contexte de travail et à notre objectif global.

Chapitre II : Etat de l'art

L'objectif de ce chapitre est d'analyser l'état de l'art par rapport au contexte de notre étude. Structuré selon les fonctionnalités à mettre en œuvre, nous avons accompagné ces analyses de graphiques issus de publications ou issus de notre réflexion. Dans ce cas, nous avons utilisé les dessins présentés Tableau 1.






	Utilisateur d'un système, d'un processus, etc.
	Diagramme de classe
	Code proposé en exemple (dans la formalisation d'un patron), ou généré automatiquement par un processus
	Automatisation d'un processus
	Indication d'un retour d'information à l'utilisateur s'il est à l'origine de l'action, accompagné du type du retour lui-même.

Tableau 1 : légende des graphiques utilisés dans les illustrations des analyses

1 Fonctionnalité « Assister la réalisation du processus de conception »

Nous analysons dans cette partie les propositions pour guider le concepteur dans un processus et pour l'aider à réaliser les étapes d'un processus [Cocquebert, 08b].

1.1 Guider le concepteur

Pour guider l'utilisateur dans la mise en œuvre d'un environnement de développement basé sur les composants, [Parson, 06] utilise l'approche patron et définit :

- un processus « papier » (Figure 9) pour définir les étapes à réaliser pour adapter le patron architecture (étape 2 de la figure) et pour y intégrer des composants externes hétérogènes (étape 4 de la figure) ;
- les différents acteurs et les compétences nécessaires à la réalisation des étapes du processus (Figure 10) ;
- une application canonique pour aider à la mise en œuvre d'un environnement de développement à partir du patron architecture.

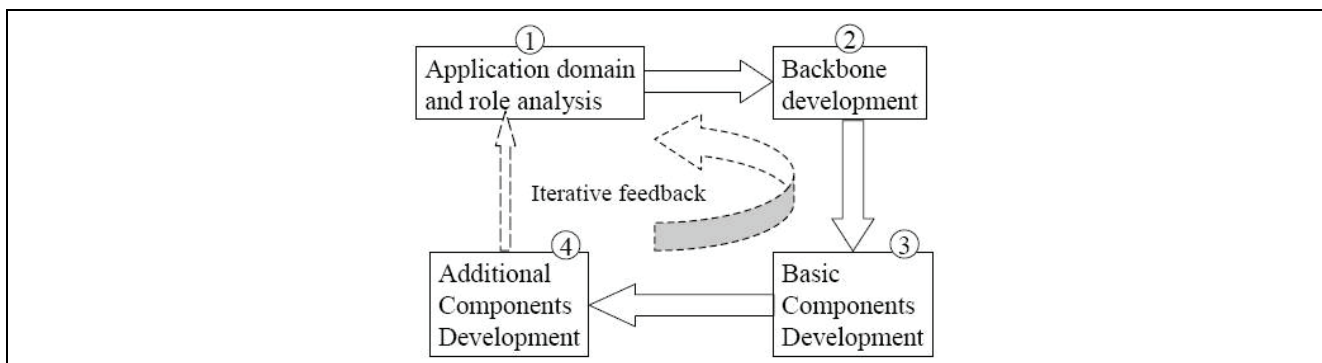


Figure 9 : processus défini pour guider le développement ([Parson, 06])

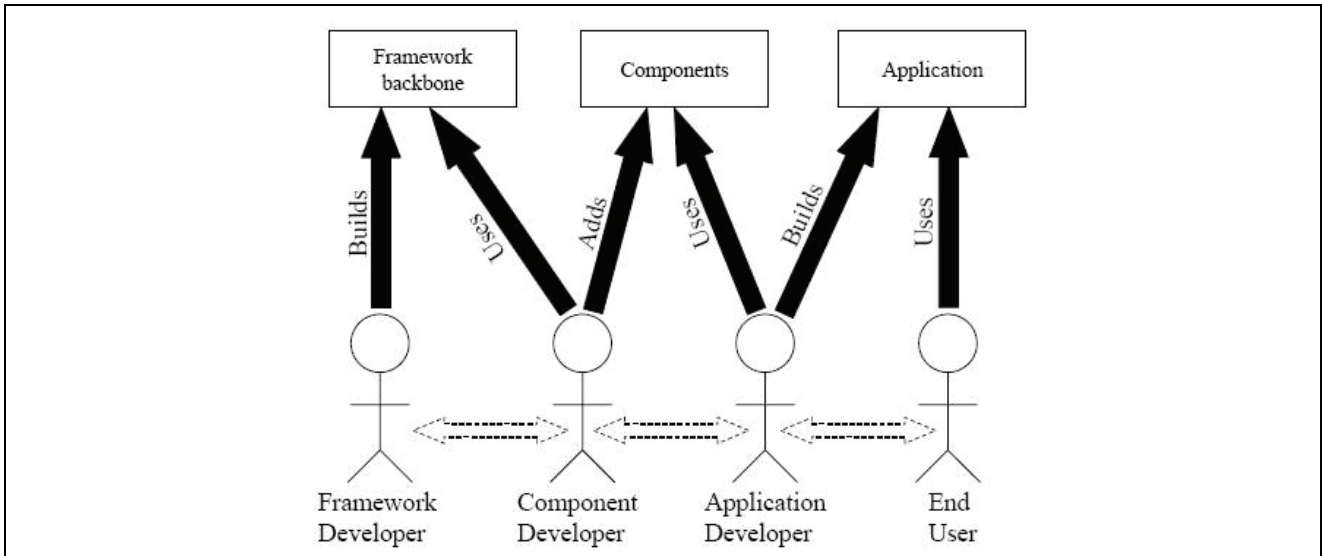


Figure 10 : acteurs impliqués par le processus ([Parson, 06])

Non informatisé, le processus est suivi par les acteurs à partir de sa spécification papier, et le(s) concepteur(s) peu(ven)t mener simultanément le déroulement du processus sur plusieurs composants en même temps sans être dans les mêmes étapes. L'utilisation de ce processus guide la personnalisation et l'extension du patron architecture par intégration de composants hétérogènes.

Les intérêts de cette approche sont de montrer sur un exemple l'utilisation d'un processus et de définir avec précision les étapes du processus et les compétences des acteurs concernés. Cependant, l'absence d'informatisation du processus peut conduire à un manque de rigueur dans le suivi de celui-ci lors de la conception. Le processus peut alors être plus long, ou le résultat non conforme à ce qui aurait dû être si le processus avait été informatisé.

Dans l'ingénierie du web, des méthodes ont été définies pour guider le processus de développement des sites, en s'inspirant parfois des travaux sur le développement d'hypermédias. Le Tableau 2, construit à partir des présentations des méthodes de conception qui sont faites par C. Gnaho [Gnaho, 00] et M. Villanova [Villanova-Oliver, 02], indique les processus définis par ces différentes méthodes.

	Identification des besoins	Identification des utilisateurs	Navigation	présentation	Modèle de données	Choix implémentation	implémentation
RMM			3	2 – 4 – 5	1	6	
Hera	1		4	3	2		
OOHDM (2000)	1		3	4	2		5
WSDM	1	2	4	5	3 - 6		7
AWIS-M	1	1	3-4		2		
UWE	1		3-4	5	2		
WebML	1	1	3	4	2		

Tableau 2 : chronologie des processus des méthodes de conception de sites web

Si ces méthodes considèrent toutes les mêmes dimensions dans la conception d'un site web (« navigation », « présentation », « données »), [Lang, 02] montre cependant que, malgré leur existence, ces méthodes sont très peu utilisées, notamment parce qu'elles sont méconnues hors du monde universitaire et à cause d'une

absence de guide explicite des utilisateurs par leurs outils associés. Nous avons vérifié nous-mêmes ce dernier point en utilisant des outils associés à certaines de ces méthodes (VisualWADE⁶ pour la méthode OO-H, ArgoUWE⁷ pour la méthode UWE, Webratio⁸ pour la méthode WebML), mais nous l'avons également vérifié sur les logiciels diffusés par des éditeurs de logiciels (Zend Framework⁹ et PHPEclipse¹⁰). En utilisant ces outils, l'utilisateur dispose des outils pour réaliser chacune des étapes ci-dessus et ses actions sont contrôlées, mais il n'est pas guidé de manière explicite dans la démarche de conception. En fait, ces outils sont efficaces pour aider à la conception à condition de savoir comment les utiliser.

De son côté, G. Halin pense que la non-utilisation de ces méthodes est probablement due à leur inadéquation aux besoins des utilisateurs [Halin, 05]. En effet, si, les processus du Tableau 2 semblent différents au premier abord, leur analyse globale montre qu'ils sont globalement basés sur le processus synthèse suivant :

1. identification des besoins
2. définition du modèle de données
3. définition de la navigation
4. définition de la présentation

Or, la deuxième étape des concepteurs n'est pas la définition des données comme le préconise ce « processus synthèse », mais la projection des besoins de l'utilisateur sur une interface graphique, étape 4 de ce « processus synthèse » [Halin, 05] [Gnahou 00].

Ces méthodes de conception de sites web sont donc intéressantes, mais leur processus n'est pas adapté à la réalité et les outils associés nécessitent une expertise pour pouvoir être utilisés de manière efficace.

Ainsi, pour guider le concepteur dans la réalisation d'un processus, il est préférable que ce guide soit informatisé pour que le suivi soit correct et que les étapes soient adaptées à la réalité du processus afin qu'il soit réellement utilisé.

Après avoir vu comment guider un processus, la partie suivante présente comment aider à la réalisation des étapes de ce processus.

1.2 Apporter une aide à la réalisation des étapes du processus

Cette capacité à fournir des informations à l'utilisateur sur les buts, les entrées, les résultats, etc. d'une étape d'un processus est surtout considérée par l'ingénierie de la réutilisation de composants de conception, c'est-à-dire l'approche patron. Pour cela, un patron est formalisé par un ensemble de rubriques, dont le contenu peut être un diagramme de classe, un exemple d'implémentation, etc. (Figure 11)

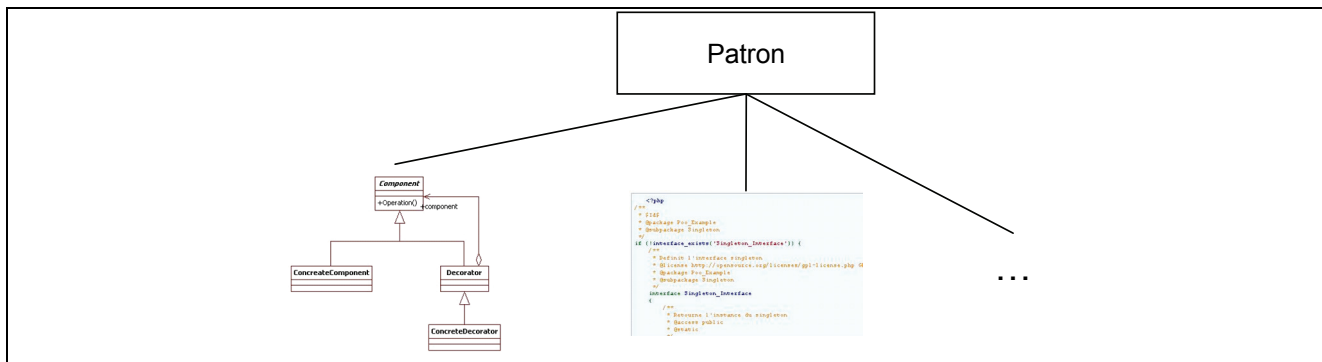


Figure 11 : formalisation d'un patron

⁶ <http://www.visualwade.com/>

⁷ <http://www.pst.informatik.uni-muenchen.de/projekte/uwe/toolargoUWE.html>

⁸ <http://www.webratio.com/portal/homePage.do?sessionId=B1F833A20A89E88C67400FECD86B0346>

⁹ <http://framework.zend.com/>

¹⁰ <http://www.phpeclipse.de/>

La première manière de formaliser les informations dans les patrons est de les exprimer en utilisant uniquement le langage naturel. C'est une solution marginale car elle est prétexte à une utilisation non rigoureuse des informations, mais c'est celle qui a été retenue dans [Lukosch, 06] pour formaliser des patrons orientés utilisateurs (l'auteur parle de « patrons de haut niveau orientés utilisateurs »). L'objectif de ces patrons est d'informer les futurs utilisateurs d'une application groupware sur ses objectifs, ses bénéfices, etc., et de les guider dans l'expression de leurs besoins réels par rapport à ce type d'application afin de faciliter sa conception par la suite. Même si cette contribution correspond à l'étape « analyse » du processus de conception, nous la citons ici car les auteurs ont couplé ces « patrons orientés utilisateurs » avec des « patrons orientés développeurs » pour la conception de l'application et que nous présentons plus loin.

En fait, la solution généralement adoptée consiste à utiliser le langage naturel et des représentations formalisées pour définir des patrons et de rassembler ensuite ces patrons dans des catalogues thématiques. Parmi les différents catalogue disponibles, on peut citer ceux définis dans :

- [Fowler, 97] pour la modélisation et le traitement d'événements récurrents (cf. une traduction du langage patron dans [Fonte, 97]) ;
- [Gamma, 95] pour la conception détaillée d'applications logicielles ;
- [Guerrero, 01] et [Licea, 00] pour la conception d'application collaborative,
- [Buschmann, 96] pour la conception globale ;
- [Jezequel, 99] pour la conception d'architectures particulières.

L'intérêt de ces catalogues est de formaliser des informations précises d'un domaine, mais la limite de leur utilisation est la présentation de ces informations de manière souvent abstraite nécessitant une grande expérience dans le domaine concerné (cf. la définition du patron « composite » dans [Gamma, 95]).

Une autre limite à l'utilisation de ces catalogues est qu'elle a conduit à la définition de nombreux formalismes, non homogènes dans le nombre et le contenu des rubriques et souvent fermés. C'est pourquoi le Laboratoire d'Informatique de Grenoble (LIG¹¹) a proposé le formalisme P-Sigma [Conte, 01a], dont nous reprenons ici les différents objectifs qui ont conduit à sa définition :

- proposer de nouvelles rubriques facilitant la réutilisation de patrons et permettant de mieux les organiser ;
- définir un formalisme commun pour exprimer des fragments de modèle et des fragments de démarche, afin d'uniformiser l'expression de patrons produits et de patrons processus ;
- extraire une sémantique commune à certains formalismes proposés dans la littérature et permettre ainsi de construire un catalogue dans lequel seraient décrits des patrons issus de catalogues différents mais appartenant à un même domaine.

Ce formalisme, présenté en Annexe 2, est composé de trois parties, regroupant chacune des rubriques composées d'un ou de plusieurs champs dont le contenu peut être un texte ou un diagramme UML :

- la partie « interface » rassemble les rubriques pour sélectionner un patron dans le catalogue ;
- la partie « réalisation » rassemble les rubriques pour résoudre le problème soulevé par le patron ;
- la partie « relation » rassemble les rubriques pour indiquer les relations d'un patron vis-à-vis d'autres patrons du catalogue.

Ainsi, l'approche patron a la capacité d'apporter une aide à l'utilisateur, mais même si cela nécessite d'avoir une expertise dans le domaine, l'aide est plus efficace si les patrons sont informatisés.

2 *Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »*

Nous analysons dans cette partie les propositions pour faciliter l'identification et la réutilisation de solutions de conception [Cocquebert, 08b], dont nous avons vu dans la partie précédente qu'elles étaient majoritairement formalisées au sein des patrons.

¹¹ <http://www-lsr.imag.fr/sigma.html>

2.1 Identifier des solutions de conception

Depuis de nombreuses années, l'identification conjointe d'un patron et d'une solution de conception pour résoudre un problème est réalisée en consultant la formalisation des patrons dans les catalogues. Comme nous l'avons déjà évoqué dans la partie précédente, le grand nombre et la diversité des catalogues permettent de trouver celui qui correspond à la thématique recherchée, mais l'abstraction de la formalisation nécessite une expérience confirmée dans l'abstraction du problème réel pour identifier un patron et une solution qui y répond.

Pour résoudre ces difficultés d'identification et d'utilisation manuelle d'un patron, une première possibilité consiste à informatiser une recherche de patrons dans les catalogues disponibles (Figure 12).

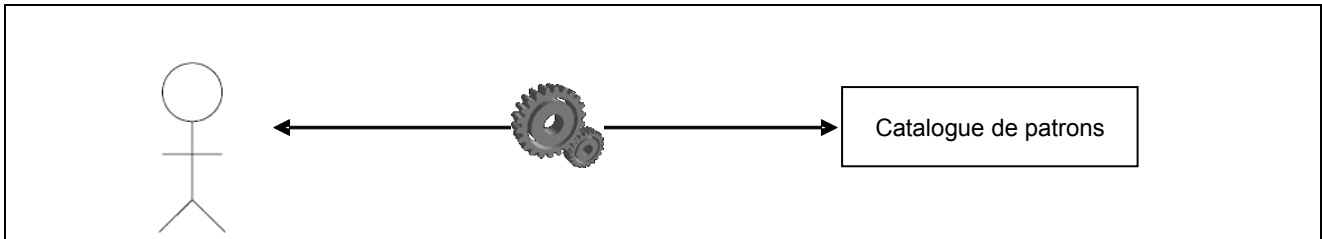


Figure 12 : informatiser la recherche dans un catalogue de patrons

Cependant, [Conte, 01a] a montré que cette tâche était complexe car certains formalismes répartissaient des informations dans plusieurs rubriques. Considérant en outre l'absence d'homogénéité des formalismes en terme de nombre et de degré de détails des rubriques proposées et le manque d'ouverture relativement important de chaque formalisme, le LIG¹¹ a proposé AGAP (Atelier de Gestion et d'Application de Patrons) pour définir, rechercher et utiliser des patrons définis à l'aide de P-Sigma [Conte, 01b]. Utilisé dans différents projets industriels, AGAP et P-Sigma ont montré l'intérêt de permettre à l'ingénieur de patrons de définir de manière interactive des patrons, et à l'ingénieur d'application de rechercher et d'imiter des solutions de conception. Pourtant, de notre point de vue, la recherche reste à l'initiative de l'ingénieur d'application et nécessite toujours une expertise dans l'approche patron pour identifier un patron et imiter une solution de conception.

Une autre possibilité pour résoudre les difficultés d'identification et d'utilisation d'un patron est de se placer dans le cadre de la conception d'une famille d'application. C'est le cas des contributions que nous avons retenu par rapport à la conception d'applications groupware :

- [Lukosch, 06] complète les patrons orientés utilisateurs » (que nous avons présenté § 1.2 p. 39) par des « patrons de bas niveau orientés développeurs », dont les rubriques se focalisent sur la proposition de solutions de conception pour le développement d'une application groupware. Dans cette solution, le concepteur identifie encore les patrons et les solutions par une consultation manuelle du catalogue, mais la situation diffère de la précédente car, dédié à un domaine, le catalogue comporte un nombre de patrons beaucoup plus faible. Par conséquent, sa consultation pour identifier un patron par rapport au problème à résoudre sera plus facile et plus rapide.
- [Guerrero, 01] et [Licea, 00] proposent un système patron dans lequel l'application est décomposée en modules fonctionnels, dont les fonctions sont reliées à quatre patrons, chacun étant dédié à leur mise en place dans l'application. Dans cette solution, l'utilisateur choisit ses solutions en identifiant dans les modules fonctionnels les fonctions de l'application à concevoir, ce qui permet d'identifier et de proposer automatiquement les solutions de conception correspondantes.

Cependant, ces deux utilisations de catalogues propres à un domaine sont toujours manuelles et limite le bénéfice de disposer d'un catalogue restreint pour identifier un patron plus facilement.

Ainsi, des solutions ont été proposées pour informatiser l'identification de patrons dans un catalogue, mais une expertise dans l'approche patron est toujours nécessaire pour choisir le patron et la solution de conception par rapport au problème à résoudre.

Nous présentons dans la partie suivante les travaux identifiés pour faciliter la réutilisation des solutions de conception proposées par les patrons.

2.2 Faciliter la réutilisation des solutions de conception

Rappelons que la réutilisation de solutions de conception proposées par des patrons consiste à imiter ce patron.

Dans [Borne, 99], l'état de l'art sur les outils disponibles pour faciliter l'imitation des patrons présente différents résultats académiques de génération de code à partir de la définition des patrons. Les auteurs proposent de les répartir selon les trois catégories suivantes : « primitives », « moulinette » et « environnement » (Figure 13).

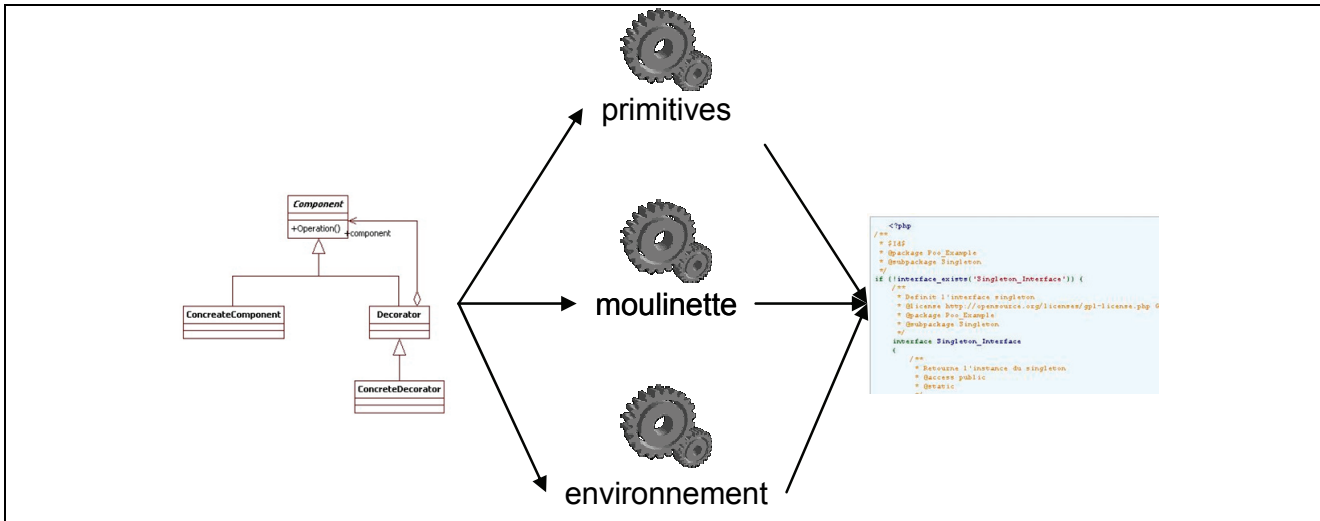


Figure 13 : génération de code à partir d'un traitement automatique sur l'application de patrons

Dans les outils de type « primitives », le modèle objet d'un langage est étendu par un ensemble de primitives spécifiques aux patrons et l'imitation du patron est reflétée dans le code par la primitive. Si cette solution réduit le fossé entre la modélisation et le programme, elle nécessite cependant de développer des langages spécifiques et les compilateurs associés.

Dans les outils de type « moulinette », le code correspondant au patron est produit à partir d'un ensemble de valeurs données en entrée. Dans ce cas, la génération du code prend en compte de manière automatique le contexte de son utilisation. Cependant, les défauts sont, d'une part la perte du lien entre le patron et son implémentation, d'autre part le risque de dissolution du code généré lors de son intégration par l'utilisateur dans le code de l'application finale.

Dans les outils de type « environnement », un système de représentation et d'instanciation de patron est intégré dans un environnement de génie logiciel. Ce type d'outil résout les problèmes liés aux deux types précédents, à savoir l'absence de définition de langage spécifique, la traçabilité de l'instanciation de patrons et la meilleure intégration du code généré dans le code de l'application.

En s'appuyant sur cette même classification, les auteurs de [Conte, 01b] analysent deux outils commerciaux de génération de code de type « environnement » qui facilitent la réutilisation de patrons de conception :

- *Framework Studio* : il réalise une copie du diagramme de classe pour que l'utilisateur l'adapte à son cas de conception, mais il ne maintient pas le lien entre la copie et la source ;
- *Objecteering* : il demande à l'utilisateur de choisir les classes sur lesquelles il doit appliquer le diagramme de classe d'un patron, mais il ne permet pas l'extension de son catalogue de patrons.

Ces deux outils, représentatifs, n'étant pas satisfaisants, [Conte, 01b] propose AGAP afin de faciliter l'imitation de patrons. Cependant et comme nous l'avons dit par rapport à la recherche, une expertise dans l'approche patron est toujours nécessaire pour imiter un patron à l'aide d'AGAP.

Si les deux premiers types d'outils offrent effectivement la possibilité de générer automatiquement du code à partir d'un patron, le peu de publications récentes semble indiquer que cette voie est trop complexe pour être mise en œuvre. Par contre, le troisième type est prometteur car AGAP est une proposition qui a été validée sur des projets industriels et qui a montré son intérêt, même s'il nécessite une expertise dans l'approche patron pour être utilisé de manière efficace.

3 Fonctionnalité « Identifier et faciliter le choix de composants logiciels »

Nous analysons dans cette partie les propositions pour identifier et faciliter le choix de composants logiciels [Cocquebert, 08b].

3.1 Identifier des composants logiciels

Depuis de nombreuses années, les composants sont modélisés par des facettes. Le principe de leur utilisation, formalisé et implémenté dans [Michel, 06], est de rassembler des informations de natures différentes (Figure 14) telles des informations fonctionnelles ou d'intégration (configurations logicielles, scénarios d'intégration, etc.) ou non fonctionnelles (retours d'expériences sur les incompatibilités, les facilités d'exploitation, etc.) et de permettre au concepteur de les consulter afin d'identifier un composant par rapport à son besoin.

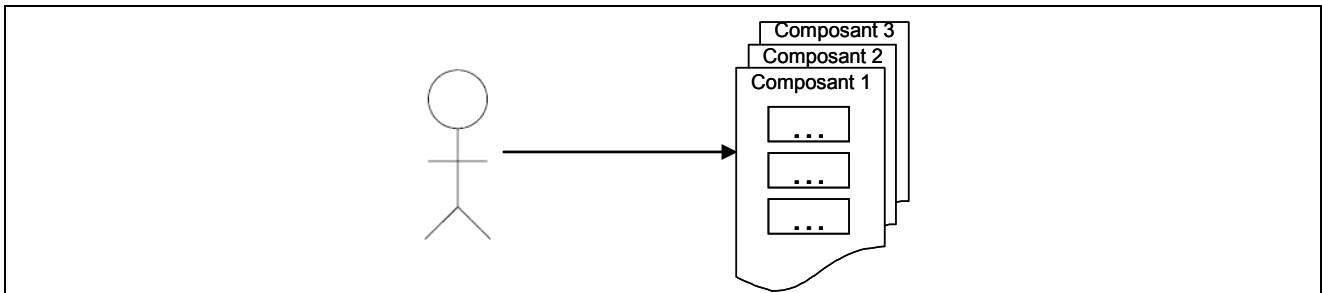


Figure 14 : caractérisation des composants

L'intérêt est de détecter dès la phase de choix des incompatibilités entre composants mais, comme l'ont noté entre autres [Clark, 04] et [Leung, 02], la nécessité de consulter des facettes pour identifier un composant peut conduire à consommer un temps important si le nombre de facettes et/ou de composants référencés est important. Pour résoudre ce problème de temps, nous présentons les solutions proposées par l'approche orientée modèle et l'approche orientée processus.

3.1.1 Approche orientée « modèle »

Dans cette approche, un processus et un modèle intermédiaire sont définis entre l'utilisateur et le modèle par facettes des composants. Dans les contributions analysées (Figure 15) :

- le modèle intermédiaire est un modèle de domaine¹² et les facettes du modèle de composant y sont associées ;
- leurs différences résident dans la mise en œuvre du processus qui identifie les composants.

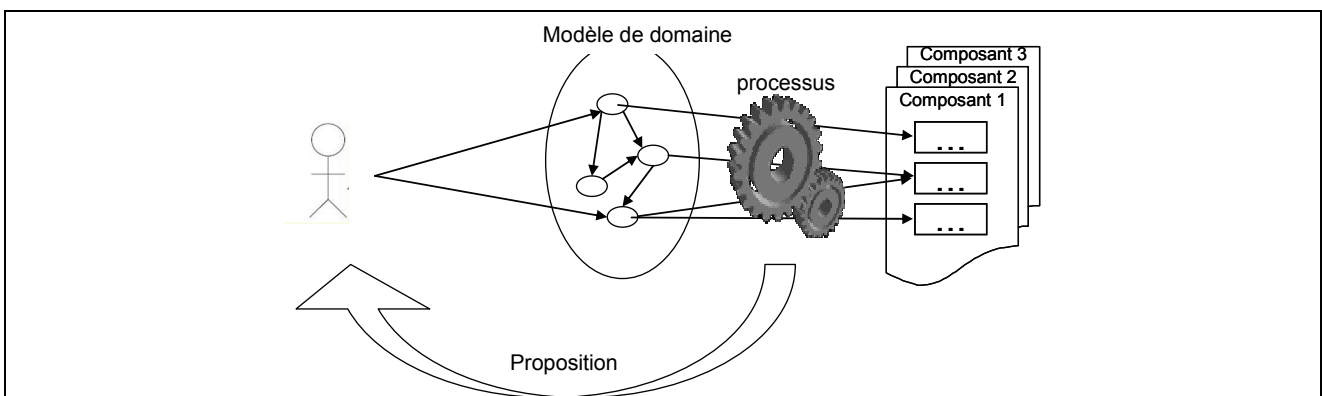


Figure 15 : approche orientée « modèle »

¹² « modèle de domaine » : ensemble d'objets, d'opérations et d'associations qui sont communs à une classe de système [Cauvet, 01]

Une première solution consiste à utiliser une des nombreuses méthodes analytiques ad hoc définies au début des années 2000 telles que STACE, OTSO, PORE, etc. (méthodes détaillées dans [Seibel, 05], [Le, 01] et [Leung, 02]). A partir de l'expression par l'utilisateur de son besoin dans le modèle de domaine, les méthodes identifient et minimisent le nombre de composants dont les facettes satisfont les modules fonctionnels sélectionnés (Figure 16). L'intérêt est d'appréhender la conception d'un point de vue fonctionnel et de laisser au processus le rôle de consulter les facettes pour mener à bien le processus de choix ; la limite de leur utilisation est leur définition ad-hoc (spécificités des traitements et des facettes utilisées, absence d'aide à la définition des facettes) et le temps de traitement dépendant du nombre de composants à examiner.

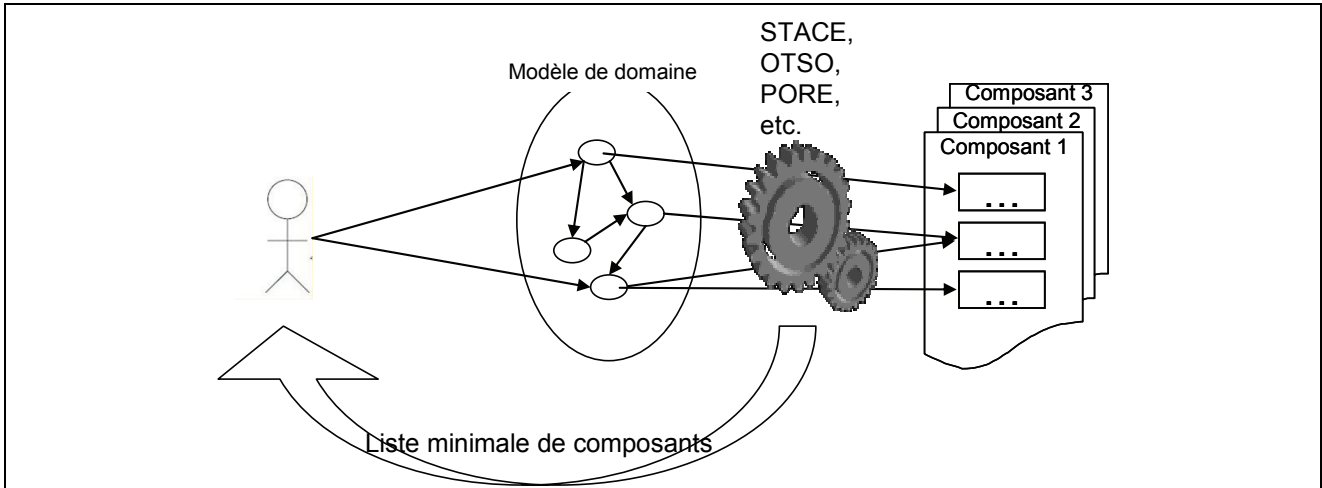


Figure 16 : utilisation d'un processus analytique

Pour résoudre ce problème de durée potentiellement important, [Leung, 02] définit et intercale un modèle supplémentaire entre l'utilisateur et le modèle de domaine (appelé « modèle de système basé sur les composants ») et associe les modules du système aux modules fonctionnels du modèle de domaine. Comme illustré dans la Figure 17, l'utilisateur choisit les modules fonctionnels dans le modèle du système basé sur les composants, ce qui identifie les modules correspondants dans le modèle de domaine. Un processus analytique minimise alors à partir du catalogue le nombre de composants satisfaisants les modules fonctionnels sélectionnés dans le modèle de système. Le niveau supplémentaire entre le modèle de domaine et l'utilisateur permet au processus de converger plus rapidement vers une liste minimale de composants, mais en constitue également la limite d'utilisation : les définitions et associations successives de modèles peuvent devenir fastidieuses, d'autant plus que les auteurs considèrent que les associations entre les composants et le modèle de domaine sont réalisées par les sociétés qui les diffusent (ce qui ne rentre pas dans leurs activités habituelles).

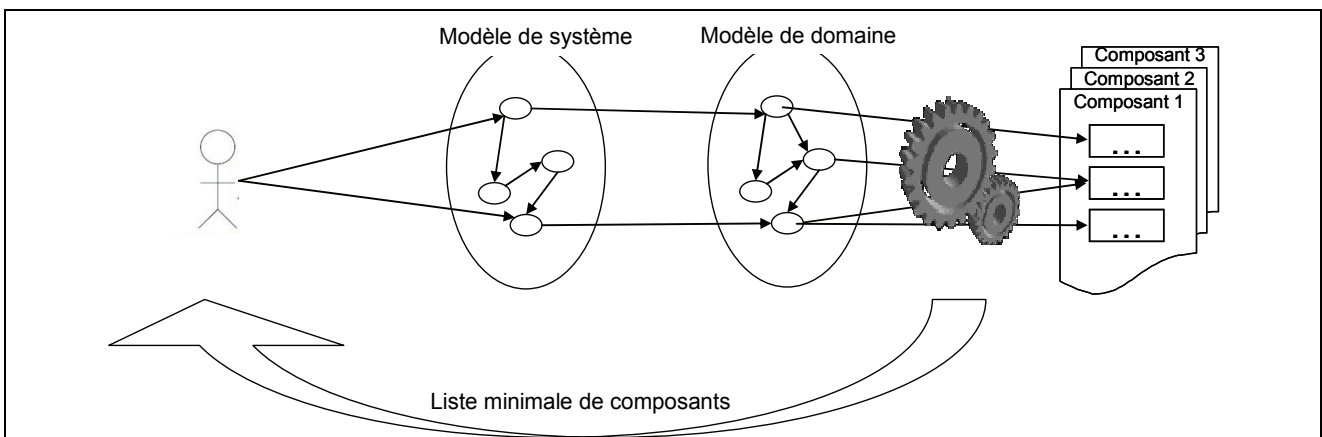


Figure 17 : méthode de sélection basée sur le modèle de domaine

Pour la conception de sites d'achat en ligne, [Pujalte, 04] propose un processus interactif de recherche de composants centré sur l'assistance à l'utilisateur et basé sur une ontologie du domaine¹³. Cette ontologie est composée d'un « modèle de domaine » et d'un « thésaurus » contenant les termes dérivés et les définitions des concepts définis dans le modèle de domaine (Figure 18). Quand l'utilisateur exprime sa requête au système, un processus utilise le thésaurus pour « reformuler » cette requête et la rendre plus précise, ce qui permet d'obtenir un ensemble restreint de composants candidats dès la première itération.

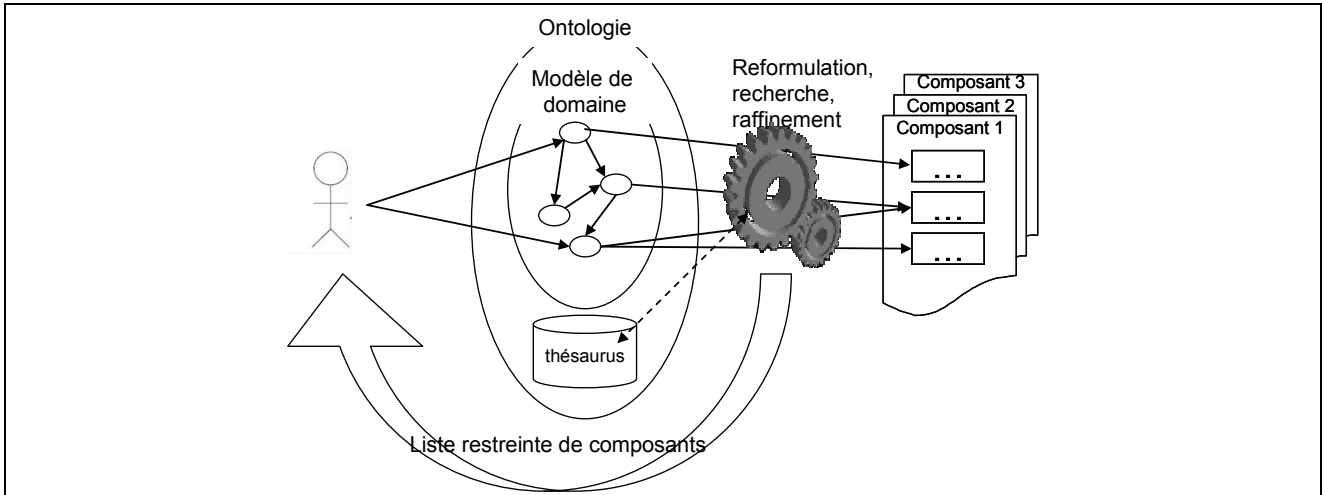


Figure 18 : utilisation d'une ontologie

L'intérêt de cette proposition est d'éviter à l'utilisateur de raffiner lui-même sa requête en lui fournissant immédiatement une liste restreinte de composants réellement appropriés à son besoin ; la limite de son utilisation est la nécessité de disposer de l'ontologie du domaine (ou de la réaliser) pour pouvoir utiliser le système.

La partie suivante analyse les travaux de la deuxième approche pour optimiser la recherche de composants dans un catalogue.

3.1.2 Approche orientée « processus »

Cette approche consiste à identifier les composants logiciels par une « reconnaissance » de leur modèle dans le modèle des spécifications.

Pour la conception d'un site web de vente en ligne, [Le, 01] propose un processus guidé par les besoins pour identifier des composants à partir de « cartes du besoin ». Ces cartes, décrivant les fonctionnalités utilisateur de l'application d'une part, le comportement fonctionnel des composants d'autre part, sont construites sous forme de graphes orientés dont les nœuds représentent les fonctions, et les arcs l'ordre d'exécution de ces fonctions (Figure 19). L'utilisateur décrit les besoins de l'application sous la forme d'une carte et un processus « reconnaît » les cartes des composants dans la carte de l'utilisateur. Ce processus tient compte qu'un composant ne satisfait généralement pas exactement aux besoins attendus et donc que la carte d'un composant ne se retrouve pas exactement dans la carte du système.

¹³ Ontologie : « spécification formelle et explicite d'une conceptualisation partagée » [Studer, 98]

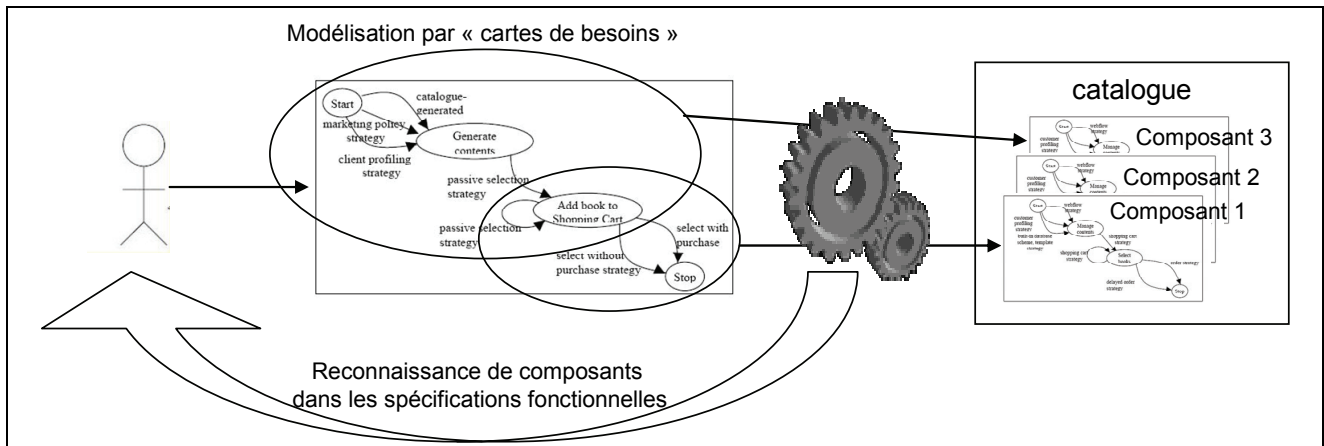


Figure 19 : reconnaissance de cartes de besoins (d'après [LE 01])

L'intérêt de cette approche est le processus semi-automatique pour l'identification des composants et l'expression du besoin en termes de fonctions ; les limites de son utilisation sont la nécessité de disposer des cartes des composants du catalogue et de réitérer le processus d'expression du besoin à chaque nouvelle application.

[Khayati, 05] propose pour la recherche de composants dans un catalogue un processus qui réalise un appariement structurel de diagramme de classe à partir de leurs transformations en prédicats selon une logique du premier ordre (Figure 20). Appliqué dans un premier temps à la reconnaissance d'un diagramme de classe de référence dans des diagrammes de classes spécifiques de composants, l'objectif de l'auteur est d'intégrer ces travaux dans un éditeur UML afin de reconnaître des composants au cours de la modélisation du système. Les intérêts de cette approche sont d'exploiter un formalisme standard pour identifier des composants et l'utilisation d'un processus automatisé pour les transformer ; la limite de son utilisation est la nécessité de disposer du diagramme de classe des composants.

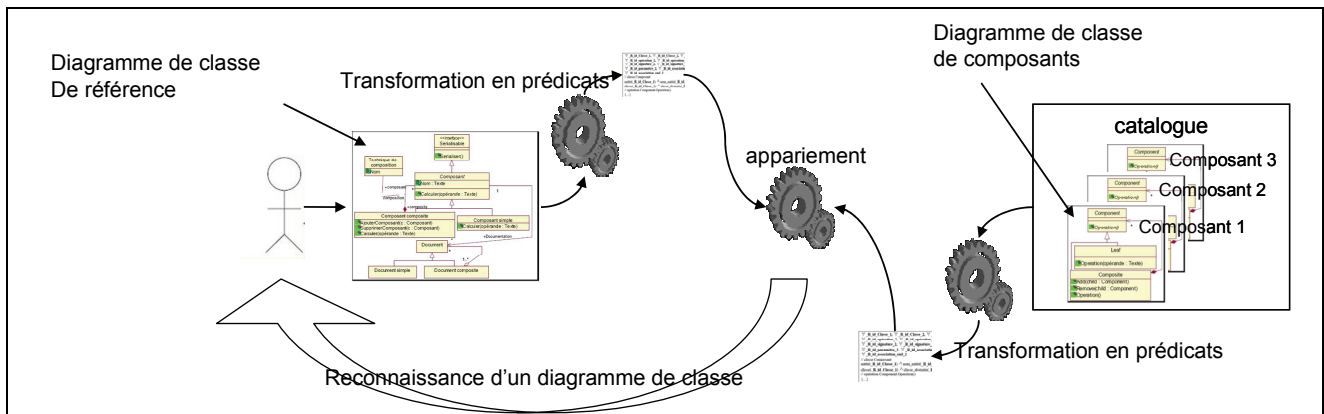


Figure 20 : appariement structurel (d'après [Khayati, 05])

Ainsi et compte tenu du contexte de notre étude, même si l'approche orientée modèle nécessite de disposer d'un modèle de domaine et d'y associer les composants, elle est préférable à l'approche orientée processus, car cette dernière nécessite de disposer du modèle utilisé par le processus, ce qui est rarement le cas pour les composants « open source » qui nous concernent.

Après avoir vu comment optimiser la recherche de composants dans un catalogue, la partie suivante analyse les travaux pour faciliter le choix d'un composant.

3.2 Faciliter le choix d'un composant

La première solution pour faciliter le choix d'un composant est constituée par le modèle de composant lui-même qui, comme nous l'avons présenté dans la partie précédente, rassemble des informations de natures différentes qui sont consultées par l'utilisateur. Cependant, cette solution ne peut être utilisée que sur un nombre restreint de composants.

La deuxième solution est d'optimiser les résultats des traitements des méthodes ad-hoc (présentées également dans la partie précédente), mais nous avons noté que ces processus ont l'inconvénient d'être trop spécifiques au domaine d'application.

Une troisième solution est de catégoriser les composants et de les organiser selon différentes catégories afin de restreindre le champ de la recherche pour un besoin donné et diminuer ainsi le temps de recherche. Dans ce cas, la navigation dans le catalogue se fait par des liens hypertextes organisés sous la forme d'une arborescence (Figure 21).

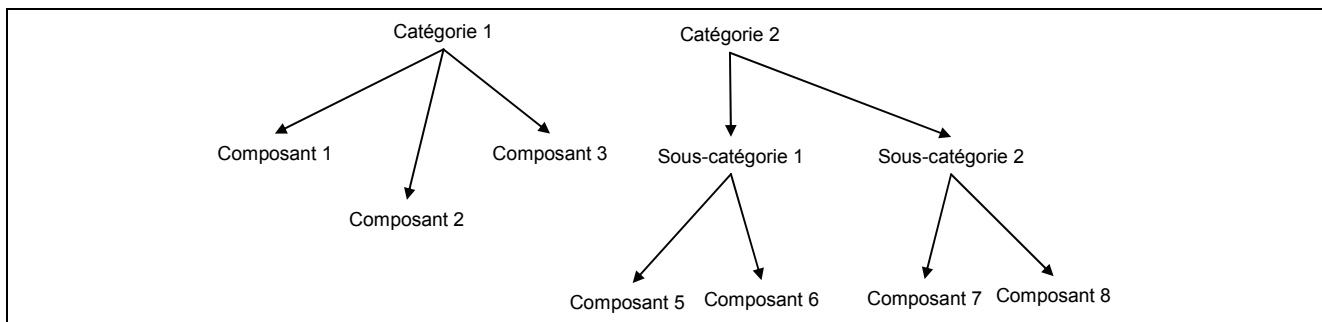


Figure 21 : catégorisation des composants

C'est la méthode retenue par la plupart des catalogues disponibles sur internet, parmi lesquels nous pouvons citer « CMSMatrix¹⁴ » pour les CMS (« Content Management System »), « opensource¹⁵ » pour des composants métiers, « PEAR¹⁶ » et « PECL¹⁷ » pour des composants techniques.

L'intérêt de cette approche est l'identification d'un composant à partir de sa catégorie ; sa limite est la nécessité de trouver un équilibre entre la finesse de la catégorisation des composants et la limitation du nombre de catégories afin d'orienter efficacement la navigation dans l'arborescence sans rendre la navigation trop fastidieuse. Quand le nombre de composants est important, la catégorisation est complétée par une recherche par mots-clés, mais ceux-ci doivent avoir été associés aux composants au préalable et ils doivent être connus de l'utilisateur pour que le choix soit réellement facilité.

Ainsi, la solution pour faciliter le choix d'un composant parmi un ensemble dépend du nombre de composants de cet ensemble et du nombre de catégories différentes de ces composants.

4 Fonctionnalité « Accélérer la mise en place d'un site web »

Nous analysons dans cette partie les propositions pour la génération de données à partir des choix de conception et pour accélérer la conception d'un site web [Cocquebert, 08b].

4.1 Génération de données

En utilisant l'approche patron pour faciliter le développement de systèmes coopératifs utilisant des sources de données hétérogènes et distribuées, [Couturier, 04] propose un ensemble de patrons (basés sur les patrons

¹⁴ <http://www.cmsmatrix.org/>

¹⁵ <http://www.opensourcecms.com/>

¹⁶ <http://pear.php.net/>

¹⁷ <http://pecl.php.net/>

de domaine de Fowler) qui génèrent à partir de la modélisation du besoin la solution d'implémentation sous la forme de composants. L'auteur propose (Figure 22) :

- des « patrons de coopération » qui aident à la conception du système indépendamment de la plateforme ;
- des « patrons supports de coopération », eux-mêmes séparés en deux catégories :
 - les « patrons supports d'utilisation » : ils guident l'utilisation et l'assemblage des précédents en exploitant les relations de transformations et d'intégration entre les modèles,
 - les « patrons supports techniques » : ils génèrent les composants pour l'implémentation de la solution en fonction de la plateforme en utilisant un générateur de code .

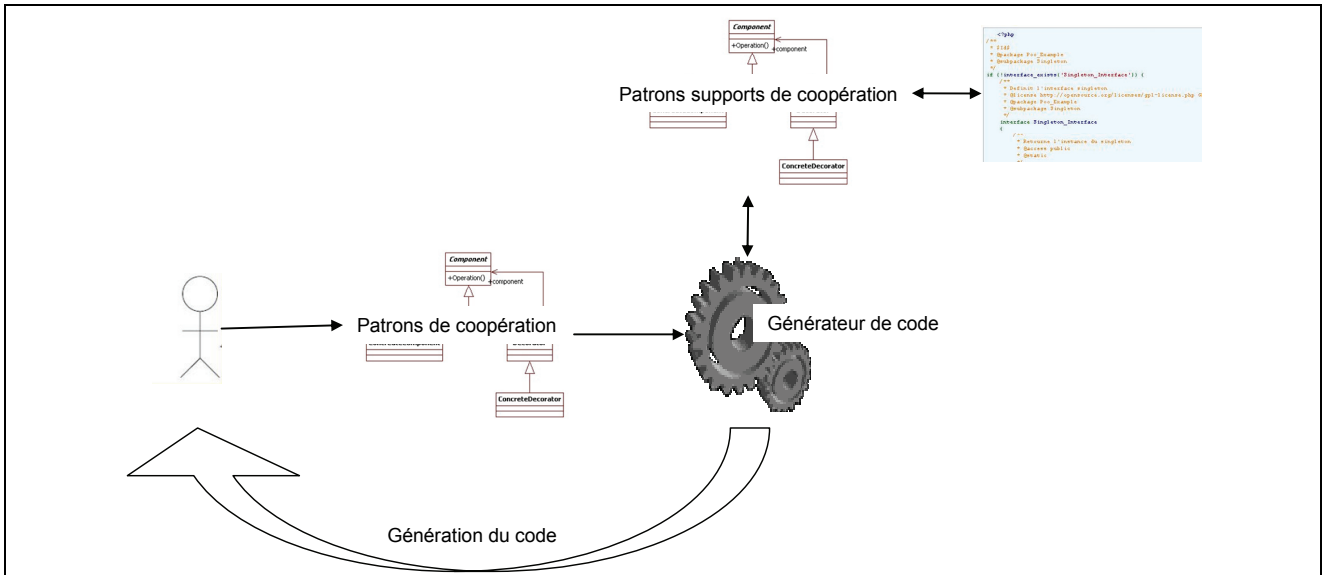


Figure 22: génération de code à partir de patrons (d'après [Couturier, 04])

En utilisant l'IDM pour construire la cartographie d'une application logicielle basée sur des technologies hétérogènes, [Favre, 06b] propose un processus de "rétro-ingénierie dirigée par les métamodèles". Ce processus génère les modèles pour des composants dont il explore le code et intègre ensuite ces modèles au sein d'un modèle global (Figure 23). L'intérêt de cette approche est la prise en compte de développements existants (c'est la seule parmi toutes les approches que nous avons identifiées) dans une démarche guidée par l'IDM ; la limite de cette approche est la difficulté de résoudre les problèmes posés par l'intégration et la composition de différents modèles générés.

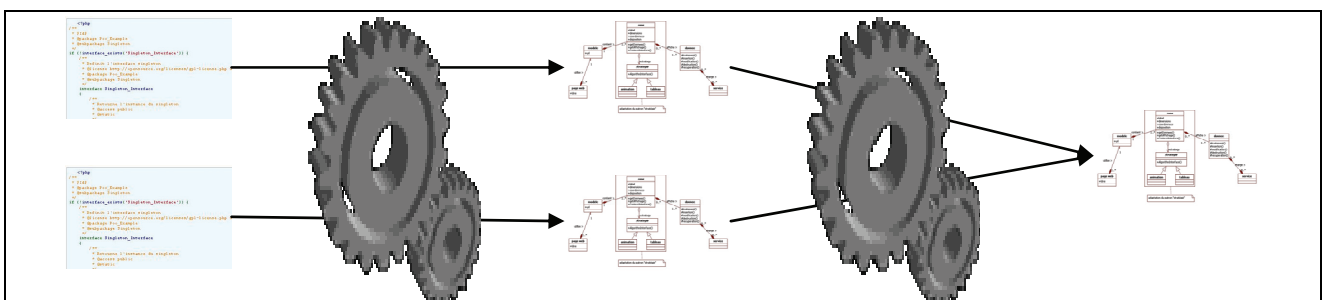


Figure 23 : rétro-ingénierie dirigée par les méta-modèles

Dans les outils associés aux méthodes d'aide au développement de sites web (VisualWADE⁶, ArgoUWE⁷, Webratio⁸, etc.), ou d'éditeurs de logiciels (Zend Framework⁹, PHPEclipse¹⁰, etc.), l'assistance se traduit par la génération des squelettes des classes correspondantes et par la mise à disposition de composants pour faciliter l'implémentation de ces classes. Cependant, l'analyse que nous avons menée a montré que la génération de code est uniquement descendante (une modification du code n'est pas répercutée dans la modélisation).

Les gadgets (ou « widgets ») constituent un exemple de mise en œuvre du concept « web 2.0 » où le contenu d'une page web peut être défini sur le poste client de manière interactive dans le navigateur internet. Sur les

sites web de iGoogle¹⁸ (Figure 24) et de Netvibes¹⁹, l'utilisateur a la possibilité de créer sa page web en incluant des gadgets prédéfinis qui proposent des services en ligne tels que :

- flux de données externes (météo, Le Monde.fr, Le Figaro.fr, etc.) ;
- services évolués (google maps, gmail, notes, géo-localisation, etc.) ;
- jeux (solitaire, échec, pac-man, etc.) ;
- utilitaires (calendrier, heures multi-fuseau, dictionnaire, etc.) ;
- etc.

Sur le site Webwag²⁰, l'utilisateur peut inclure ses propres gadgets, générés à partir de la capture d'une portion de page web spécifique que l'utilisateur souhaite suivre.

Ces descriptions montrent que dans ce cas des gadgets, l'assistance à la génération de code est à son plus haut niveau car les pages web sont générées de manière totalement interactive. Cependant, ces mises en œuvre ne permettent pas de générer des applications web à part entière, notamment en connectant des pages web à des bases de données.



Figure 24 : recherche et configuration de gadgets sur iGoogle

Ainsi, ces propositions montrent la mise en œuvre de la génération automatique de données à partir de décisions de conception en appliquant un traitement sur des données initiales. Cependant, elles ne prennent pas en compte l'existence de développements existants car elles nécessitent des traitements difficiles à mettre en œuvre. C'est probablement la raison pour laquelle cette prise en compte semble rarement envisagée au vu des travaux que nous avons identifiés.

Après avoir analysé les travaux pour la génération de données à partir de choix de conception, la partie suivante analyse les travaux pour accélérer la conception d'un site web.

¹⁸ <http://www.google.fr/ig>

¹⁹ <http://www.netvibes.com/>

²⁰ <http://www.webwag.com/>

4.2 Accélérer la conception d'un site web

En utilisant l'approche patron, D. Parson propose un framework d'accueil représenté Figure 25. Comme le montre cette figure, il est composé [Parson, 05] :

- d'un noyau (« Framework backbone ») qui met en œuvre les fonctionnalités de bas niveau du framework ;
- de composants de base (« BC ») qui, déjà intégrés au noyau, offrent de manière native des fonctions de haut niveau ;
- d'un ensemble de métadonnées (« metadatas ») afin de faciliter l'intégration de composants externes (« AC »), permettant ainsi d'étendre facilement les fonctionnalités offertes par le framework.

Cette architecture donne donc au concepteur une implémentation de départ prête à être utilisée telle que, ou prête à être étendue par l'intégration de nouveaux composants.

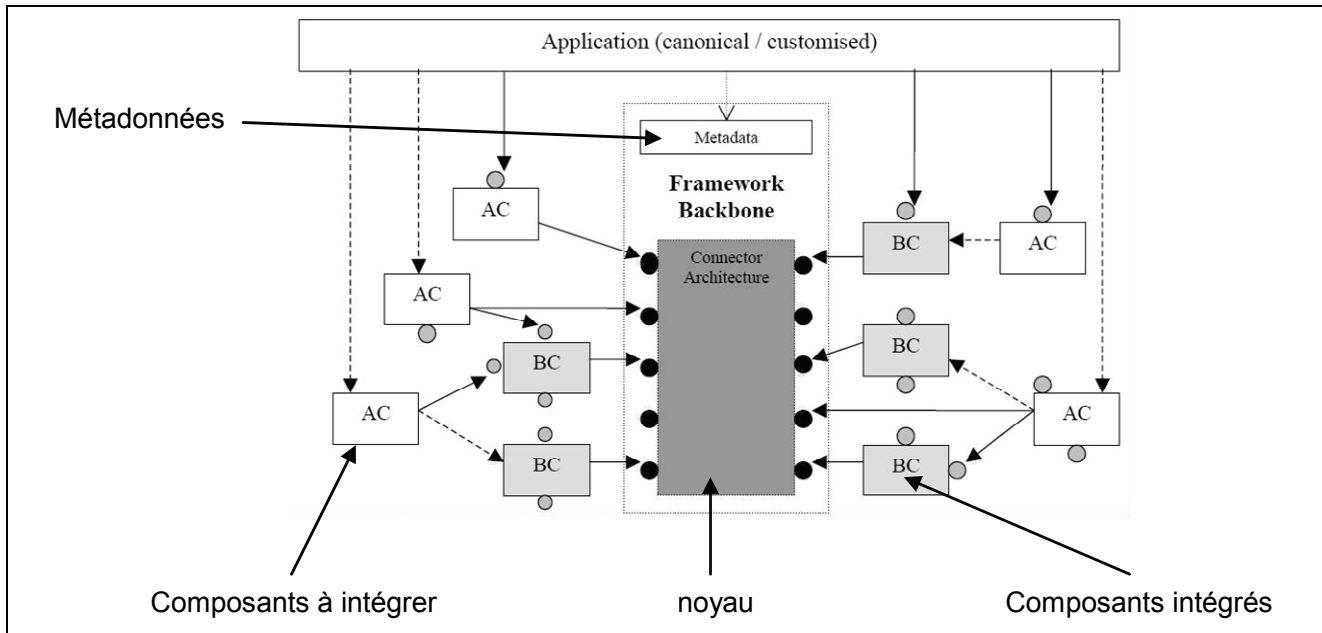


Figure 25 : architecture logicielle d'accueil ([Parson, 05])

Le système patron de J. Licea est complété par la définition explicite d'un framework de classes [Licea, 00]. Ce framework implémente les solutions de conception proposées par les patrons du système, additionnées par des classes spécifiques dont le but est de faciliter l'implémentation d'une application collaborative (Figure 26).

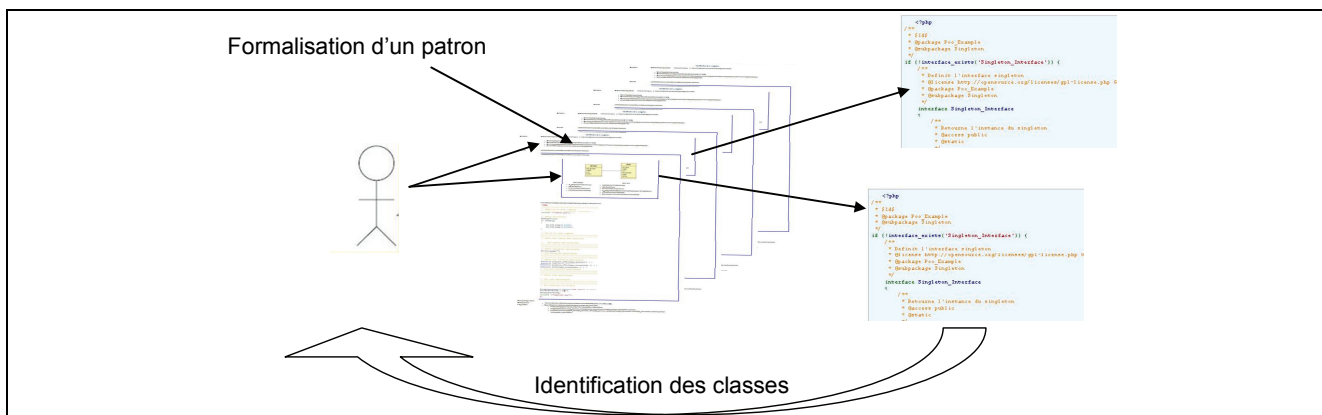


Figure 26 : proposition d'implémentation à partir de patrons

Dans un tel système, l'assistance à la génération de code se traduit par une prise en charge implicite de fonctions de bas niveau telles que l'envoi et la réception de messages entre les modules fonctionnels de

l'application. Cela permet aux développeurs de se concentrer sur les fonctionnalités de l'application, et de fournir au final une application présentant plus de fonctionnalités que s'ils n'avaient pas utilisé le framework de classes. Cependant, il n'en reste pas moins qu'il subsiste toujours une part manuelle du codage de l'application finale.

Dans une optique d'IDM, [Muller, 05] propose un langage action qui génère automatiquement un modèle indépendant de la plateforme à partir d'une modélisation du besoin, puis un processus de compilation qui génère le code en fonction de la plateforme cible (Figure 27). L'intérêt est de modéliser le site web à partir de sa modélisation (même si c'est « from scratch »), et de générer ensuite le code en fonction de la plateforme ; sa limite est de ne pas considérer l'existence de développements logiciels existants.

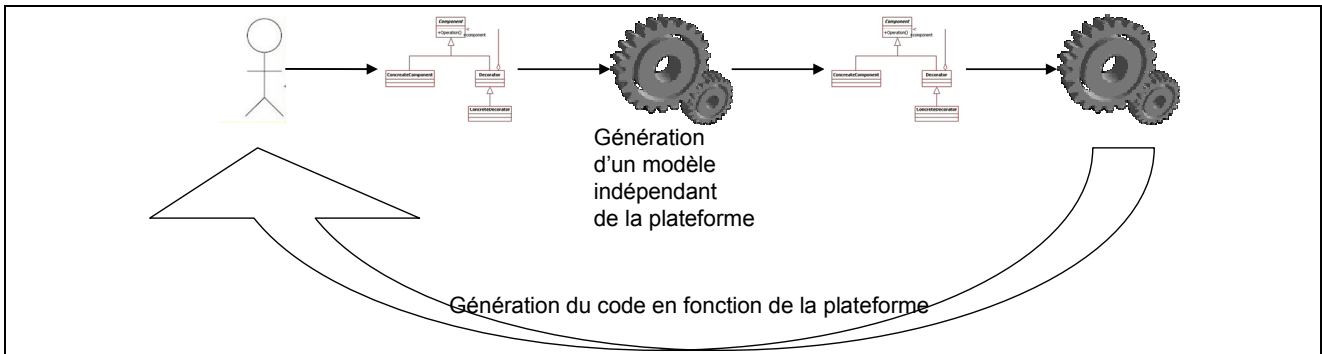


Figure 27 : génération automatique du code

Les frameworks proposés par l'industrie du logiciel pour l'ingénierie du web (Symfony²¹, Copix²², etc.) permettent la conception d'une application à part entière, mais l'assistance est minimale. Elle se traduit par les deux manières d'utiliser les frameworks : spécialisation via le mécanisme d'héritage ou par leurs interfaces s'ils sont définies, ou intégration dans un autre système après configuration et/ou adaptation. Mais même si le framework fournit le moyen de se concentrer sur le développement des aspects métiers plutôt que sur les fonctions de bas niveau, le développement de l'application est entièrement à la charge du concepteur (Figure 28). De plus :

- Il est nécessaire de respecter l'architecture du framework lors de son utilisation, sous peine de ne pas en bénéficier pleinement ;
- Il est difficile d'y intégrer un développement déjà existant s'il ne respecte pas les principes de développement imposés par le framework.

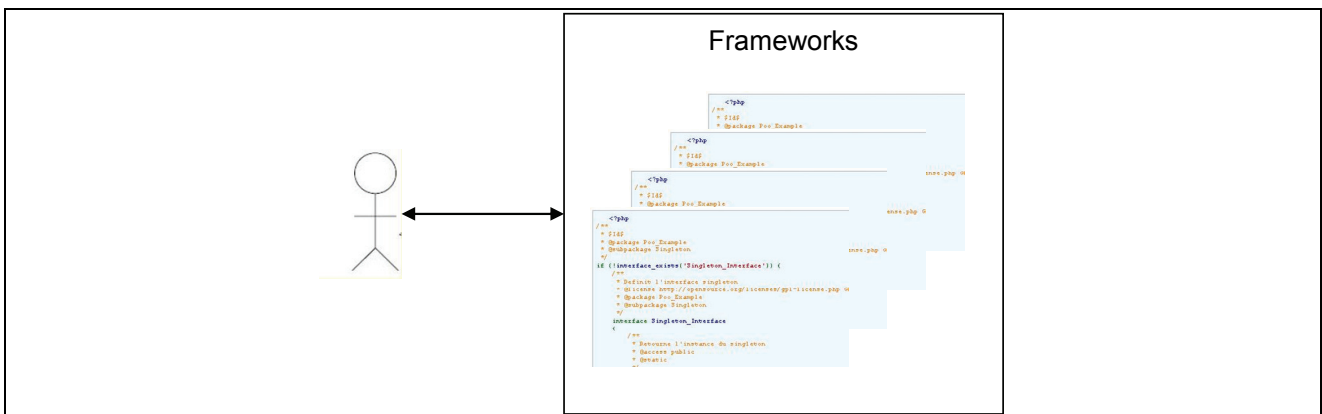


Figure 28 : utilisation d'un framework ou d'un CMS

²¹ <http://www.symfony-project.org/>

²² <http://www.copix.org/index.php/wiki/copix>

Les CMS (Content Management System) constituent un exemple de l'utilisation des frameworks pour développer des familles génériques d'applications. Ce sont des logiciels de gestion de contenu dédiés à la création de sites web communautaires. Le principe est de proposer sous la forme de composants des fonctions que l'on retrouve généralement dans les sites web portails (forums de discussion, enquêtes en ligne, profil d'accès, etc.), que l'on peut configurer ensuite selon le contexte d'utilisation (cf. Figure 29 la page de configuration de Joomla²³).

Parmi les nombreux CMS disponibles¹⁴, certains offrent la possibilité de développer, d'intégrer et de diffuser sous la forme de composants des développements spécifiques. Dans ce cas, l'assistance est semi-automatique car des composants spécifiques sont intégrés manuellement, mais, de manière analogue aux frameworks (cf. paragraphe précédent), l'architecture de développement des CMS doit être respectée pour en bénéficier pleinement.

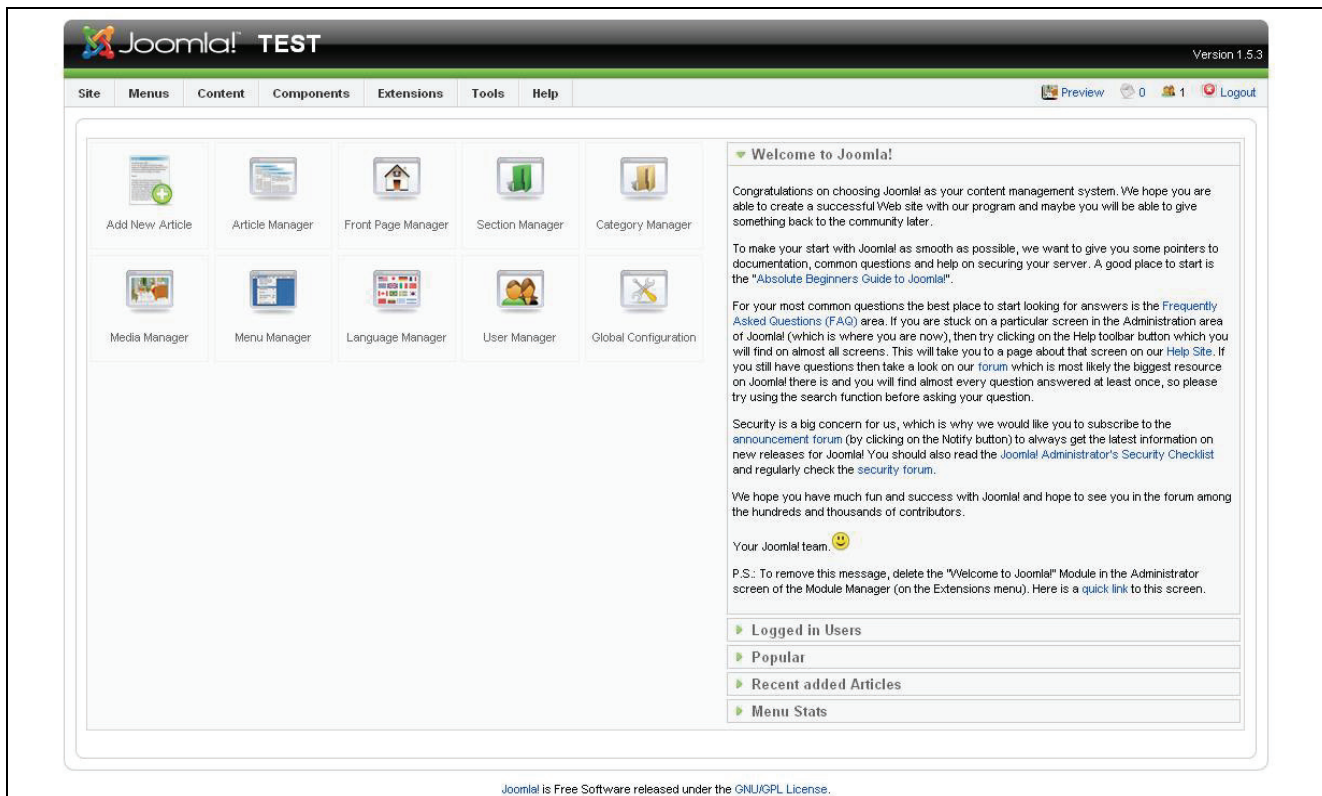


Figure 29 : page de configuration de Joomla

Ainsi, pour accélérer la mise en place d'un site web, il est préférable de définir une architecture logicielle d'accueil plutôt qu'une génération automatique de code à partir de modèles car elle permet l'intégration de composants logiciels existants et offre un framework de classes intégré pour implémenter en natif les fonctions de bas niveau, ce qui n'est pas le cas de la deuxième solution.

5 Conclusion

Nous avons analysé dans ce chapitre un état de l'art dans les différents domaines dont notre étude relève pour chacune des fonctionnalités que nous proposons de mettre en œuvre. C'est par rapport à ces dernières que nous synthétisons ci-dessous les résultats de cette analyse [Cocquebert, 08a].

Fonctionnalité « Assister la réalisation du processus de conception ». Pour guider un processus en général, il est préférable que ce guide soit informatisé pour que le suivi soit correct ; que les étapes soient adaptés à la réalité du processus afin qu'il soit utilisé. Concernant les sites web, les méthodes existantes considèrent

²³ <http://www.joomla.org/>

toutes la modélisation d'un site web selon trois dimensions « présentation », « navigation » et « donnée », mais certaines intègre l'implémentation. Pour aider à la réalisation d'un processus, l'approche patron a montré sa capacité à apporter cette aide au concepteur, mais une expertise dans l'approche patron est nécessaire pour l'exploiter efficacement.

Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception ». Pour identifier des solutions de conception, l'utilisation de P-Sigma à travers AGAP a montré son intérêt pour exploiter au mieux le formalisme patron, même si une expertise dans l'approche patron est toujours nécessaire pour identifier un patron dans un catalogue et pour identifier une solution de conception. Pour faciliter la réutilisation des solutions de conception, la solution la plus prometteuse est d'intégrer des mécanismes de représentation et d'instanciation de patrons dans un environnement de génie logiciel, afin d'avoir le moyen de garder la trace de l'imitation d'un patron dans le code de l'application finale.

Fonctionnalité « Identifier et faciliter le choix de composants logiciels ». Pour identifier des composants, l'approche orientée modèle est préférable à l'approche orienté processus car, même si la première nécessite d'associer un modèle de composant à un modèle de domaine, la seconde utilise un processus de reconnaissance qui nécessite de disposer de modèles spécifiques rarement disponibles dans le cas des composants « open source ». Pour faciliter le choix de composants, les solutions identifiées dépendent du nombre de composants et de catégories.

Fonctionnalité « Accélérer la mise en place d'un site web ». Pour générer une structure de données correspondant à la réutilisation d'une solution de conception, les exemples de génération de données ont en commun de ne pas prendre en compte des composants existants et de définir les traitements en fonction de la source et de l'objectif (des données vers les modèles, ou des modèles vers les données). Pour accélérer la conception d'un site web, dans notre objectif de faciliter la réutilisation de composants logiciels, il est préférable de définir une architecture d'accueil ouverte qui intègre un framework de classes et des comportements de bas niveau plutôt qu'une génération automatique de code à partir de modélisations.

L'objectif du chapitre suivant est de présenter la démarche à suivre pour définir une méthode d'aide à la conception en nous appuyant sur cette analyse.

Chapitre III : Démarche

En nous basant sur l'analyse de l'état de l'art, l'objectif de ce chapitre est de présenter la démarche à suivre pour définir une méthode d'aide à la conception de site web qui facilite la réutilisation de choix de conception et de composants logiciels. Egalement structuré par rapport aux différentes fonctionnalités, ce chapitre détaille les différentes étapes à réaliser pour les mettre en œuvre, et décrit le processus de leur enchaînement sous la forme de diagrammes SADT.

1 Fonctionnalité « Assister la réalisation du processus de conception »

La mise en œuvre de cette fonctionnalité nécessite de définir comment guider le concepteur dans le processus de conception de site web et comment l'aider à réaliser les étapes de ce processus.

1.1 Guider le concepteur dans le processus

Pour guider efficacement l'utilisateur, il faut informatiser le processus et définir les étapes à partir du processus réel. Pour cela, il faut définir les étapes du processus, le processus lui-même et le modèle de données utilisé (Figure 30).

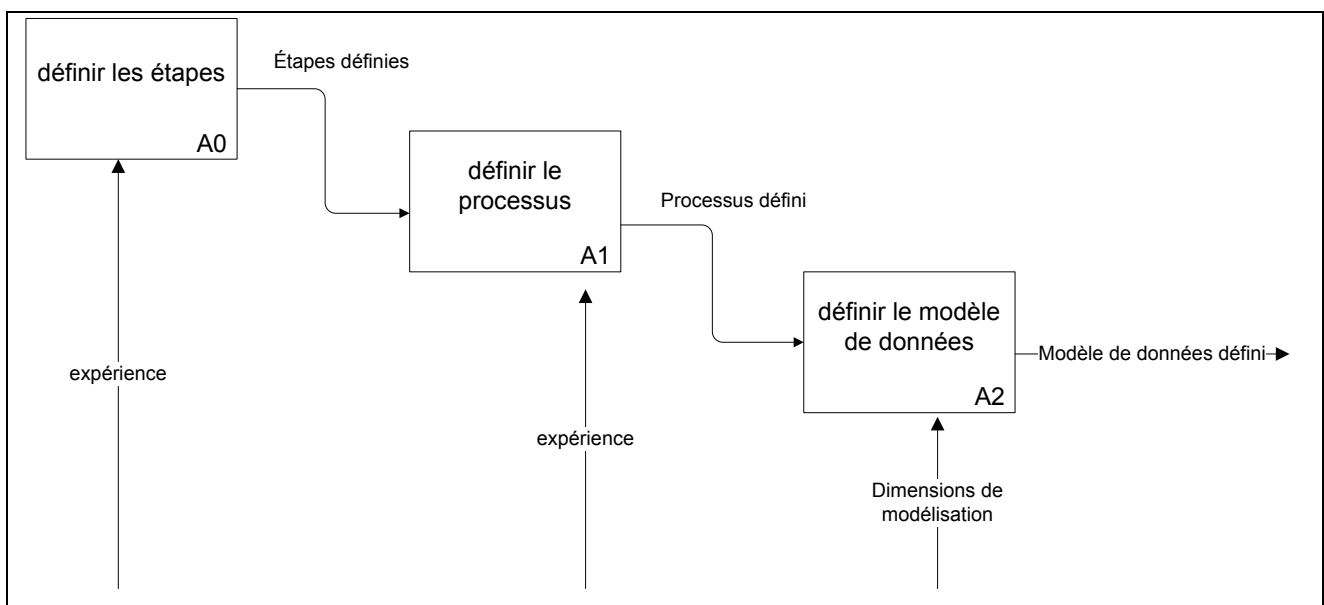


Figure 30 : définir le processus de conception

1.1.1 Définir les étapes

Les étapes du processus doivent permettre de définir les données relatives aux dimensions de modélisations unanimement retenues par les méthodes de conception des sites web :

- la navigation pour accéder aux différentes pages des sites web ;
- la présentation des données dans ces pages ;
- les données contenues dans les pages.

Cependant, il faut également reprendre une dimension qui n'est considéré que par quelques méthodes : l'implémentation au sein d'une architecture logicielle de l'affichage des données. En ajoutant cette dimension, le choix d'un composant pour implémenter un affichage ou un service fait partie intégrante du processus.

1.1.2 Définir le processus

Les méthodes de conception ayant toutes été définies pour des besoins spécifiques, il n'y a pas de consensus sur un processus de conception de sites web comme il en existe pour la conception des applications logicielles classiques. En conséquence, il faut définir le processus à partir de notre propre expérience de conception et proposer la succession des étapes pour définir les données relatives aux dimensions présentées ci-dessus :

- « navigation » pour accéder aux pages ;

- « présentation » pour définir la manière selon laquelle les données sont affichées dans les pages ;
- « données » pour définir le contenu des pages ;
- « implémentation » pour définir la manière selon laquelle l’affichage des données est implémenté.

1.1.3 Définir le modèle de données

Le modèle de données doit définir les informations relatives à chacune des dimensions de modélisation considérées par les étapes du processus de conception. (Figure 31).

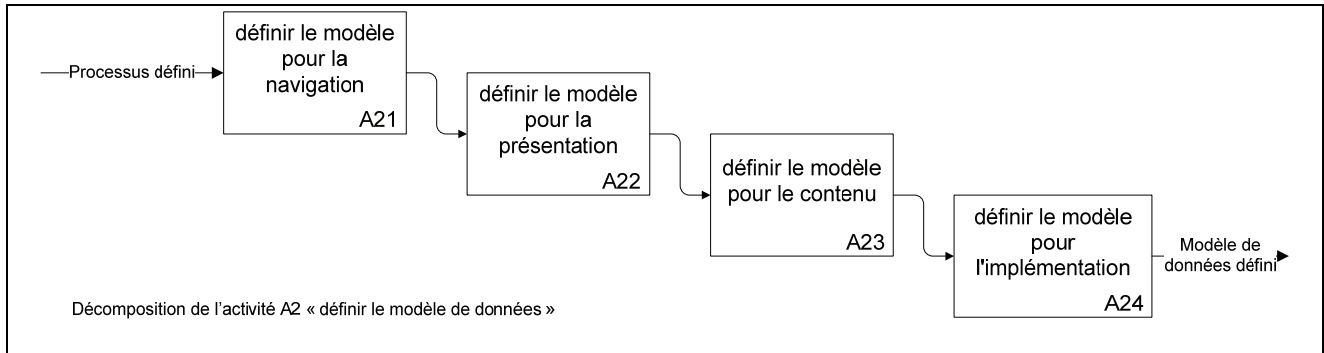


Figure 31 : définir le modèle de données

1.1.3.1 Définir le modèle pour la navigation

La première dimension de modélisation unanimement reconnue par l’état de l’art considère la navigation, c’est-à-dire le menu pour accéder aux différents contenus d’un site web.

Un menu est constitué d’un ensemble d’items, éventuellement répartis en groupes si le nombre d’items est important. Chacun de ces items peut accéder à un fichier pour afficher des données statiques ou dynamiques, ou charger un service interne ou externe au site web.

Si on considère qu’un menu peut se modéliser sous la forme d’une arborescence dont les nœuds sont les groupes d’items et les feuilles les items, le modèle pour la navigation doit permettre de définir cette arborescence de manière non limitée en profondeur (Figure 32).

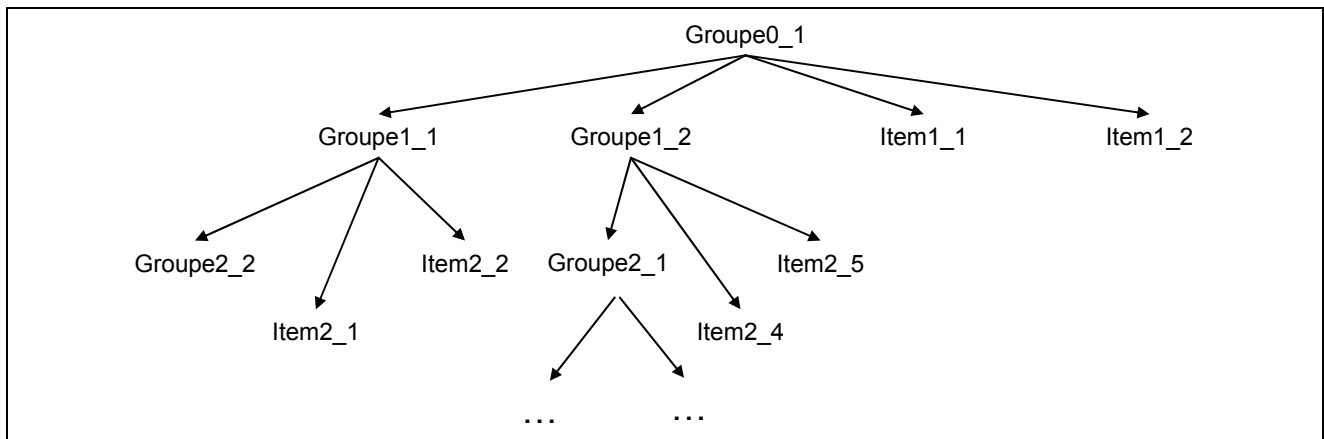


Figure 32 : représentation du menu d’un site web sous la forme d’une arborescence

Pour permettre de définir différents affichages pour un même menu (par exemple : verticalement dans le bandeau gauche, ou horizontalement sous la forme d’onglets dans la partie supérieure du contenu de la page), l’affichage de ce menu doit être modélisé de manière indépendante.

1.1.3.2 Définir le modèle pour la présentation

La seconde dimension de modélisation unanimement reconnue par l'état de l'art considère la présentation, c'est-à-dire l'organisation et la charte graphique²⁴ selon laquelle les contenus sont présentés au sein de chaque page d'un site web. L'aspect graphique n'étant pas pris en compte dans le cadre de notre étude, nous nous focalisons sur l'organisation au sein de la page, définie dans le modèle de présentation des pages.

En nous basant sur les analyses de 50 sites web en production, réalisées pour alimenter un catalogue d'analyses de sites web proposé dans le chapitre suivant, nous avons constaté que les pages sont organisées selon différentes zones spécifiques. Comme l'illustre la Figure 33, ces zones peuvent afficher des informations :

- relatives à la navigation ;
- communes à toutes les pages ;
- spécifiques à la page d'accueil ;
- spécifiques aux pages internes.

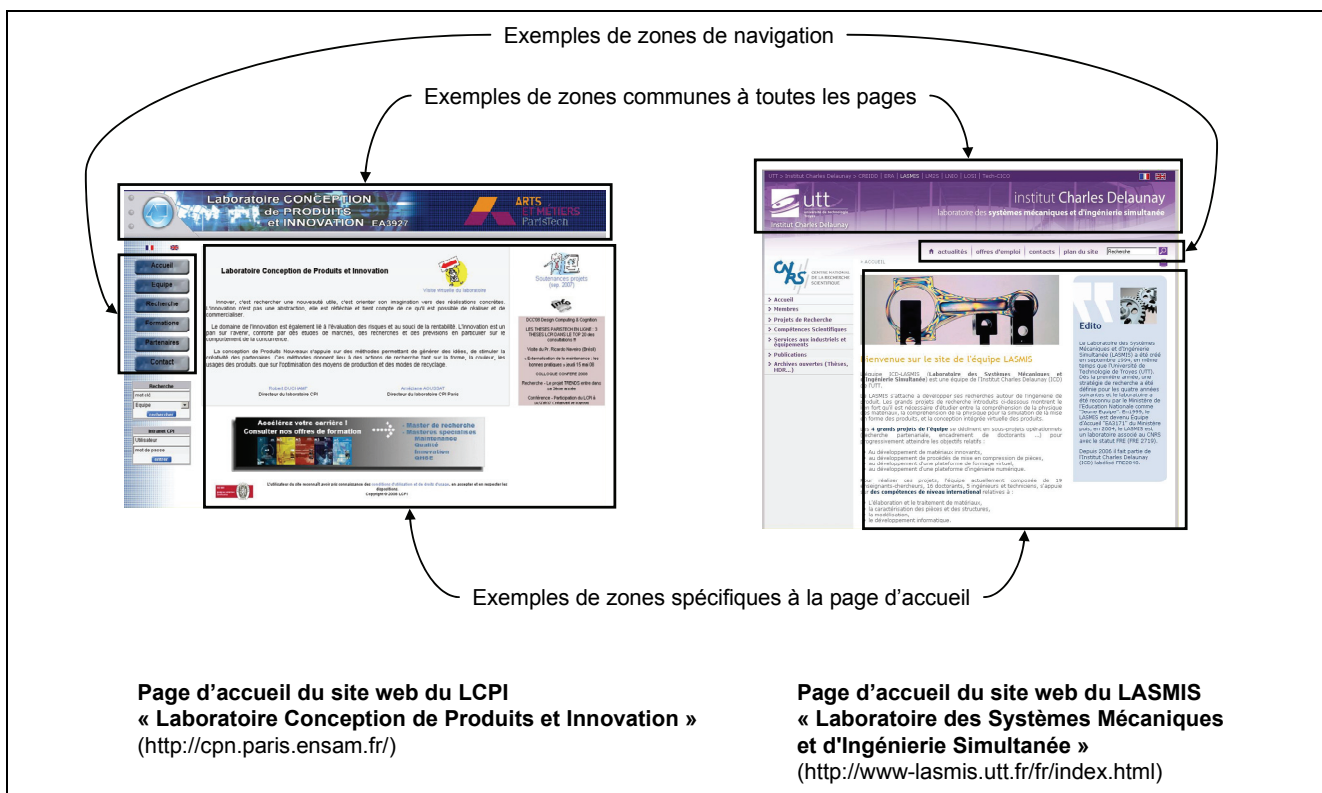


Figure 33 : exemples de types de zones dans deux sites web différents en production

Ainsi, le modèle pour la présentation doit :

- permettre la modélisation d'une page web comme un ensemble de zones d'affichage ;
- définir l'organisation relative de ces zones au sein d'un fichier modèle de présentation ;
- associer toutes les pages d'un site web à ce fichier modèle de présentation ;
- associer à chaque zone le contenu à afficher ;
- définir l'affichage lui-même de manière indépendante afin de, par exemple, changer facilement la charte graphique d'une page web.

²⁴ polices de caractères, couleurs, images, etc.

1.1.3.3 Définir le modèle pour le contenu

La troisième dimension de modélisation unanimement reconnue par l'état de l'art considère les données, c'est-à-dire le contenu affiché dans les pages d'un site web. Cependant, nous pensons que le terme « donnée » est trop restrictif car si le contenu associé à un item peut être un ensemble de données statiques ou dynamiques, il peut également se présenter sous la forme d'un service externe ou interne au site web. C'est pourquoi le terme utilisé dans la suite de ce mémoire pour cette troisième dimension est « contenu ».

Dans le cas de données statiques ou d'un service, le modèle pour le contenu doit permettre de définir l'adresse du fichier statique ou du service.

Dans le cas d'informations dynamiques, le modèle pour le contenu doit permettre de définir différents types de données afin de ne pas surcharger l'affichage et respecter des critères élémentaires d'ergonomie. Ces types sont :

- « donnée principale » qui permet de différencier de manière unique mais significative pour l'utilisateur les enregistrements de la base ;
- « donnée secondaire » afin de compléter la définition d'une donnée principale ;
- « donnée interne » pour compléter les deux types précédents et contrôler leur affichage (par exemple : date d'insertion pour définir l'affichage d'une image « nouveauté », date de modification pour définir l'affichage d'une image « modification »).

1.1.3.4 Définir le modèle pour l'implémentation

Nous reprenons comme quatrième dimension de modélisation une prise en compte explicite de l'implémentation dans le processus de conception comme cela a été proposé par les méthodes OOADM et WSDM (cf. Tableau 2 page 38). En effet, l'implémentation est une des étapes considérée par notre étude et elle doit être prise en compte par notre modèle de données.

Une implémentation peut être réalisée de différentes manières (Figure 34) :

- un composant métier ou un développement spécifique qui offre une fonction globale évoluée (ex : gestion d'un forum), ou qui utilise des données distantes (ex : affichage de la météo) ;
- un flux RSS qui permet la récupération de données distantes pour un affichage local (ex : « la une » de « lemonde.fr ») ;
- un composant technique ou un développement spécifique qui affiche des données dynamiques.

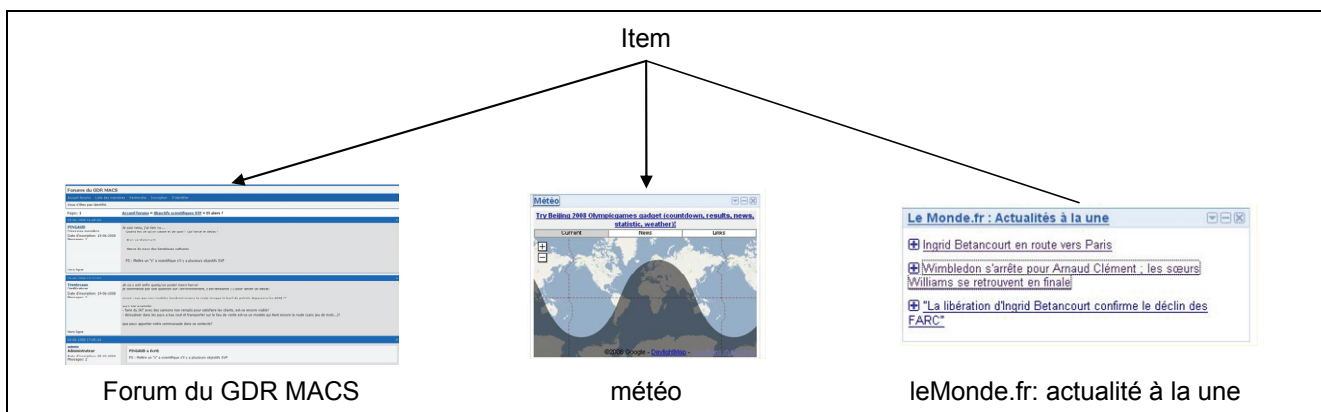


Figure 34 : types de services

Ainsi, le modèle de cette dimension doit caractériser les différents types d'implémentation et permettre de les associer directement à un item du menu, ou à un contenu associé à un item de menu.

Après avoir présenté nos spécifications pour guider le concepteur, la partie suivante aborde les spécifications pour aider à réaliser les étapes du processus.

1.2 Aider le concepteur à réaliser les étapes du processus

Pour apporter une aide à la réalisation des étapes du processus, il faut définir des patrons de domaine pour les étapes du processus. Afin de bénéficier de la réflexion qui a été faite au LIG pour l'informatisation de la recherche de patrons dans le cadre du projet AGAP, il faut formaliser ces patrons à l'aide de P-Sigma, c'est-à-dire définir le contenu des parties « interface », « réalisation » et « relations » (Figure 35).

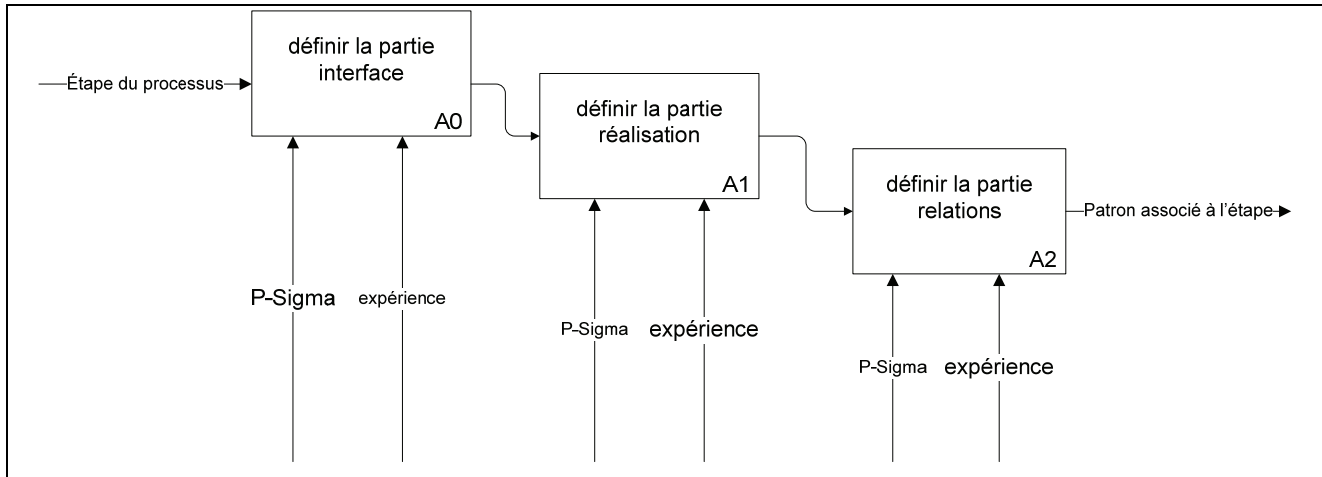


Figure 35 : aider le concepteur

Malgré les nombreux catalogues de patrons, nous n'avons pas trouvé de catalogues spécifiques à la conception de sites web. Ainsi, de la même manière que pour le processus de conception, il faut les formaliser à partir de notre propre expérience.

2 Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »

La mise en œuvre de cette fonctionnalité nécessite de spécifier comment identifier et proposer des solutions de conception et comment faciliter leur réutilisation quand elles sont choisies par le concepteur.

2.1 Identifier et proposer des solutions de conception

Pour identifier et proposer des solutions de conception le long du processus de conception, il faut définir un modèle intermédiaire entre l'utilisateur et les patrons associés aux étapes de ce processus.

Cependant, si on considère que notre objectif est de proposer des solutions de conception utilisées dans des sites web en production, il faut définir non pas un modèle fonctionnel de sites web, mais un modèle d'analyse de sites web en production. Ensuite, pour proposer des solutions de conception à partir de ce modèle d'analyse, il faut définir (Figure 36) :

- l'utilisation du modèle pour faire des analyses ;
- l'exploitation de ces analyses.

De cette manière, l'identification de solutions de conception est réalisée à partir de solutions utilisées en production, et ne nécessitent pas d'expertise dans l'approche patron pour être identifiées.

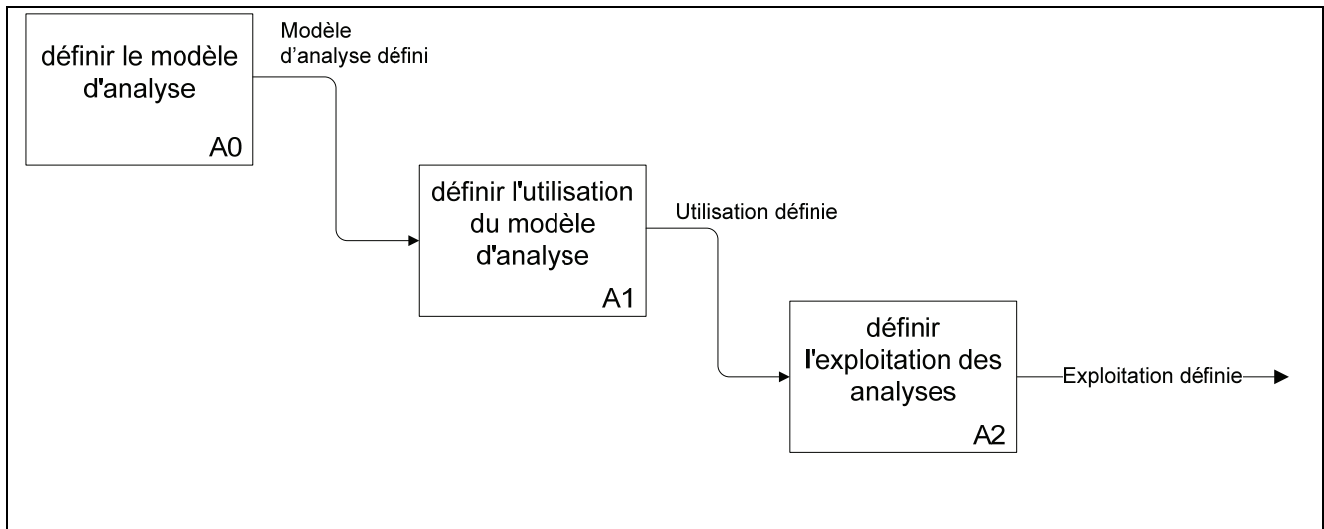


Figure 36 : modèle d'analyse de sites web

2.1.1 Définir le modèle d'analyse

Pour définir le modèle d'analyse, il faut modéliser les solutions de conception en considérant les mêmes quatre dimensions qui sont utilisées pour modéliser un site web (« navigation », « présentation », « contenu », et « implémentation »). De cette manière, la correspondance entre les solutions de conception analysées et les entités définies lors de la conception d'un site web seront naturelle.

2.1.2 Définir l'utilisation du modèle d'analyse

Pour permettre l'utilisation du modèle d'analyse, il faut construire une interface utilisateur qui, en fonction de la dimension considérée, propose à l'utilisateur :

- les solutions de conception qui ont déjà été identifiées dans les analyses de sites web de la même catégorie ;
- les fonctions interactives suivantes afin de réaliser l'analyse d'un site web : sélection des solutions dans la liste et insertion de solutions qui n'ont jamais été définies dans des sites web analysés.

Cependant, dans un souci de généricité de la liste, les solutions doivent être :

- identifiées par un label général auquel différentes solutions de site web peuvent être rattachées tout en ayant un label différent ;
- être explicitées par une définition afin de lever toute ambiguïté éventuelle sur la signification du label.

2.1.3 Définir l'exploitation des analyses

L'exploitation des analyses consiste à proposer les solutions de conception par rapport à une dimension. Pour cela, les analyses indiquant la présence de ces solutions pour chacun des sites web analysés, il faut calculer le nombre de fois que chaque solution de la liste a été sélectionnée lors des analyses de sites web. Cela permet d'indiquer immédiatement celles que l'on retrouve en général dans les sites web d'une catégorie et de ne pas les oublier. Dans la suite de ce document, nous appelons cette valeur « **indice de réutilisation** ».

Après avoir défini comment identifier et proposer des solutions de conception, la partie suivante présente nos spécifications pour faciliter la réutilisation de ces solutions.

2.2 Faciliter la réutilisation des solutions de conception

Pour faciliter la réutilisation des solutions de conception, il faut définir la sélection, l'instanciation et l'adaptation des solutions proposées au concepteur (Figure 37).

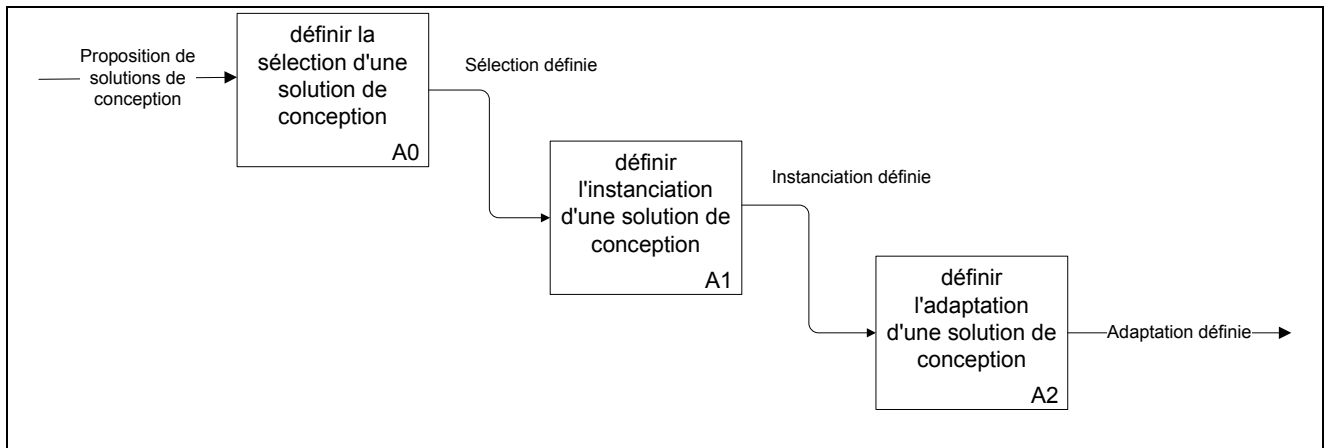


Figure 37 : réutilisation de solutions de conception

2.2.1 Définir la sélection d'une solution de conception

Pour sélectionner une solution de conception selon une dimension de modélisation, il faut proposer dans une interface utilisateur :

- la liste des solutions dont la présence a été identifiée lors des analyses (nous l'appelons « **liste des propositions** ») ;
- des fonctions interactives afin de les sélectionner pour le site en cours de conception.

2.2.2 Définir l'instanciation d'une solution de conception

Afin d'adapter une solution au site en cours de conception, il faut instancier cette solution et l'associer ensuite au modèle de conception du site web. L'ensemble formé par ces instanciations est appelé « **liste des sélections** ». Ensuite, afin de supporter les itérations dans le processus de conception qui peut conduire à une modification de la « liste des propositions », il faut garder une traçabilité de cette instanciation afin de répercuter ces modifications dans la « liste des sélections ».

2.2.3 Définir l'adaptation d'une solution de conception

Afin de permettre l'adaptation d'une solution de conception au site en cours de conception, il faut donner le moyen d'adapter la « liste des sélections ». Pour cela, il faut offrir dans une interface utilisateur les fonctions suivantes :

- modifier le label d'une solution sélectionnée ;
- insérer une nouvelle solution de conception dans la « liste des sélections » ;
- modifier automatiquement la « liste des propositions » si la « liste des sélections » a été modifiée.

3 Fonctionnalité « Identifier et faciliter le choix de composants logiciels »

La mise en œuvre de cette fonctionnalité nécessite de spécifier comment identifier des composants logiciels, et comment faciliter leur choix par l'utilisateur [Cocquebert, 07a] [Cocquebert, 07b].

3.1 Identifier des composants logiciels

Selon l'approche orientée modèle analysée dans l'état de l'art, pour identifier et proposer des composants logiciels, il faut définir un modèle de composant et un modèle intermédiaire, ce dernier permettant de minimiser le nombre de composants satisfaisant un ensemble de fonctions. Or, dans notre contexte, l'objectif est opposé : nous voulons identifier un ensemble de composants satisfaisant une seule fonction, celle qui a été associée à l'item de menu. Ainsi, pour identifier et proposer des composants logiciels, il faut uniquement spécifier la définition, l'utilisation et l'exploitation d'un modèle de composant (Figure 38) car nous ne voulons pas utiliser les capacités du modèle intermédiaire à minimiser le nombre de composants.

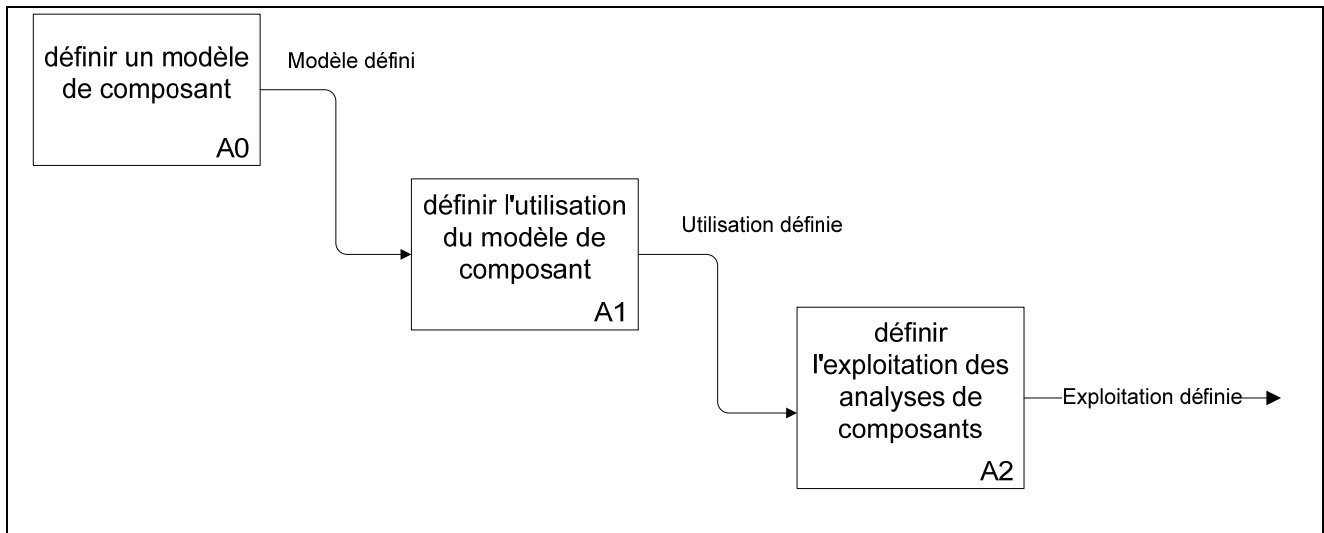


Figure 38 : identifier des composants logiciels

3.1.1 Définir un modèle de composant

Pour définir un modèle de composant, il faut identifier les caractéristiques fonctionnelles et non fonctionnelles des composants qui correspondent aux informations utiles au choix et à l'intégration du composant au sein de l'architecture du site web.

3.1.2 Définir l'utilisation du modèle de composant

Pour utiliser ce modèle afin de caractériser des composants existants, il faut proposer dans une interface utilisateur :

- la liste des caractéristiques fonctionnelles et non fonctionnelles du modèle ;
- les fonctions interactives standards de gestion d'un formulaire (insertion, modification et contrôle de la saisie de valeurs) afin de caractériser un composant analysé.

3.1.3 Définir l'exploitation des analyses de composants

Pour exploiter les analyses de composants, il faut :

- définir une fonction de recherche sur une ou plusieurs facettes du modèle afin d'identifier un ensemble de composants ;
- donner la possibilité à partir du résultat de la recherche de consulter l'ensemble des caractéristiques des composants afin de choisir un ou plusieurs composants parmi la liste et d'affiner ensuite le choix par un test réel.

Après avoir spécifié comment identifier des composants logiciels, la partie suivante présente comment faciliter leur choix.

3.2 Faciliter le choix de composants

Pour faciliter le choix d'un composant logiciel, il faut utiliser un processus automatique car il ne dépend pas du nombre de composants ou de catégories. Cependant, comme on ne peut pas réutiliser les méthodes ad-hoc telles que STACE, OTSO, PORE, etc. car elles sont spécifiques à leur contexte d'utilisation, il faut définir ce processus au préalable (Figure 39).

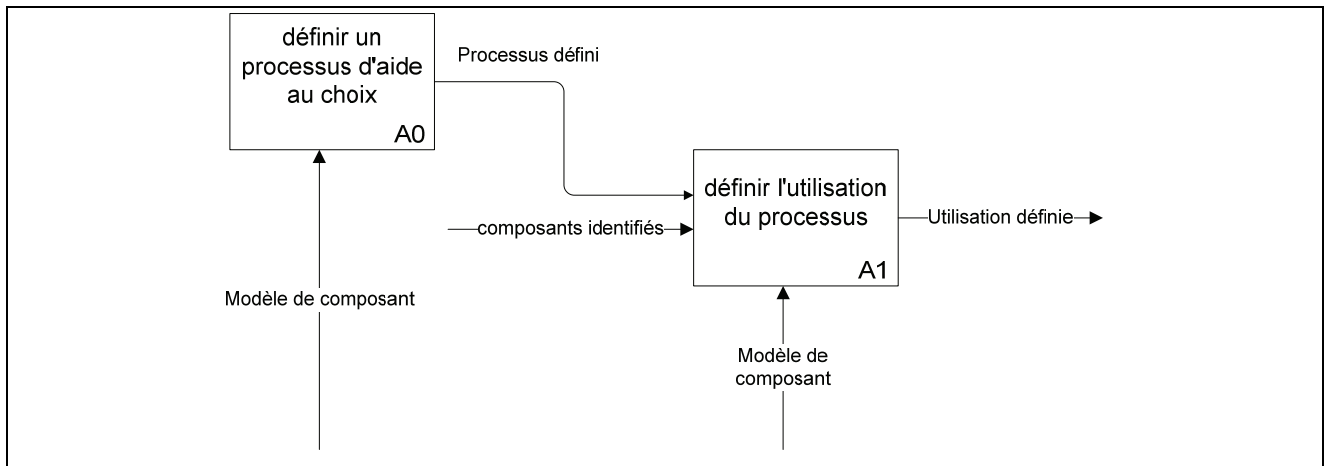


Figure 39 : faciliter le choix d'un composant logiciel

Si on considère que l'exploitation des analyses se fait sous la forme d'une recherche simple sur les facettes du modèle de composants, le processus automatique doit se présenter sous la forme d'une recherche avancée multicritère à partir de la définition itérative des informations suivantes :

- pondération sur les facettes du modèle de composants ;
- valeur minimale pour une ou plusieurs facettes numériques du modèle de composants ;
- valeur spécifique pour une ou plusieurs facettes textuelles du modèle de composants.

De cette manière, le processus va identifier le composant (ou un ensemble restreint de composants), à partir d'un calcul analytique sur les valeurs des facettes.

4 Fonctionnalité « Accélérer la mise en place d'un site web »

La mise en œuvre de cette fonctionnalité nécessite de spécifier comment générer une structure de données à partir des choix de conception et comment accélérer la conception d'un site web.

4.1 Génération de structures de données

Pour générer des structures de données à partir du modèle de données initial, il faut (Figure 40) :

- identifier les données qu'il est possible de générer de manière automatique ;
- identifier la source dans le modèle de données initial et le résultat du traitement ;
- définir et développer la génération de données elle-même.

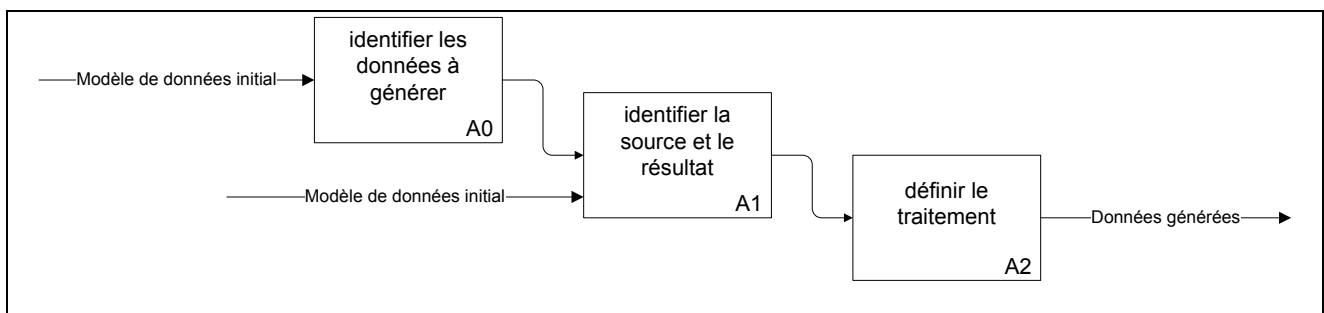


Figure 40 : génération de structures de données

Afin d'assurer une traçabilité entre les données sources et le résultat, il faut que les modifications sur la source soient répercutées dans le résultat de manière automatique et inversement.

4.2 Accélérer la conception d'un site web

Pour accélérer la conception d'un site web en prenant en compte l'existence de composants logiciels, il faut définir une architecture d'accueil ouverte pour ne pas bloquer l'intégration de développements externes et contenant un framework de classes intégré.

Nos spécifications sont de (Figure 41) :

- définir un modèle d'architecture d'accueil ;
- définir l'instanciation du modèle d'architecture d'accueil ;
- définir l'utilisation de l'architecture d'accueil instanciée.

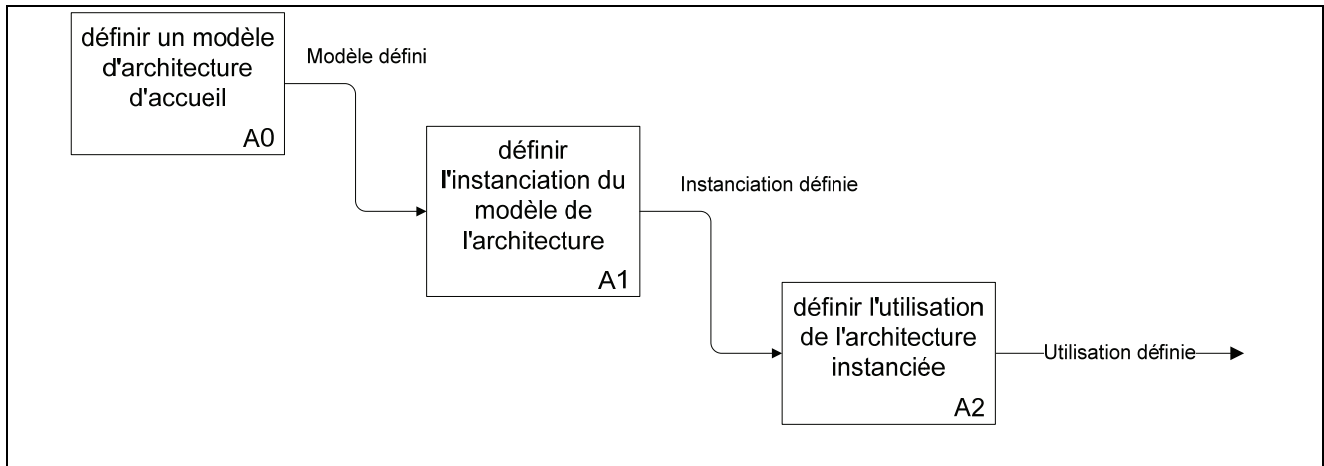


Figure 41 : accélérer la mise en place d'un site web

4.2.1 Définir un modèle d'architecture d'accueil

En nous basant sur notre expérience de développement de sites web et sur notre étude d'outils de développement de sites web, le modèle de l'architecture d'accueil doit répondre aux nécessités suivantes :

- *Etre organisée.* L'architecture doit être organisée afin de faciliter la prise en main par un développeur. Elle ne doit pas bloquer l'intégration d'une mise en œuvre externe dans l'architecture logicielle du site web en imposant, par exemple, la séparation de fichiers concernant l'affichage, le traitement des données et l'interaction avec la base de données.
- *Permettre le développement rapide des fichiers.* Afin de bénéficier du framework de classes intégré à l'architecture et permettre un développement rapide des fichiers d'un nouveau site web, il faut définir un modèle de fichier à personnaliser lors de la mise en œuvre d'une nouvelle page web associée à un item de menu. Ce modèle doit contenir une partie non modifiable qui fait le lien avec l'architecture logicielle, et une partie modifiable dans laquelle le développement spécifique doit être programmé.
- *Etre facilement extensible.* Afin de faciliter l'intégration d'un développement existant au sein de l'architecture logicielle, il faut définir une connexion simple de ce développement existant avec l'architecture logicielle. Ainsi :
 - les variables et méthodes de l'architecture sont utilisables au sein de ce développement ;
 - ce développement peut être adapté ou enrichi en utilisant les variables et méthodes de l'architecture.

4.2.2 Définir l'instanciation du modèle de l'architecture

Pour instancier le modèle de l'architecture d'accueil, il faut développer un processus qui :

- instancie le modèle de l'architecture définie ci-dessus ;
- génère les fichiers associés aux items du menu de navigation à partir du modèle de fichier fourni dans d'architecture d'accueil ;
- génère les structures de données associées aux choix de conception réalisés au cours du processus.

4.2.3 Définir l'utilisation de l'architecture instanciée

Pour utiliser l'architecture d'accueil, il faut :

- personnaliser les fichiers associés aux items en programmant l'interaction avec les structures de données, le traitement et l'affichage des données en utilisant si possible le framework de classes intégré ;
- intégrer si nécessaire des développements existants en utilisant le moyen fourni par l'architecture pour le faire simplement.

5 Conclusion

L'objectif de ce chapitre était de présenter la démarche pour définir une méthode d'aide à la conception de sites web en nous appuyant sur l'état de l'art présenté dans le chapitre II. Elle a été décrite sous la forme d'un ensemble de processus pour mettre en œuvre chacune des quatre fonctionnalités que nous avons identifiées. Nous synthétisons ci-dessous les résultats de ce chapitre [Cocquebert, 08b].

Fonctionnalité « Assister la réalisation du processus de conception ». Pour guider l'utilisateur, il faut définir notre propre processus de conception à partir de notre expérience car les processus de conception existants ne sont pas adaptés ; pour modéliser les données d'un site web utilisées le long de ce processus, il faut définir les données selon les dimensions « navigation », « présentation », « contenu » et « implémentation » ; pour aider à la réalisation d'une étape, il faut formaliser notre expérience dans des patrons à l'aide de P-Sigma, et associer ces patrons aux étapes du processus de conception.

Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception ». Pour ne pas imposer une expertise pour l'identification des patrons et la proposition des solutions de conception, il faut définir un modèle d'analyse de sites web en production afin de caractériser les solutions de conception par rapport aux dimensions de modélisation d'un site web. Pour faciliter la réutilisation des solutions de conception, il faut implémenter des mécanismes pour adapter les solutions proposées à partir des analyses des sites web, en assurant la traçabilité des adaptations par rapport aux solutions proposées.

Fonctionnalité « Identifier et faciliter le choix de composants logiciels ». Pour identifier et proposer des composants logiciels, il faut définir un modèle de composant pour caractériser les composants et donner accès à ces informations. Pour faciliter le choix d'un composant, il faut implémenter un processus analytique de recherche multicritère afin de définir des pondérations ou des valeurs spécifiques sur une ou plusieurs caractéristiques du modèle de composants.

Fonctionnalité « Accélérer la mise en place d'un site web ». Pour offrir cette fonctionnalité, il faut générer automatiquement des données d'implémentation à partir de choix de conception réalisés, et il faut générer une architecture logicielle d'accueil qui soit :

- simple à prendre en main et à adapter pour développer un nouveau site web par réutilisation de fonctions de bas niveau ;
- suffisamment ouverte pour permettre l'intégration de développement existant.

L'objectif du chapitre suivant est de détailler les différents éléments d'une méthode en nous appuyant sur cette démarche.

Chapitre IV : Proposition

En s'inspirant de la démarche que nous venons de présenter, ce chapitre présente les éléments constitutifs d'une méthode d'aide à la conception de sites web que nous avons appelé WISDOM, afin de mettre en œuvre les fonctionnalités que nous avons identifiées. Il est donc également structuré par ces fonctionnalités.

Notre proposition est basée sur notre réflexion par rapport à :

- la conception des onze sites web suivants :
 - les éditions 2005²⁵ et 2007²⁶ du congrès national « LaPlagne », congrès national avec comité de lecture sur la conception mécanique organisé tous les deux ans par l'AIP-PRIMECA,
 - les éditions 2003²⁷, 2005²⁸ et 2007²⁹ du congrès francophone « PENTOM », congrès francophone sur les « Nouvelles technologies en maintenance »,
 - la journée technique (JT) organisées sous l'égide de l'AIP-PRIMECA sur les SGDT (JT SGDT 04³⁰),
 - les journées semestrielles STP du GDR MACS organisées à Valenciennes en novembre 2006 (STP'06³¹) et les journées prospective STP à Lyon (Lyon '07),
 - le pôle ST2 (Pôle "Sciences et Technologies pour la Sécurité dans les Transports")³²,
 - le RTP-PCM, Réseau Thématique Pluridisciplinaire « Production Coopérative Médiatisée », lancé par le CNRS de 2003 à 2005 (le site web est fermé),
 - le GDR MACS³³, Groupe De Recherche « Modélisation, Analyse et Conduite de Systèmes dynamiques », regroupant sous l'égide du CNRS depuis 2002 la communauté productique et automatique ;
- la conception de composants métiers concernant la gestion de la relecture des communications et des inscriptions pour :
 - les éditions 2005 et 2007 du congrès national « LaPlagne »,
 - les éditions 2003, 2005 et 2007 du congrès francophone « PENTOM »,
 - l'édition 2007 de « CPI³⁴ »,
 - l'édition 2008 de « MOSIM³⁵ »,
 - Les journées techniques AIP-PRIMECA sur les SGDT.

Dans la proposition de WISDOM présentée dans ce chapitre, nous n'avons pas considéré nos spécifications pour la recherche multicritère qui facilite le choix de composants logiciels dans un contexte donné. En effet, elles nécessitent une capitalisation de règles métiers pour considérer plusieurs facettes du modèle de composant, ce qui sort du cadre de notre étude. De plus, cette recherche constitue plus une amélioration de la méthode qu'un point fondamental de sa définition.

Nous avons choisi UML comme formalisme de modélisation³⁶ car il est devenu un outil standard *de facto* pour la modélisation.

1 Fonctionnalité « Assister la réalisation du processus de conception »

Pour mettre en œuvre cette fonctionnalité, nous proposons :

- les étapes du processus de conception, le processus lui-même et le modèle de données utilisé le long du processus pour guider le concepteur dans la conception d'un site web ;
- des patrons de domaines pour aider à la réalisation des étapes.

²⁵ <http://www.univ-valenciennes.fr/laplagne2005/>

²⁶ <http://laplagne2007.insa-lyon.fr/>

²⁷ <http://www.univ-valenciennes.fr/sp/PENTOM2003/>

²⁸ <http://www.univ-valenciennes.fr/sp/PENTOM2005/>

²⁹ <http://www.univ-valenciennes.fr/sp/PENTOM2007/>

³⁰ <http://www.univ-valenciennes.fr/JTSGDT/>

³¹ <http://www.univ-valenciennes.fr/congres/STP06/>

³² <http://www.polest2.fr/>

³³ <http://www.univ-valenciennes.fr/GDR-MACS/>

³⁴ <http://www.supmeca.fr/cpi2007/>

³⁵ <http://www.mosim08.org/>

³⁶ cf. par exemple à [Muller, 04] pour des détails sur ce formalisme.

1.1 Etapes et processus de conception

Nous proposons dans cette partie les étapes et le processus de conception afin de guider le concepteur dans la conception d'un nouveau site web [Cocquebert, 07a] [Cocquebert, 07b] [Cocquebert, 08b]. Les étapes et le processus étant étroitement liés, nous présentons les étapes du processus dans un déroulement type du processus de conception d'un site web. Formalisé sous la forme d'un diagramme d'activité UML, ce processus itératif est présenté Figure 42. Dans la suite de cette partie, chaque étape est présentée de la manière suivante : description de son objectif général, puis détail des informations qui la concerne (« données en entrée », « données en sortie », « fonction » et « étape suivante »).

Etape 1 « définition du contexte » (cf. détails dans Tableau 3) : elle définit les paramètres du contexte de la conception.

Donnée en entrée	Cahier des charges du site web.
Fonction	Définition de : <ul style="list-style-type: none"> – la catégorie du site web afin de sélectionner les solutions de conception ; – la valeur de filtrage³⁷ afin de garantir la pertinence des propositions de solutions de conception.
Données en sortie	Catégorie du site web à concevoir et valeur de filtrage.
Etape suivante	Etape 2 pour concevoir/modifier la navigation.

Tableau 3 : détails de l'étape 1 « définition du contexte »

Etape 2 « conception/modification de la navigation » (cf. détails dans le Tableau 4) : elle permet la conception ou la modification du menu du site web, afin d'assurer la consultation des informations disponibles ou l'utilisation d'applicatifs spécifiques.

Données en entrée	Paramètres du contexte de la conception, complétés par : <ul style="list-style-type: none"> – le cahier des charges donné par le client sur les items potentiels, les fonctions à mettre en œuvre et les contenus à afficher sur le site (cas d'une nouvelle conception) ; – le modèle « navigation » de la conception en cours ou issu d'une sauvegarde antérieure (cas de la modification).
Fonction	<ul style="list-style-type: none"> – identification et proposition de solutions de conception selon la dimension « navigation » en tenant compte de la valeur de filtrage et de la catégorie du site web ; – gestion de l'interface utilisateur pour en permettre l'adaptation.
Données en sortie	Structure de données du menu, fichiers associés aux items du menu, architecture d'accueil.
Etape suivante	Selon les cas : <ul style="list-style-type: none"> – étape 3 si la présentation n'est pas définie ; – étape 5 si la présentation est définie ; – elle-même si la navigation doit être modifiée.

Tableau 4 : détails de l'étape 2 « conception/modification de la navigation »

³⁷ Cette « valeur de filtrage » est présentée en détail dans le paragraphe 12.1.3 p. 90

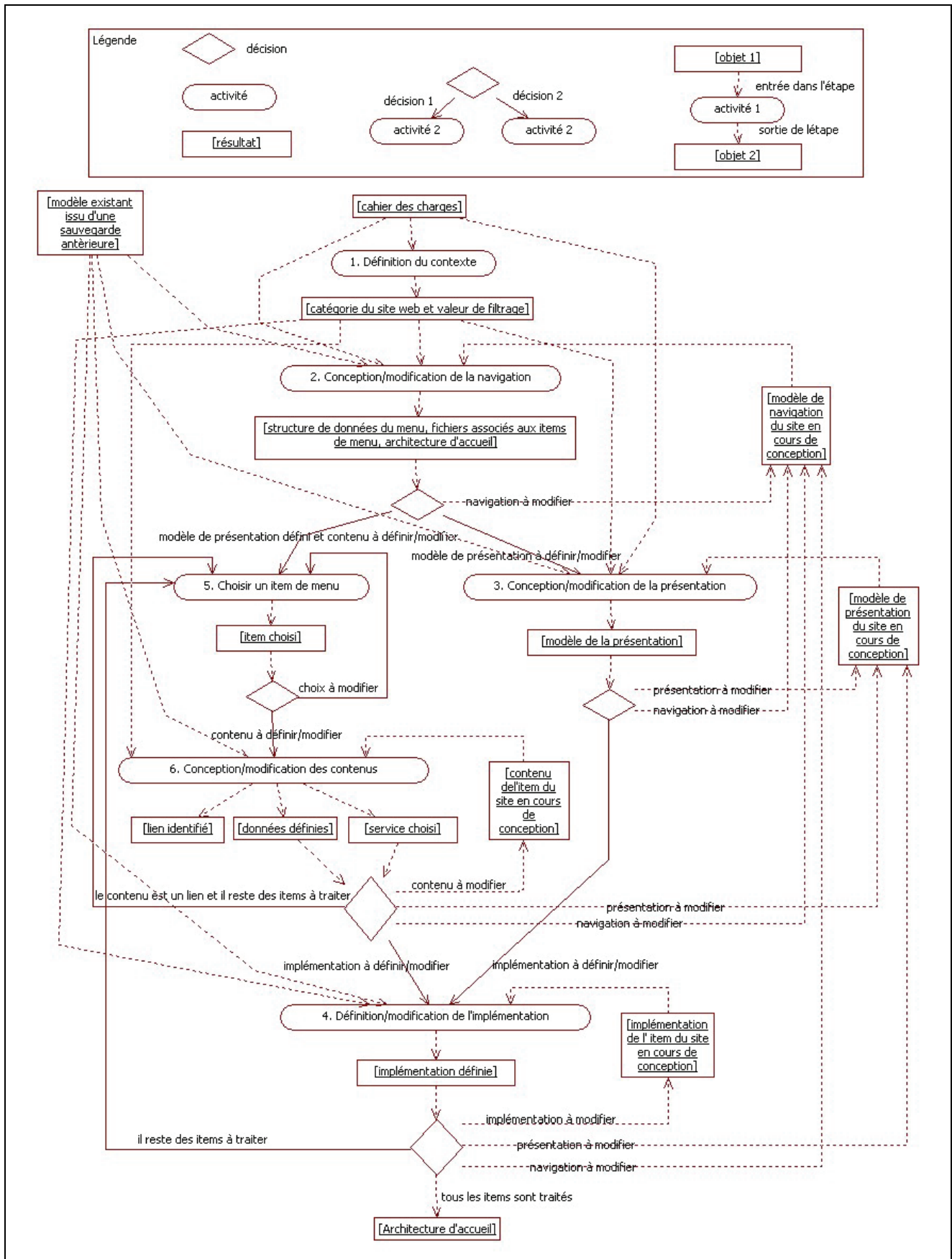


Figure 42 : diagramme d'activité du processus de conception d'un site web

Etape 3 « conception/modification de la présentation » (cf. détails dans le Tableau 5) : elle permet la conception ou la modification du modèle de présentation des pages du site web.

Données en entrée	Paramètres du contexte de la conception, complétés par : – le cahier des charges fourni par le client sur la technologie support de conception du site web (FLASH, « graphique », « web 2.0 », « standard », etc.) (cas d'une nouvelle conception) ; – le modèle « présentation » de la conception en cours ou issu d'une sauvegarde antérieure (cas de la modification).
Fonction	– identification et proposition de solutions de conception selon la dimension « présentation » en tenant compte de la valeur de filtrage et de la catégorie du site web ; – gestion de l'interface utilisateur pour en permettre l'adaptation.
Donnée en sortie	Modèle de présentation
Etape suivante	Selon les cas : – étape 4 pour définir/modifier une implémentation ; – elle-même si le modèle de présentation doit être modifié ; – étape 2 si la navigation doit être modifiée.

Tableau 5 : détails de l'étape 3 « conception/modification de la présentation »

Etape 4 « définition/modification d'une implémentation » (cf. détails dans Tableau 6) : cette étape fait suite à l'étape 3 pour implémenter un menu, ou à l'étape 6 pour implémenter un affichage de données ou d'un service. Elle permet de réaliser ou de modifier le choix d'un composant métier pour implémenter un service, ou d'un composant technique pour implémenter un menu ou l'affichage de données.

Données en entrée	Paramètres du contexte et l'élément à implémenter (menu, données, service) dans le cas de la définition d'une implémentation, complétés par les données sur l'implémentation de l'élément du site en cours de conception ou issu d'une conception sauvegardée dans le cas de la modification d'une implémentation.
Fonction	– identification et proposition de composants techniques ou métiers ; – gestion de l'interface utilisateur pour les proposer à l'utilisateur.
Donnée en sortie	Implémentation choisie.
Etape suivante	Selon les cas : – étape 5 s'il reste encore des items qui n'ont pas de contenu ; – elle-même si le choix de l'implémentation doit être modifié ; – étape 3 si le modèle de présentation doit être modifié ; – étape 2 si la navigation doit être modifiée.

Tableau 6 : détails de l'étape 4 « définition/modification d'une implémentation »

Etape 5 « choisir un item de menu » (cf. détails dans Tableau 7) : elle permet de choisir un item du menu pour y associer un contenu ou pour en modifier le contenu associé.

Données en entrée	Modèle « navigation » pour choisir un item.
Fonction	gestion de l'interface utilisateur pour choisir un item de menu et afficher le suivi des associations de contenus.
Donnée en sortie	Item choisi.
Etape suivante	Selon les cas : – étape 6 pour concevoir le contenu ; – elle-même si le choix doit être modifié.

Tableau 7 : détails de l'étape 5 « choisir un item de menu »

Etape 6 « conception/modification d'un contenu » (cf. détails dans Tableau 8) : elle permet d'associer un contenu à un item du menu ou de modifier ce contenu.

Données en entrée	Paramètres du contexte, complétés par l'item du menu sans contenu associé (nouvelle conception), ou avec les données du contenu associé (modification de la conception en cours ou reprise d'une conception sauvegardée).
Fonction	– identification et proposition de contenu selon la dimension « contenu » en tenant compte de la valeur de filtrage ; – gestion de l'interface utilisateur pour en permettre la conception ou l'adaptation.

Donnée en sortie	Contenu défini (lien identifié, données définies ou service choisi).
Étape suivante	<p>Selon les cas :</p> <ul style="list-style-type: none"> – étape 4 pour implémenter l’affichage du contenu ; – elle-même si le contenu doit être modifié ; – étape 5 si le contenu associé est un lien et qu’il reste des items à traiter ; – étape 3 si le modèle de présentation doit être modifié ; – étape 2 si la navigation doit être modifiée.

Tableau 8 : détails de l’étape 6 « conception/modification d’un contenu »

Lors d’un déroulement normal de la conception, la sortie du cycle se fait à la sortie de l’étape 4 quand un contenu est associé à tous les items de menu. Le résultat est alors une architecture d’accueil dans laquelle :

- l’interaction avec la base de données doit être programmée ;
- l’utilisation des composants techniques ou des composants métiers doit être intégrée.

La sauvegarde étant continue lors du processus, le concepteur peut interrompre le processus à tout moment et le reprendre ensuite là où il s’était arrêté.

1.2 Modèle de site web

Les données utilisées le long du processus de conception d’un site web sont modélisées selon les dimensions « navigation », « présentation », « contenu », et « implémentation »³⁸ [Cocquebert, 07a] [Cocquebert, 07b].

Dimension « navigation ». La modélisation associée à cette dimension est présentée Figure 43. Selon ce modèle, instancié lors de l’étape 2 « conception/modification de la navigation », un *site web* est caractérisé par sa catégorie, son nom et son adresse internet (url) et contient un *menu*.

Ce menu est défini à l’aide d’une adaptation du patron « composite » [Gamma, 95] afin de représenter la navigation sous la forme d’un arbre sans limiter la profondeur du menu. Cela nous a conduit à relier *menu* à *item général*, classe issue de cette adaptation.

L’implémentation de ce menu est définie à l’aide d’une adaptation du patron « stratégie » [Gamma, 95] afin de la rendre indépendante de la définition du menu. Nous avons ainsi relié *menu* à deux exemples de types d’affichage (*onglets* et *menu déroulant*) à travers *implémentation*, classe issue de cette adaptation.

Le type de *contenu « donnée »* est associé à *item général* afin de pouvoir l’associer à un item ou à un groupe d’item (dans ce dernier cas, pour afficher par exemple sur une seule page les nouveautés des items du groupe). Par contre, les types de *contenu « service »* et « *lien externe* » sont associés uniquement à *item* car ils y sont spécifiques.

Un *item général* est associé à un *fichier*, définissant une *page web* du *site web*.

³⁸ Dans le descriptif d’un modèle, un mot écrit dans la police *courrier italique* fait référence à la classe de même nom dans la figure associée au texte

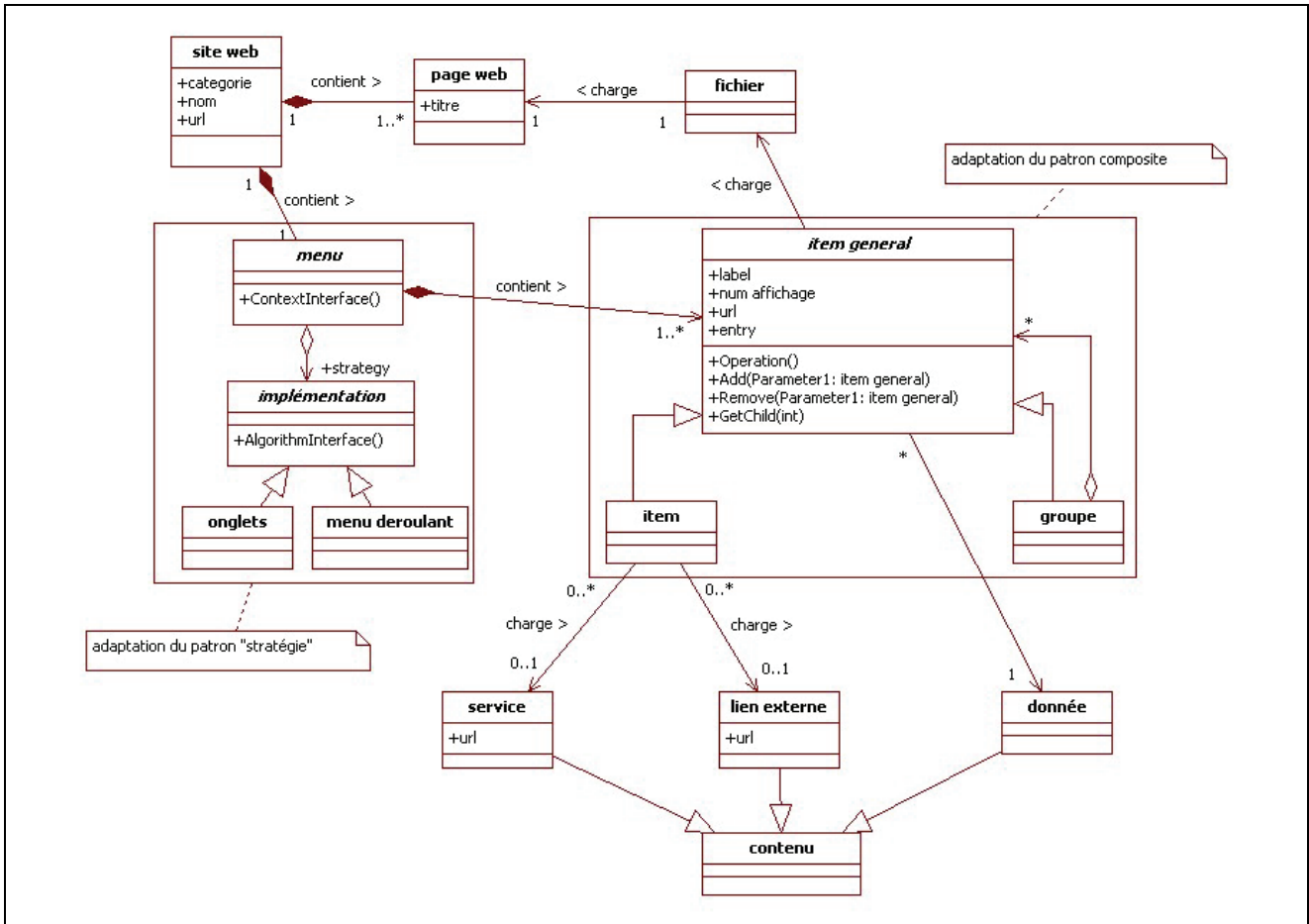


Figure 43 : modèle associé à la dimension « navigation »

Dimension « présentation ». La modélisation associée à cette dimension est présentée Figure 44. Selon ce modèle, instancié lors de l'étape 3 « conception/modification de la présentation » :

- toutes les instances de *page web* sont associées à un *modèle* de présentation ;
- un *modèle* est composé d'une ou de plusieurs *zones* d'affichage de données ;
- une *zone* affiche une ou plusieurs données ;
- Une *donnée* peut être associée à un *lien externe* ou à un *service* (par exemple : « HAL³⁹ »).

Afin également de rendre l'affichage du contenu de la zone indépendant de sa définition, nous avons intégré à notre diagramme de classe une autre adaptation du patron « stratégie » [Gamma, 95]. De cette manière, nous avons relié *donnée* à deux exemples d'affichage (*animation* et *tableau*) à travers *implémentation*, classe issue de l'adaptation de ce patron.

³⁹ <http://hal.archives-ouvertes.fr/>

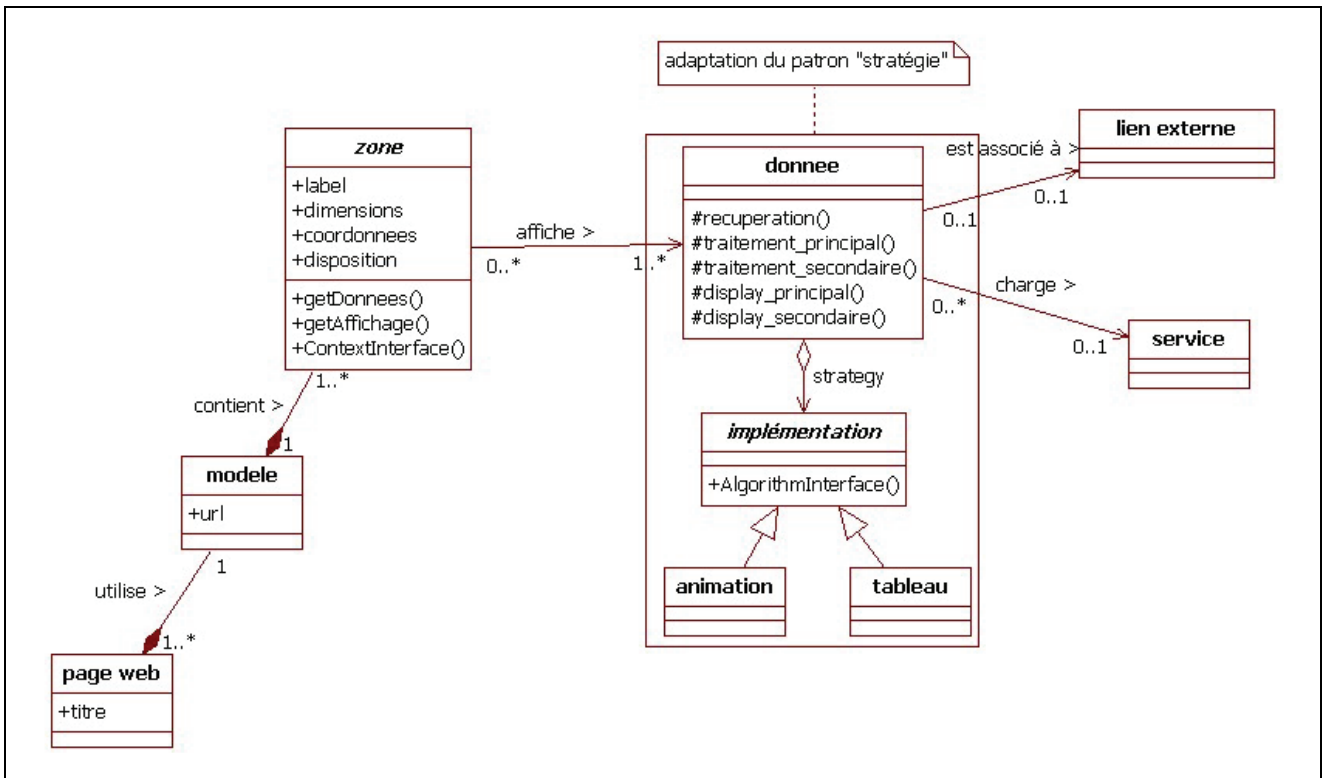


Figure 44 : modèle associé à la dimension « présentation »

Dimension « contenu ». La modélisation associée à cette dimension est présentée Figure 45. Selon ce modèle, instancié lors de l'étape 6 « conception/modification des données », un contenu peut être un *service*, un *lien externe* ou une *donnée*.

Dans les deux premiers cas, *service* et *lien externe* sont définis par une adresse internet (url).

Dans le troisième cas, une *donnée* est *statique* ou *dynamique*. Si elle est dynamique, elle est :

- *principale* si elle permet de différencier un enregistrement dans une table ;
- *secondaire* si elle précise la définition d'une donnée principale ;
- *interne* si elle n'est pas affichée.

Les méthodes de gestion de ces données (récupération, traitement et affichage des données principales et secondaires) sont définies de manière protégée dans la classe *donnée* afin de :

- les appeler de manière générique dans un fichier modèle associé aux items ;
- définir le corps de ces méthodes dans les classes filles.

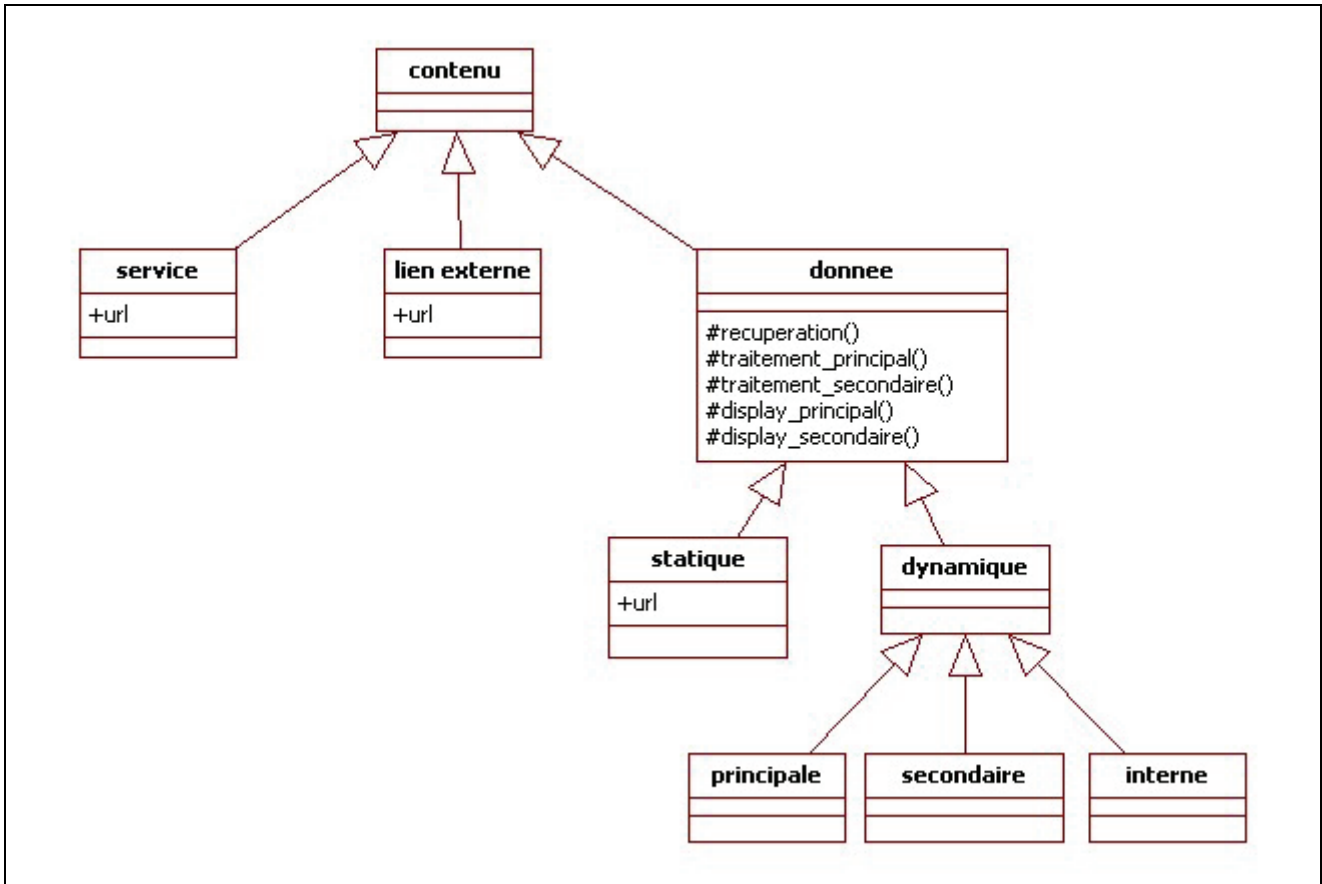


Figure 45 : modèle associé à la dimension « contenu »

Dimension « implémentation ». La modélisation de cette dimension est présentée Figure 46. Selon ce modèle qui est instancié lors de l'étape 4 « définition/modification de l'implémentation », une *implémentation* affiche une *donnée* (dans une zone) et met en œuvre un *service*. Une *implémentation* est réalisée par un *composant* référencé dans un *catalogue* ou par un *développement spécifique*.

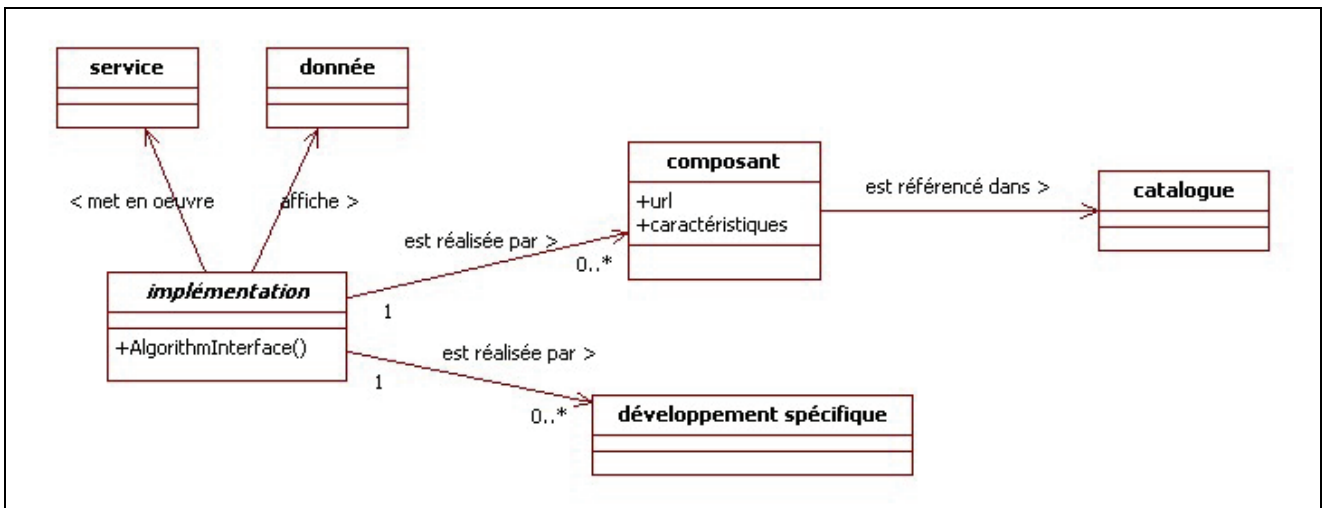


Figure 46 : modèle associé à la dimension « implémentation »

La présentation globale des modèles des différentes dimensions est présentée en Annexe 1.

1.3 Patrons de domaine

Afin d'assister le concepteur dans la réalisation des étapes du processus de conception, nous avons associé un patron de domaine aux étapes reliées aux dimensions de modélisation (« navigation », « présentation », « contenu », « implémentation »). Ces patrons sont formalisés en P-Sigma [Conte, 01a], dont la définition est rappelée en Annexe 2. Dans le contexte de notre étude, les parties « interface », « réalisation » et « relations » ont les objectifs suivants :

- la partie « interface » définit pour chaque étape son objectif, les entrées et les résultats ;
- la partie « réalisation » définit la réalisation de l'étape elle-même (processus, diagrammes de données, etc.) ;
- la partie « relation » définit les relations du patron avec d'autres patrons.

Comme les rubriques des parties « interface » et « relations » ont été abordées dans la présentation des étapes (paragraphe 1.1 p. 72), nous nous focalisons dans la suite de notre présentation sur la partie « réalisation » du formalisme P-Sigma, et plus particulièrement sur la rubrique « processus » [Cocquebert, 08b]. Néanmoins, la formalisation complète des patrons est disponible en Annexe 3.

De manière analogue aux étapes et au processus de conception, les processus sont présentés sous la forme d'un scénario UML, suivi d'une description de chacune des étapes.

1.3.1 Patron « navigation »

Le processus défini par ce patron est présenté Figure 47.

Si le menu du site en cours de conception n'est pas encore défini, le processus débute par l'étape 1a ; si le menu est déjà défini, le processus débute à l'étape 1b.

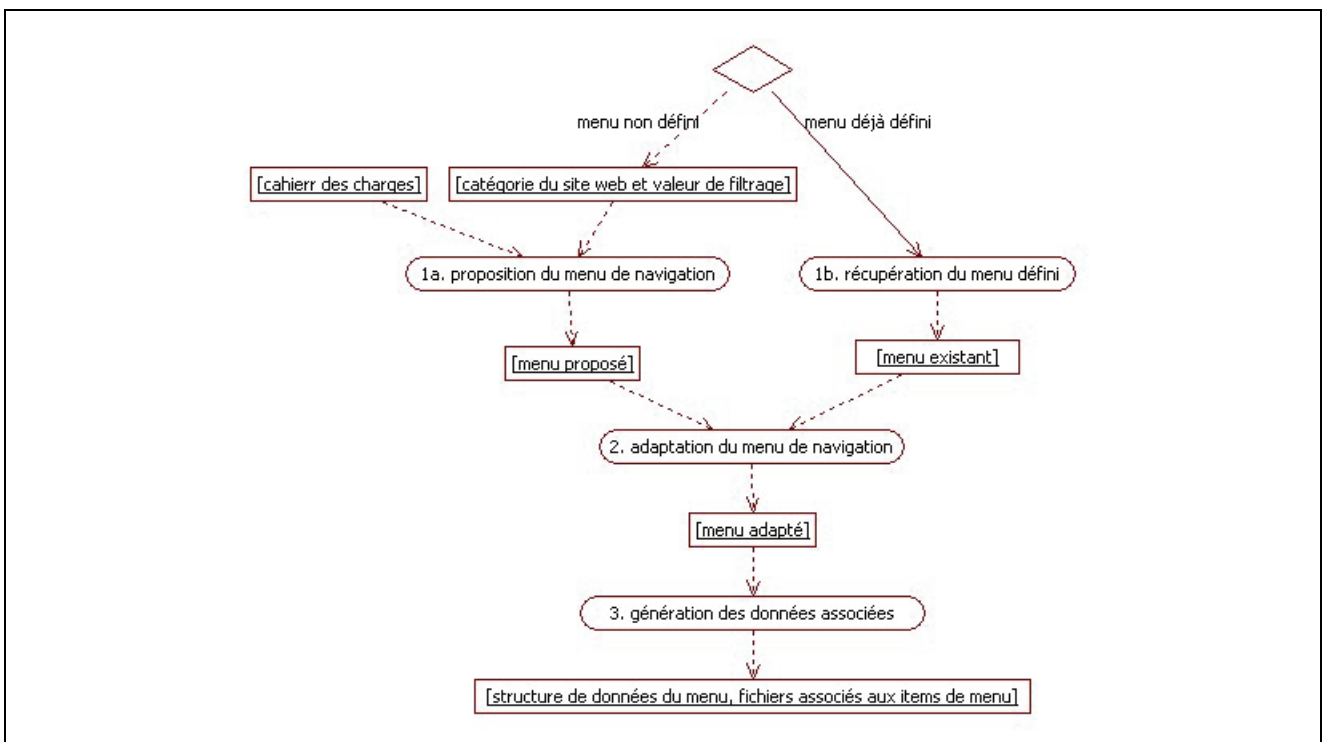


Figure 47 : processus du patron navigation

Etape 1

- a. « Proposition du menu de navigation » (cf. détails dans Tableau 9) : elle propose l'ensemble des items de menu qui ont été identifiés dans les analyses des sites web de la même catégorie.

Données en entrée	Cahier des charges sur les items potentiels du site web, catégorie du site web en cours de conception et valeur de filtrage.
Fonction	<ul style="list-style-type: none"> – construction dynamique de la liste des items de menu à partir de l’analyse des menus des sites web selon la dimension « navigation » ; – calcul de l’indice de réutilisation de chacun des items de la liste (l’indice de réutilisation d’un item étant le nombre de fois qu’il a été identifié dans les analyses des sites web).
Donnée en sortie	Liste des items de menus dont les indices de réutilisation sont supérieurs à la valeur de filtrage, ordonnée de manière décroissante selon les indices de réutilisation.
Etape suivante	Etape 2 pour adapter le menu de navigation.

Tableau 9 : détails pour la proposition de menu de navigation

- b. « *récupération du menu défini* » (cf. détails dans Tableau 10) : elle permet de récupérer les données d’un menu sauvegardé.

Donnée en entrée	Nom du site web auquel le menu est associé.
Fonction	<ul style="list-style-type: none"> – récupération de l’identificateur du site web ; – récupération des données dans la table associée à l’identificateur.
Donnée en sortie	Menu existant.
Etape suivante	Etape 2 pour adapter le menu de navigation.

Tableau 10 : détails pour la récupération d’un menu sauvegardé

Etape 2 « adaptation du menu de navigation » (cf. détails dans Tableau 11) : elle permet d’adapter au site en cours de conception le menu proposé.

Donnée en entrée	Liste générée à l’étape 1a ou 1b.
Fonction	Adaptation de la liste à l’aide des fonctions suivantes : <ul style="list-style-type: none"> – sélection et modification éventuelle des items pour le site en conception ; – insertion d’items qui n’ont jamais été définis ; – définition de groupes d’items si nécessaire ; – définition de l’ordre d’affichage des groupes et des items dans un groupe ou dans le menu.
Donnée en sortie	Menu adapté, sauvegardé dans le modèle de site web.
Etape suivante	Etape 3 pour générer les données d’implémentation en fonction des choix de conception.

Tableau 11 : détails pour l’adaptation du menu de navigation

Etape 3 « génération des données associées » (cf. détails dans Tableau 12) : elle permet de générer automatiquement les données d’implémentation en fonction des choix de conception du menu.

Donnée en entrée	Menu adapté.
Fonction	<ul style="list-style-type: none"> – génération de la structure de données d’implémentation du menu ; – génération de l’architecture d’accueil ; – génération des fichiers pour chacun des items auxquels des données sont associées
Données en sortie	<ul style="list-style-type: none"> – structure de données du menu générée en base de données ; – architecture d’accueil générée ; – fichiers des items de menus créés dans l’architecture d’accueil du site web.
Etape suivante	Aucune (fin du processus).

Tableau 12 : détails pour générer des données associées au menu de navigation

1.3.2 Patron « présentation »

Le processus défini par ce patron est présenté Figure 48.

Si le modèle de présentation des pages du site en cours de conception n’est pas encore défini, le processus débute par l’opération 1a ou 1b ; si le modèle est déjà défini, le processus débute par l’étape 1c.

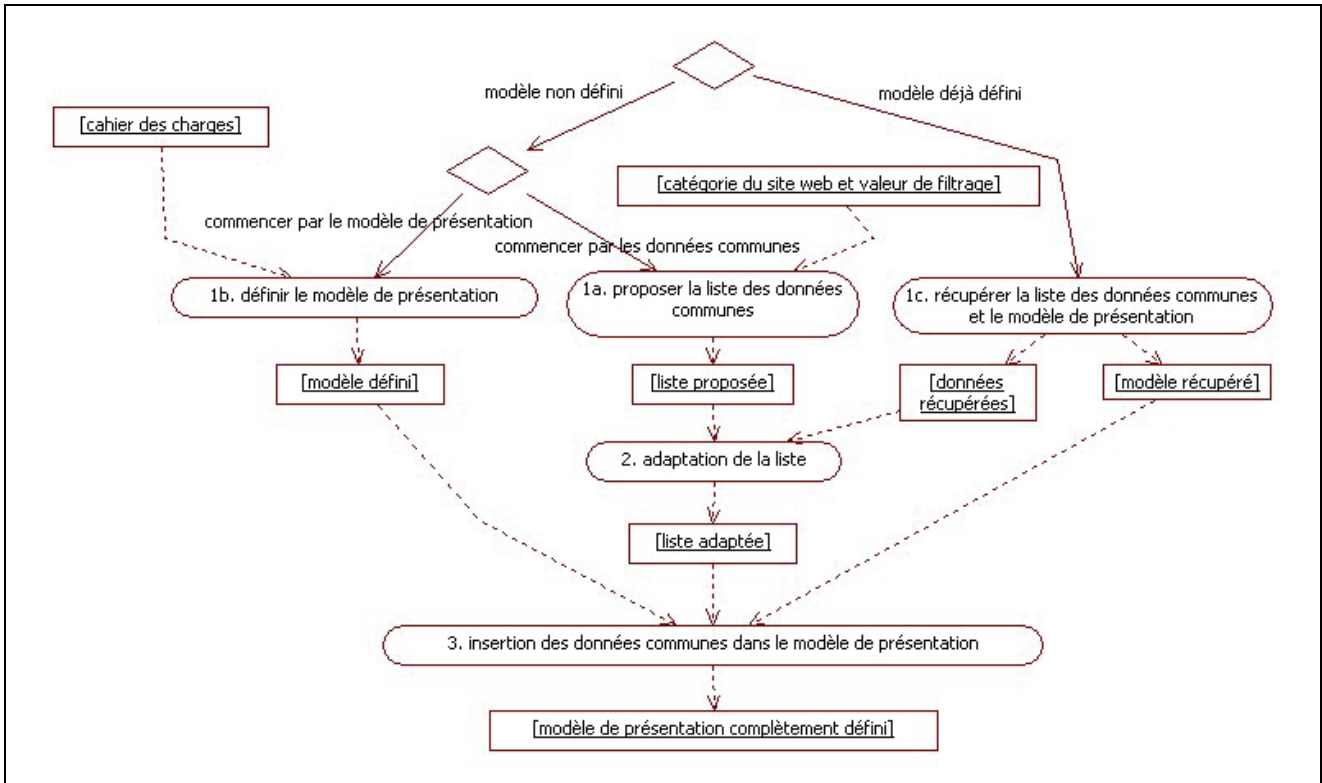


Figure 48 : processus du patron « présentation »

Etape 1

- a. « proposer la liste des données communes » (cf. détails dans Tableau 13) : elle propose la liste des informations communes à toutes les pages.

Données en entrée	Catégorie du site web en cours de conception et valeur de filtrage.
Fonction	<ul style="list-style-type: none"> – construction de la liste des informations communes à partir de l’analyse des pages des sites web selon la dimension « présentation » ; – calcul de l’indice de réutilisation de chacun des items de la liste.
Donnée en sortie	Liste des informations communes dont les indices de réutilisation sont supérieurs à la valeur de filtrage, ordonnée de manière décroissante selon les indices de réutilisation.
Etape suivante	Etape 2 pour adapter la liste des propositions.

Tableau 13 : détails pour proposer des données communes

- b. « définir le modèle de présentation » (cf. détails dans Tableau 14) : elle permet de définir le modèle de présentation des pages.

Donnée en entrée	Cahier des charges sur le type d'interface utilisateur (Flash, web 2.0, HTML standard, graphique interactif, etc.).
Fonction	Définir le modèle de présentation des pages en définissant des zones d'affichage à l'aide d'un éditeur WYSIWYG. Pour faciliter la définition de ce modèle, utiliser les types suivants de zones proposés (cf. Figure 49) : <ul style="list-style-type: none"> – « zone commune » pour afficher les informations communes à toutes les pages, – « zone accueil » pour afficher un contenu spécifique à la page d'accueil, – « zone navigation » pour afficher le menu défini à l'aide du patron « navigation » (la définition de cette zone peut utiliser les standards <i>de facto</i> de la conception des sites web⁴⁰), – « zone liens » pour afficher un ou plusieurs liens pour accéder rapidement à une page interne, – « zone annonces » pour afficher un rappel d'événements à venir si le site web contient ce type d'information ou pour afficher les modifications du site web ; – « zone contenu » pour afficher le contenu associé à un item du menu. Cette zone est généralement située au centre de la page.
Donnée en sortie	Modèle de présentation défini dans un fichier.
Etape suivante	Etape 3 pour insérer les données communes à toutes les pages dans le modèle.

Tableau 14 : détails pour définir le modèle de présentation

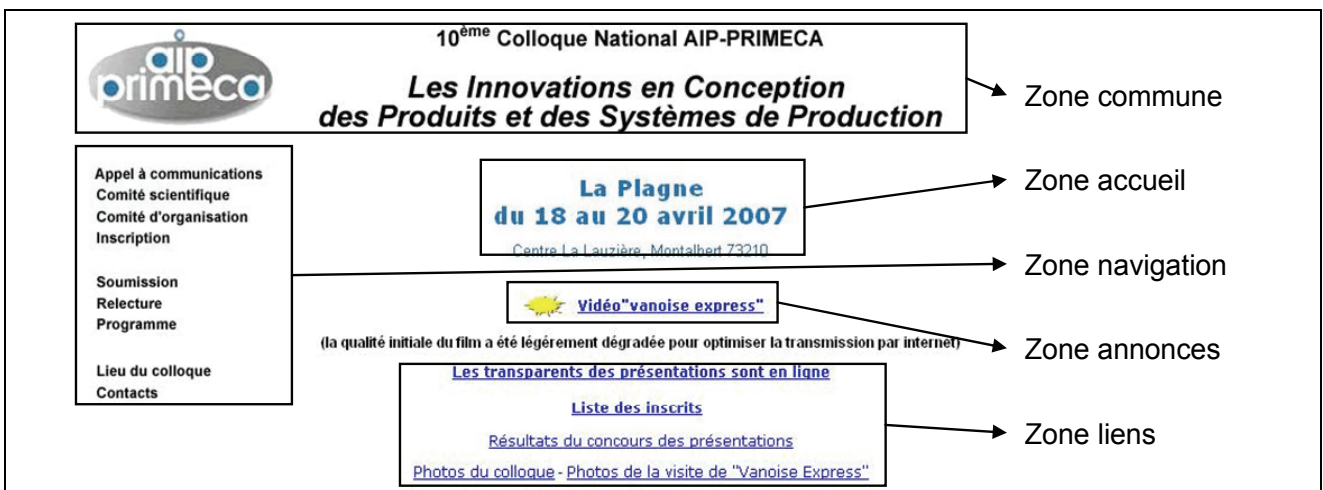


Figure 49 : exemples de zones sur une page d'accueil (colloque AIP-PRINECA)

c. « récupérer la liste des données communes et le modèle de présentation » (cf. détails dans Tableau 15) : elle permet de récupérer les données de la liste adaptée et le modèle de présentation sauvegardé.

Donnée en entrée	Nom du site web auquel les données sont associées.
Fonction	Les données et le modèle sont récupérés dans la table associée au site web.
Donnée en sortie	Données et modèle récupérés.
Etape suivante	Selon les cas : <ul style="list-style-type: none"> – étape 2 pour adapter la liste des propositions ; – étape 3 pour insérer les données communes dans le modèle de présentation s'il a été modifié.

Tableau 15 : détails pour récupérer la liste des données communes et le modèle de présentation

Etape 2 « adaptation de la liste » (cf. détails dans Tableau 16) : elle permet d'adapter au site en cours de conception la liste des données communes à toutes les pages.

⁴⁰ Par exemple : <http://www.cs.vu.nl/~martijn/patterns/index.html> et http://www.mit.edu/~jtjwell/common_ground.html

Donnée en entrée	Liste proposée des données communes (si l'étape précédente était l'étape 1a), ou liste existante des données communes (si l'étape précédente était l'étape 1c).
Fonction	Adaptation de la liste à l'aide des fonctions suivantes : – sélection et modification éventuelle des items qui correspondent au site en cours de conception ; – insertion d'items qui n'ont jamais été définis.
Donnée en sortie	Liste adaptée.
Etape suivante	Etape 3 pour insérer les données communes à toutes les pages dans le modèle.

Tableau 16 : détails pour l'adaptation de la liste

Etape 3 « insertion des données communes dans le modèle de présentation » (cf. détails dans Tableau 17) : elle permet de finaliser le modèle de présentation de toutes les pages web du site.

Données en entrée	Liste des données communes et modèle de présentation.
Fonction	Les données communes sont insérées à l'aide de l'éditeur WYSIWYG utilisé pour concevoir le modèle de présentation.
Donnée en sortie	Modèle de présentation finalisé.
Etape suivante	Aucune (fin du processus).

Tableau 17 : détails pour l'insertion de données communes

1.3.3 Patron « contenu »

Selon les cas, le contenu associé à un item est un ensemble de données, un lien externe ou un service. Le processus est présenté Figure 50.

Le point d'entrée dans le processus est l'item auquel des données doivent être associées. Si des données n'y sont pas encore associées, le processus débute à l'étape 1a ; si des données y sont déjà associées, le processus débute à l'étape 1b.

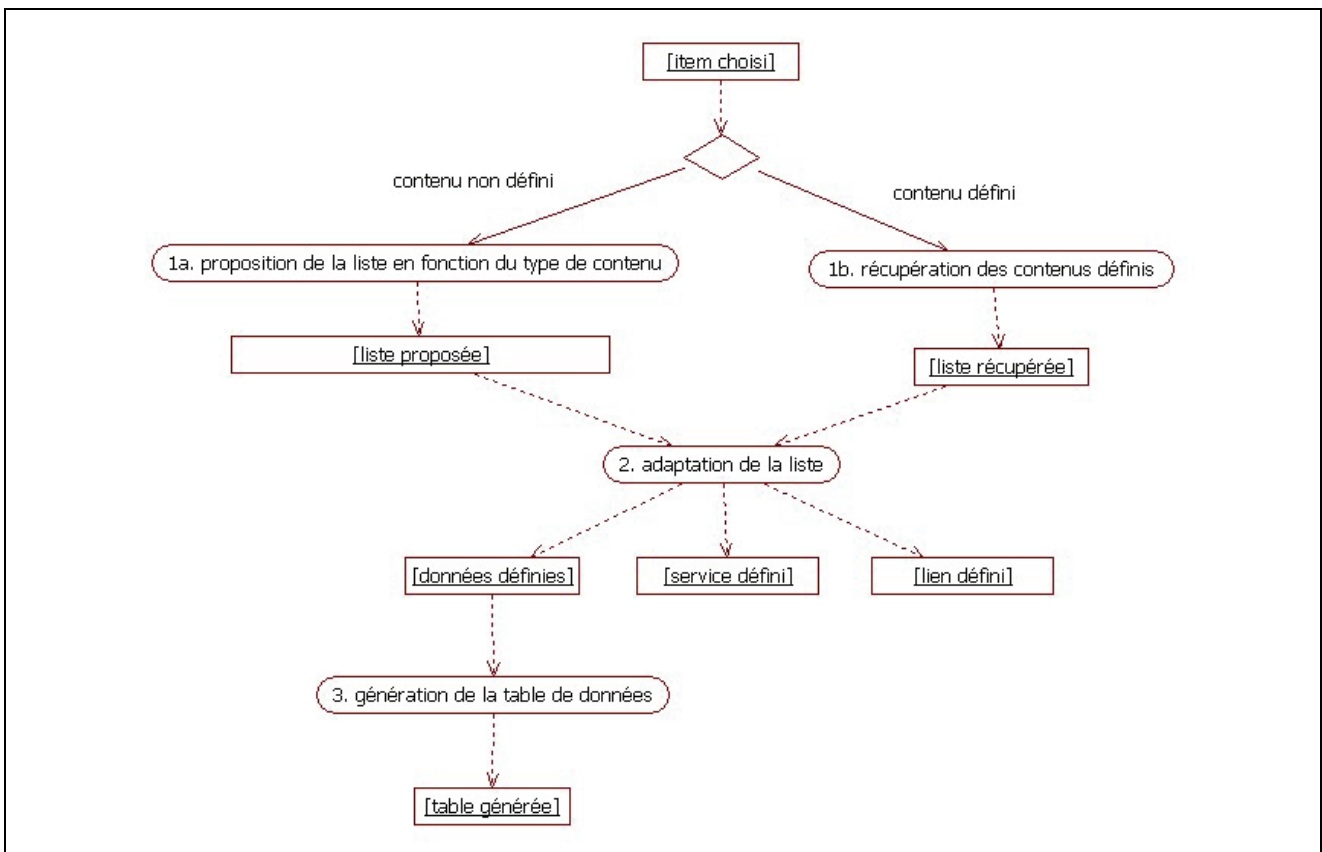


Figure 50 : processus du patron « contenu »

Etape 1

- a. « **proposition de la liste des contenus associés** » (cf. détails dans Tableau 18) : elle permet de proposer la liste en fonction du type de contenu choisi.

Donnée en entrée	Nom de l'item concerné par l'étape.
Fonction	<ul style="list-style-type: none"> – construction de la liste en fonction du type de contenu et à partir de l'analyse des pages des sites web selon la dimension « contenu » ; – calcul de l'indice de réutilisation de chacun des items de la liste.
Donnée en sortie	Liste des contenus associés à cet item dont les indices de réutilisation sont supérieurs à la valeur de filtrage, ordonnée de manière décroissante selon les indices de réutilisation.
Etape suivante	Etape 2 pour adapter la liste des propositions.

Tableau 18 : détails pour la proposition de la liste des contenus associés

- b. « **récupération des contenus définis** » (cf. détails dans Tableau 19) : elle permet de récupérer la liste des contenus déjà associés à l'item.

Donnée en entrée	Nom de l'item concerné par l'étape.
Fonction	Récupération des contenus dans le modèle de site web.
Donnée en sortie	Contenus récupérés.
Etape suivante	Etape 2 pour adapter la liste des propositions.

Tableau 19 : détails pour la récupération des contenus définis

Etape 2 « adaptation de la liste » (cf. détails dans Tableau 20) : elle permet d'adapter au site en cours de conception la liste des contenus associés à l'item.

Donnée en entrée	Liste proposée des contenus de l'item (si l'étape précédente était l'étape 1a), ou liste existante des contenus déjà définis (si l'étape précédente était l'étape 1c).
Fonction	<p>Définir les contenus associés (données, service, lien). Dans le premier cas, les données peuvent être définies selon l'un des trois types suivants :</p> <ul style="list-style-type: none"> – donnée statique, qui nécessite d'identifier le fichier correspondant, – donnée dynamique, qui peut être (cf. Figure 51) : <ul style="list-style-type: none"> – une donnée principale si elle différencie les informations entre elles au sein d'une page, – une donnée secondaire si elle est affichée à partir d'une sélection d'une donnée principale et lui est complémentaire, – une donnée interne si elle n'est pas affichée mais est utile à la gestion de l'affichage des deux types précédents.
Donnée en sortie	Liste adaptée.
Etape suivante	Etape 3 pour générer les données en fonction des choix de conception.

Tableau 20 : détails pour l'adaptation de la liste

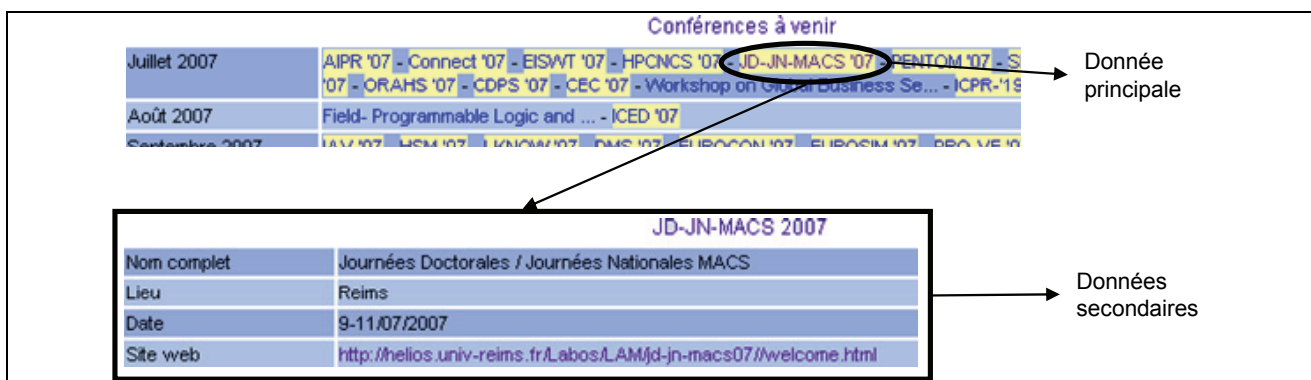


Figure 51 : exemple de types de données sur un extrait de la page « conférence » du GDR-MACS

Etape 3 « génération de la table de données » (cf. détails dans Tableau 21) : elle permet de générer la table de données en fonction des choix de conception de l'étape précédente.

Donnée en entrée	Liste adaptée.
Fonction	Génération de la table à partir de la liste adaptée, en se basant sur un schéma générique de table de données défini dans le patron.
Donnée en sortie	Table générée.
Etape suivante	Aucune (fin du processus).

Tableau 21 : détails pour la génération de la table de données

1.3.4 Patron « implémentation »

Le processus défini par ce patron est présenté Figure 52.

Le processus débute à l'étape 1.

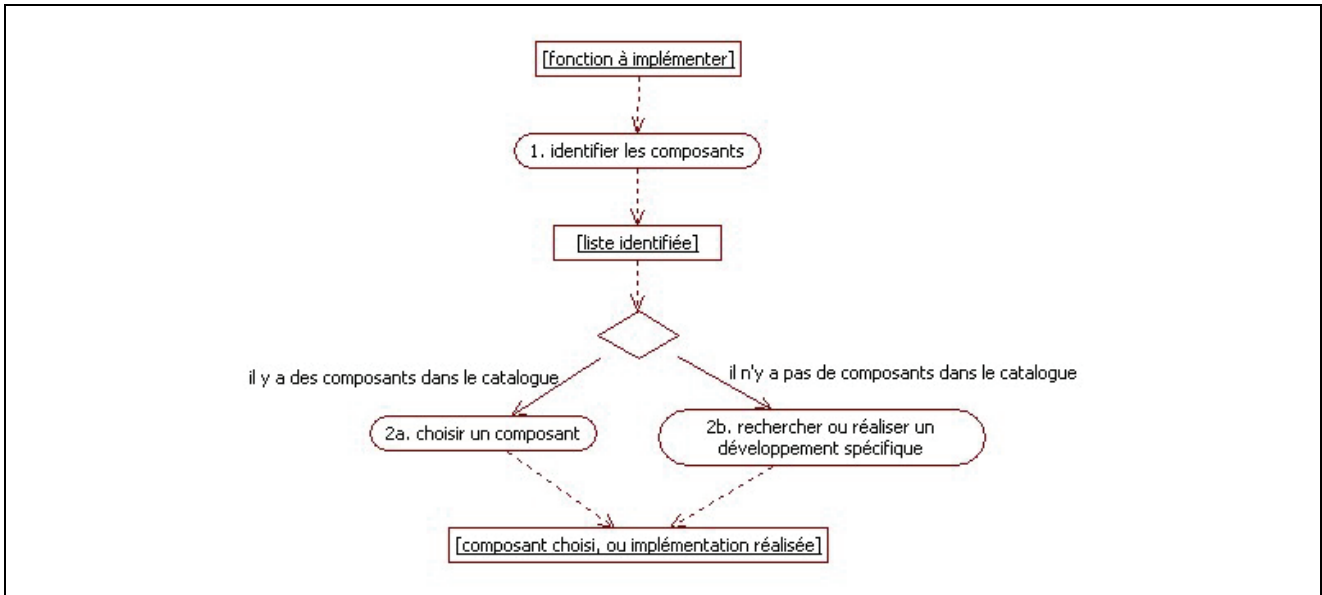


Figure 52 : processus du patron « identification du type d'implémentation »

Etape 1 « identifier les composants » (cf. détails dans Tableau 22) : elle permet d'identifier un composant à partir du catalogue en fonction du besoin à remplir (implémentation d'un menu, affichage de données, mise en œuvre d'un service).

Donnée en entrée	Fonction à implémenter (« affichage d'un menu », « affichage de données », service à implémenter).
Fonction	Identification des composants dans le catalogue à partir de la fonction « affichage d'un menu », de la fonction « affichage de données », ou de la fonction du service à implémenter.
Donnée en sortie	Résultat de la recherche.
Etape suivante	S'il y a des composants dans le catalogue, l'étape suivante est l'étape 2a ; s'il n'y a pas de composant dans le catalogue pour mettre en œuvre la fonction recherchée, l'étape suivante est l'étape 2b.

Tableau 22 : détails pour identifier les composants

Etape 2

- a. « choisir un composant » (cf. détails dans Tableau 23) : elle permet de choisir un composant parmi la liste proposée.

Donnée en entrée	Résultat de la recherche de l'étape précédente.
Fonction	Mise à disposition de possibilités interactives pour aider à choisir un composant à partir de leur caractérisation dans le catalogue.
Donnée en sortie	Composant choisi.
Etape suivante	Aucune (fin du processus).

Tableau 23 : détails pour choisir un composant

- b. « rechercher ou réaliser un développement spécifique » (cf. détails dans Tableau 24) : elle permet de trouver une solution pour palier l'absence de composant dans le catalogue.

Donnée en entrée	Aucune, car la recherche est à la charge du concepteur.
Fonction	Recherche d'un composant, laissé à l'initiative de l'utilisateur (moteur de recherche pour la recherche sur internet ; outil de modélisation et de développement pour le développement spécifique).
Donnée en sortie	Implémentation réalisée.
Étape suivante	Aucune (fin du processus).

Tableau 24 : détails pour rechercher ou réaliser un développement spécifique

2 Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »

Pour mettre en œuvre cette fonctionnalité, nous proposons dans cette partie [Cocquebert, 08b] :

- un modèle d'analyse de sites web en production pour identifier des solutions de conception ;
- un mécanisme d'exploitation des analyses réalisées pour faciliter la réutilisation des solutions identifiées.

2.1 Proposer des solutions de conception

2.1.1 Modèle de fiche d'analyse

Dans un premier temps, nous présentons Figure 53 le méta-modèle des fiches d'analyses.

Une fiche d'analyse est spécifique à une dimension de modélisation, indiquée par l'attribut « dimension ». Elle est composée d'un ensemble d'*items*, et chaque *item* est défini par une ou plusieurs *valeurs* fonction de l'item analysé.

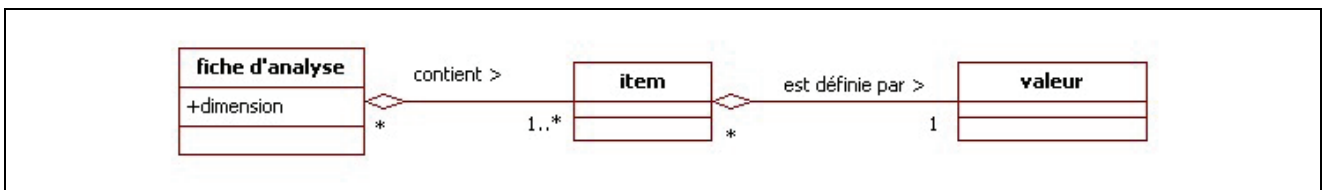


Figure 53 : modèle général de la fiche d'analyse

A partir de ce méta-modèle, nous présentons maintenant les modèles associés à chacune des dimensions de modélisations des sites web (« navigation », « présentation », « contenu » et « implémentation »), dont chacun est une instance du méta-modèle ci-dessus.

Dimension « navigation ». Comme représenté Figure 54, la *fiche d'analyse* de cette dimension est composée d'un ensemble d'*item de menu* dont chacun est caractérisé par :

- un label générique, permettant le référencement à différents sites web ;
- une définition, explicitant le label ;
- un attribut « présent » qui indique la présence d'un item dans le site analysé

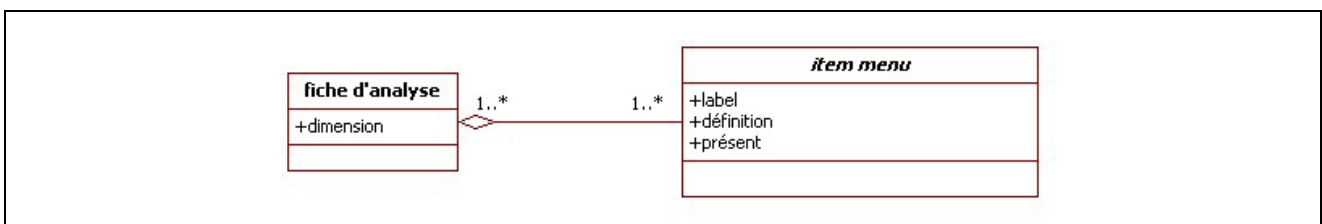


Figure 54 : modélisation de la fiche d'analyse de la dimension « navigation »

Dimension « présentation ». Comme représenté Figure 55, la *fiche d'analyse* de cette dimension est composée d'un ensemble d'*informations* dont chacune est caractérisée par :

- un label générique ;
- une définition explicitant le label ;
- un attribut « type d'organisation » qui indique si une information est présente sur toutes les pages ou si elle est spécifique à la page d'accueil.

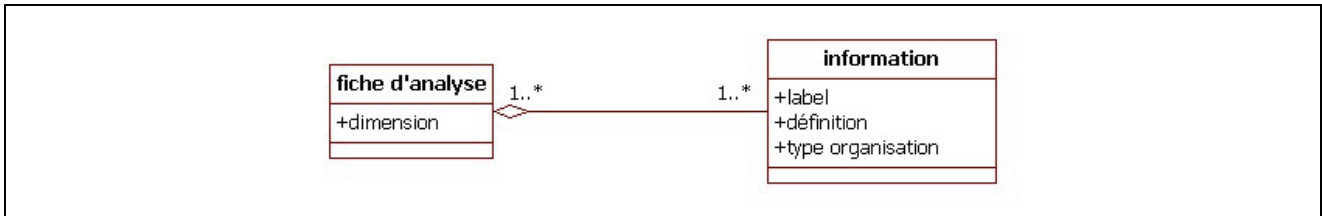


Figure 55 : modélisation de la fiche d'analyse de la dimension « présentation »

Dimension « contenu ». Comme représenté Figure 56, la *fiche d'analyse* de cette dimension est composée de plusieurs *item de menu*, à chacun desquels un ensemble de données peut être associé. Chaque *donnée* de cet ensemble est caractérisée par :

- un label générique ;
- une définition afin d'explicitier le label ;
- un attribut « nature » pour définir si la donnée est principale, secondaire ou interne.

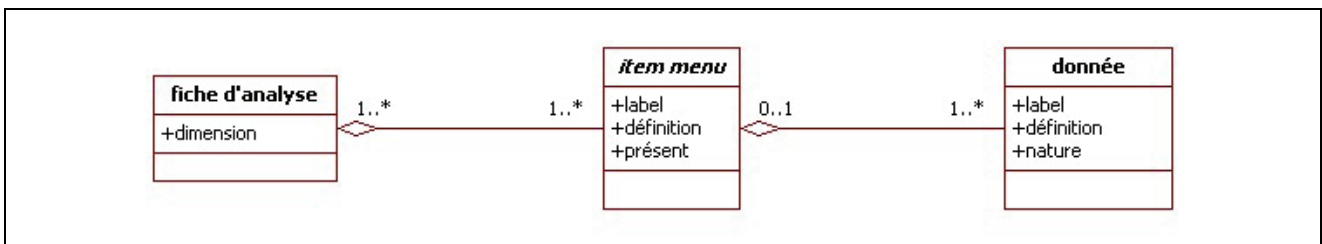


Figure 56 : modélisation de la fiche d'analyse de la dimension « contenu »

Dimension « implémentation ». Représenté Figure 57, une *implémentation* affiche des *données* ou met en œuvre un *service*, tous deux associés à *item de menu*.

Un *service* est défini par :

- une adresse internet (url) ;
- une définition explicitant le service rendu ;
- un attribut « présent » qui indique la présence d'un service dans le site analysé.

Une *implémentation* est définie par :

- un label générique ;
- une définition explicitant le service rendu.

Elle est réalisée par un composant.

Un *composant* est défini par :

- un label générique ;
- une référence sur sa caractérisation dans le catalogue d'analyse de composants.

Le contenu associé à un item de menu peut également être un *lien externe*, défini par une adresse internet (url) et un label pour l'explicitier.

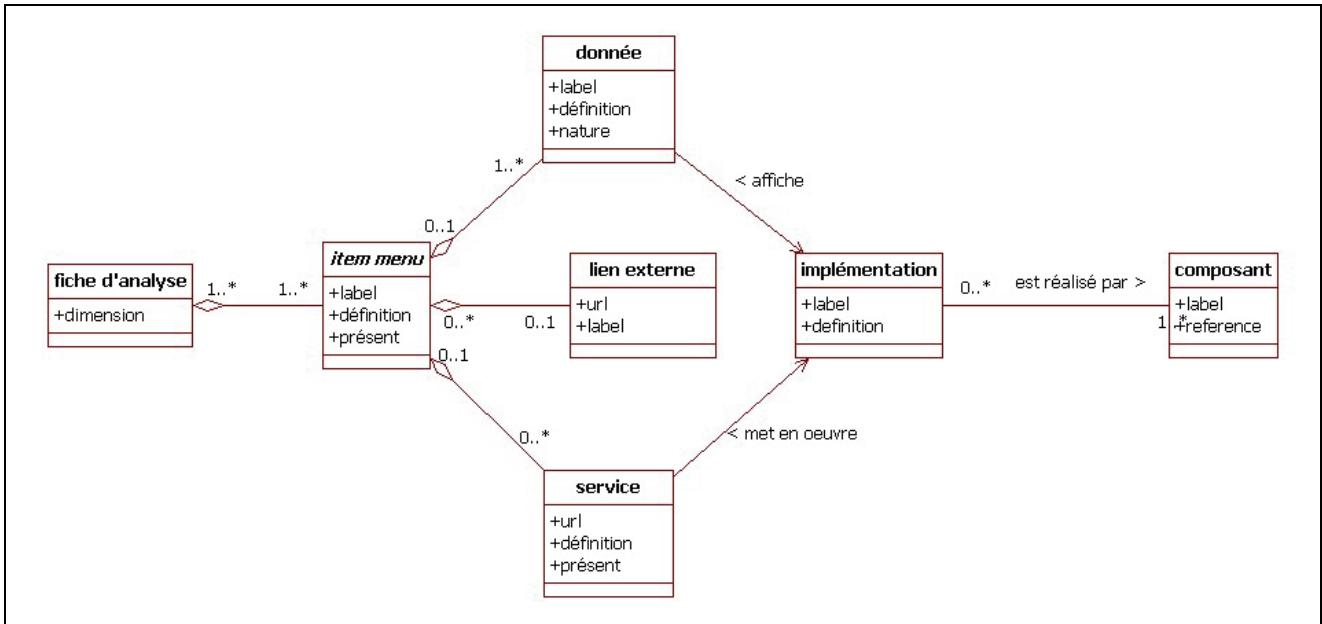


Figure 57 : modélisation de la fiche d’analyse de la dimension « service »

2.1.2 Utilisation des fiches d’analyses

Pour analyser un site web en production, il doit être référencé dans le catalogue. Comme illustré Figure 58, un *site web* en production est analysé à l’aide de *fiches d’analyse* spécifique à chaque dimension de modélisation, et il est référencé par les informations suivantes :

- un label qui correspond au nom du site web afin de l’identifier simplement ;
- une adresse internet (url) pour disposer de son adresse internet ;
- sa catégorie.

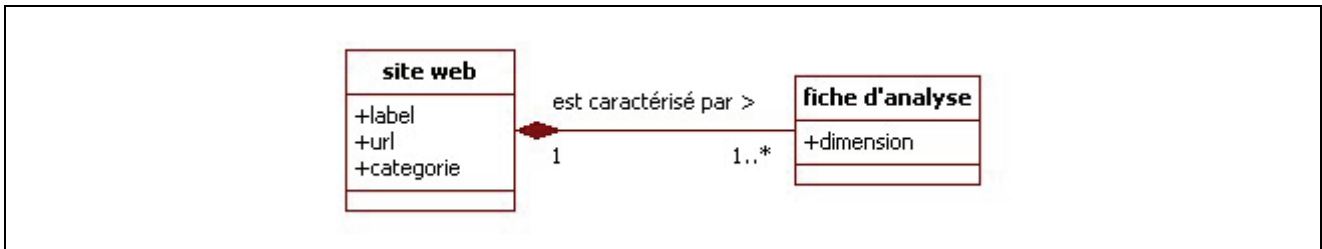


Figure 58 : modélisation de l’analyse d’un site web

Quand un site est référencé, il est analysé selon les quatre dimensions sans imposer d’ordre dans l’analyse des dimensions.

La réalisation d’une analyse consiste à indiquer la présence d’un item en sélectionnant dans la liste proposée le label dont la sémantique est la plus proche de l’item du site web. Par exemple, la Figure 59 présente un extrait de l’analyse de la navigation du site web, de l’entité L2TI⁴¹, dans laquelle l’item « membres » a été sélectionné car il était présent sur le site web.

⁴¹ <http://www-l2ti.univ-paris13.fr/>

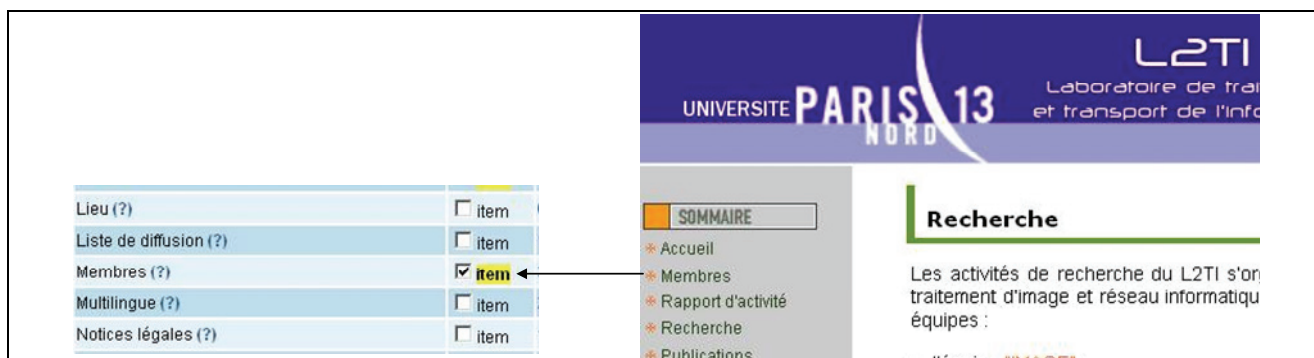


Figure 59 : extrait d'une analyse de la dimension « navigation » d'un site web

Quand un item présent sur le site web en production ne correspond à aucun des items génériques de la liste proposée, il faut le rajouter dans la liste associée à la dimension afin de le sélectionner ensuite dans l'analyse du site web. Dans le cas du site web L2TI, l'absence de l'item « publications » dans la liste proposée (Figure 60), a entraîné son insertion dans la liste afin de le sélectionner ensuite (Figure 61).

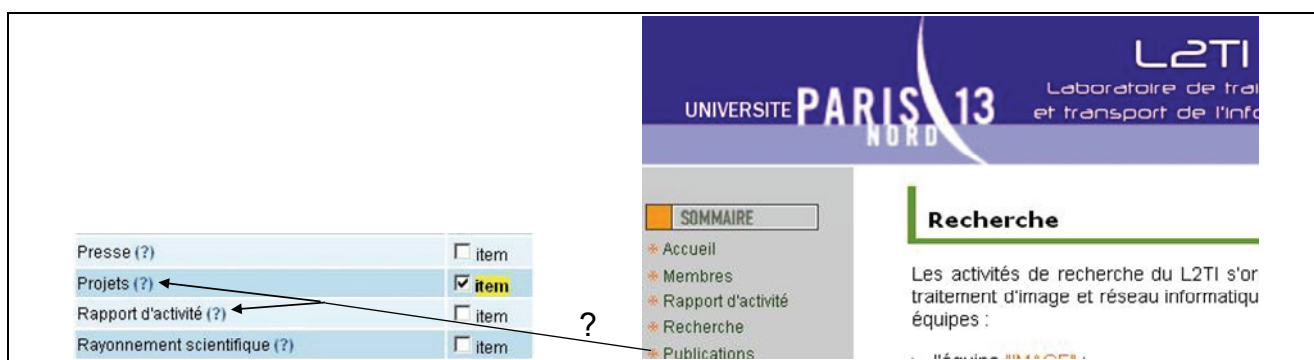


Figure 60 : besoin d'insertion d'un item dans la fiche d'analyse

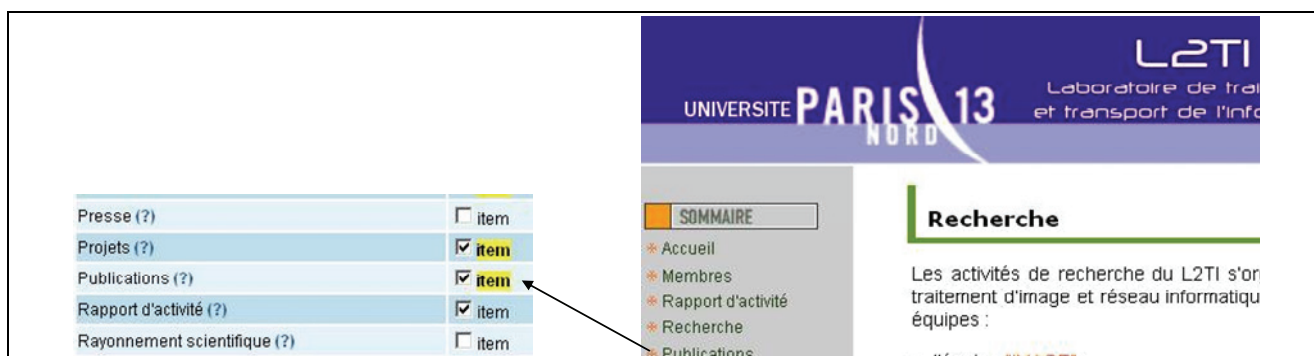


Figure 61 : fiche d'analyse modifiée

Le Tableau 25 synthétise l'ensemble des informations qui sont analysées et dont nous venons de décrire les modélisations.

	Navigation	Présentation	Contenu	Implémentation
Ce qui est analysé sur un site web et ce qui est proposé sous la forme d'une liste de labels génériques	Les items du menu	Les informations communes à toutes les pages du site web	Les données de définition d'un contenu associé à un item du menu	- Les types d'affichage des contenus - les services ou les liens externes associés aux items

Tableau 25 : résultat des analyses en fonction de la dimension de modélisation

2.1.3 Mécanisme d'exploitation des fiches d'analyses

Pour une dimension donnée, l'analyse d'un site web indique la présence des items de la liste pour ce site web. L'exploitation de l'ensemble des analyses consiste à :

- calculer pour chacun des items de cette liste leur indice de réutilisation, c'est-à-dire le nombre de fois que chaque item a été sélectionné dans les analyses ;
- ordonner ensuite la liste des items selon cette fréquence de réutilisation et la proposer à l'utilisateur.

Ainsi, l'utilisateur dispose d'une liste d'items de départ correspondant à la proposition de solutions de conception pour le site en cours de conception, avec l'indication de la fréquence d'utilisation.

Cependant, afin que la liste soit réaliste, les items qui la composent doivent avoir un taux de réutilisation significatif. Nous avons défini une « valeur de filtrage » minimale de réutilisation, calculée au prorata du nombre d'analyses de sites web. Ainsi, cela évite par exemple de proposer dans un premier temps des items qui n'ont été utilisés qu'une fois.

2.2 Faciliter la réutilisation des solutions de conception

Afin de faciliter la réutilisation de la liste issue de l'exploitation des analyses, c'est-à-dire d'adapter cette liste au site web en cours de conception, le concepteur a la possibilité d'utiliser les fonctions suivantes :

- sélection des items proposés, afin de notifier leurs réutilisations ;
- changement éventuel du label de l'item ;
- insertion de nouveaux labels non encore utilisés dans les sites web déjà analysés ;
- pour la dimension navigation, définition des groupes d'items et de l'ordre d'affichage des items dans le menu ou dans le groupe.

Appliquées à chacune des dimensions de modélisation, ces fiches d'analyses permettent d'obtenir l'arborescence, la liste des informations communes à toutes les pages, les données associées aux items de menus, les implémentations à mettre en œuvre (Tableau 26).

	Navigation	Présentation	Contenu	Implémentation
Ce qui est obtenu après adaptation	Menu de navigation	Liste des informations communes à toutes les pages	Données de définition d'items de menus	Liste des types d'implémentations des données, des services et des liens externes.

Tableau 26 : résultat de l'adaptation des propositions en fonction de la dimension de modélisations

3 Fonctionnalité « Identifier et faciliter le choix de composants logiciels »s

Pour mettre en œuvre cette fonctionnalité, nous proposons un modèle de composant pour identifier des composants à partir d'une recherche sur la fonction à remplir.

L'utilisation de l'approche « développement » nous a conduits à définir au sein de ce modèle les facettes fonctionnelles et non fonctionnelles attendues par le concepteur pour choisir un composant dans un catalogue. Comme représenté Figure 62, un *composant* est caractérisé par un ensemble de *rubriques*, dont chacune regroupe un ensemble de *facettes* définies par leur vocabulaire (espace de valeurs autorisées). Une *rubrique* est définie par :

- un label,
- une description explicitant le label,
- un point de vue afin de définir si elle est fonctionnelle ou non fonctionnelle.

Concernant les facettes, elles correspondent aux facettes attendues par les développeurs lors du choix d'un composant.

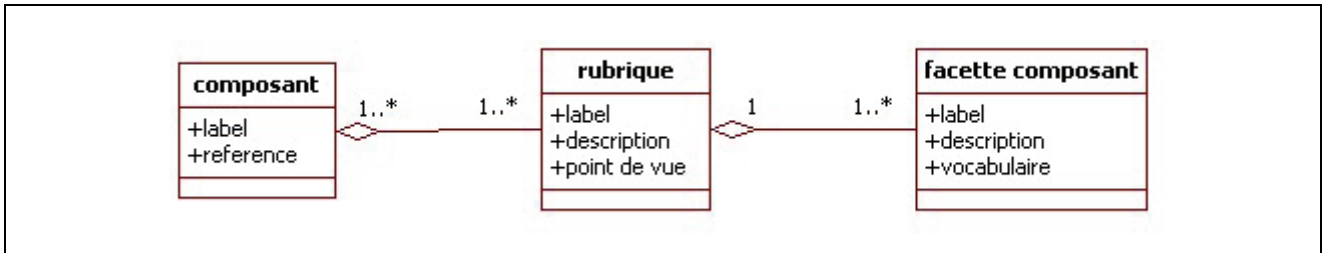


Figure 62 : modèle de composant

4 Fonctionnalité « Accélérer la mise en place d'un site web »

Pour mettre en œuvre cette fonctionnalité, nous proposons de générer automatiquement :

- des données à partir de choix de conception ;
- une architecture d'accueil prête à être personnalisée par le concepteur.

4.1 Génération de données

En reprenant le Tableau 25 p. 89, la génération de données ne concerne que les dimensions « navigation » et « contenu ». En effet, nous ne pouvons pas générer de données pour la dimension « présentation » car le résultat de cette étape est un modèle de présentation défini par un outil externe. De même, nous ne pouvons pas générer de données pour la dimension « implémentation » car le résultat de cette étape est le choix d'un composant ou d'un développement spécifique dont l'intégration dans l'architecture logicielle est à la charge du développeur.

4.1.1 Génération pour la navigation

Les données à générer concernent les structures de données pour implémenter l'arborescence et les fichiers associés aux items dans lesquels les méthodes de traitement des données doivent être implémentées.

Structure de données pour implémenter l'arborescence. La Figure 63 présente un diagramme de classe pour implémenter un menu. En application de la « règle des trois clicks⁴² », nous ne considérons qu'un seul niveau de profondeur⁴³.

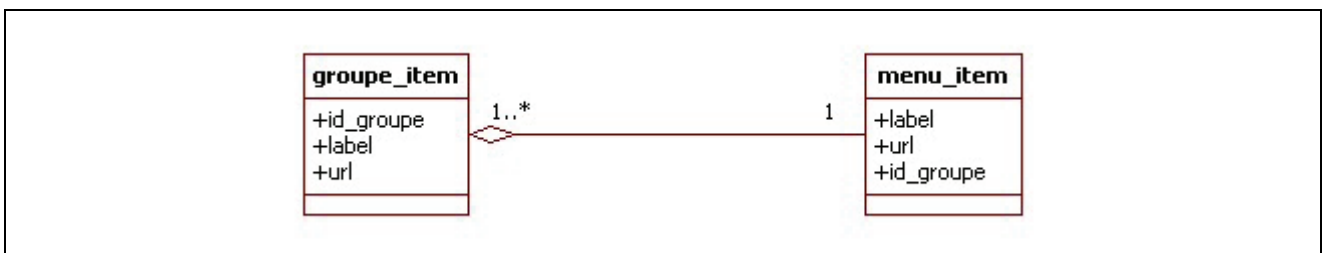


Figure 63 : diagramme de classe pour générer l'implémentation du menu

Un *groupe_item* est défini par :

- un identifiant ;
- un label ;
- son adresse internet (url).

Un *menu_item* est défini par :

- un label ;
- son adresse internet (url).

⁴² « Trois clicks maximum pour accéder à une information dans un site web ».

⁴³ 1^{er} click : groupe de menu pour connaître les items ; 2^{ème} click : item de menu pour afficher la page ; 3^{ème} click : sur un lien au sein de la page affichée si elle en dispose.

Il est attaché à son groupe par `id_groupe`.

Génération de fichiers. Les fichiers sont générés à partir du modèle de fichier qui réalise le lien avec l'architecture d'accueil (présentée paragraphe suivant) selon plusieurs points (Figure 64) :

- lien avec les variables globales qui caractérisent l'environnement d'exécution du site web telles que données d'identification pour la base de données, noms génériques de tables de données, variables globales, etc. ;
- lien avec les classes générales qui implémentent en natif les accès à la base de données, l'affichage de données dans des tableaux, la génération de graphiques, etc. ;
- lien avec les composants techniques et métiers qui sont intégrés à l'architecture ;
- lien avec l'environnement de débogage ;
- déclaration des prototypes des méthodes d'interaction avec la base de données, de préparation des données pour l'affichage et de l'affichage des données elles-mêmes.

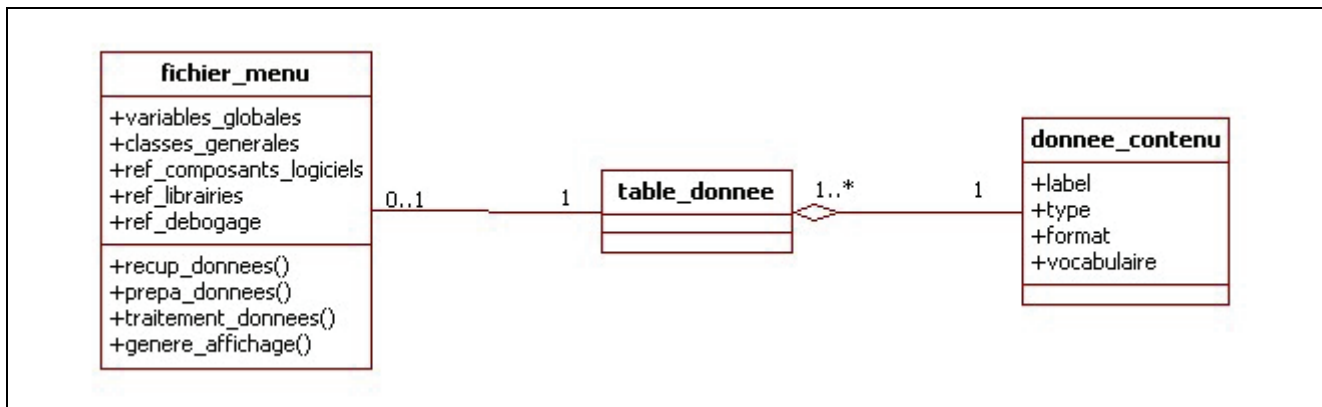


Figure 64 : méta-modèle pour la génération de données

Les classes *table_données* et *donnee_contenu* sont présentées dans la partie suivante.

4.1.2 Génération pour le contenu

La structure de données générée pour le contenu s'appuie sur l'identification des champs de définition réalisée lors de l'étape 6 (cf. Figure 42 p. 73). Elle est basée sur le méta-modèle de donnée présenté Figure 64. Dans ce méta-modèle, les champs de définition issus de l'étape vont être générés sous la forme de *donnees_contenu*, et correspondent aux champs de définition d'une *table_donnee*. La classe *donnee_contenu* est définie par :

- label : celui de l'item sélectionné par le concepteur, éventuellement modifié
- type : « principal », « secondaire » ou « interne »
- vocabulaire : espace de valeurs éventuel.

La liaison avec l'item de menu se fait par un lien entre *table_donnee* et *fichier_menu*, qui constitue le fichier de définition d'un item de menu.

4.2 Génération d'une architecture logicielle d'accueil

Afin de faciliter le développement d'un nouveau site web, l'architecture d'accueil générée est organisée selon un ensemble de dossiers contenant un ou plusieurs fichiers (Figure 65).

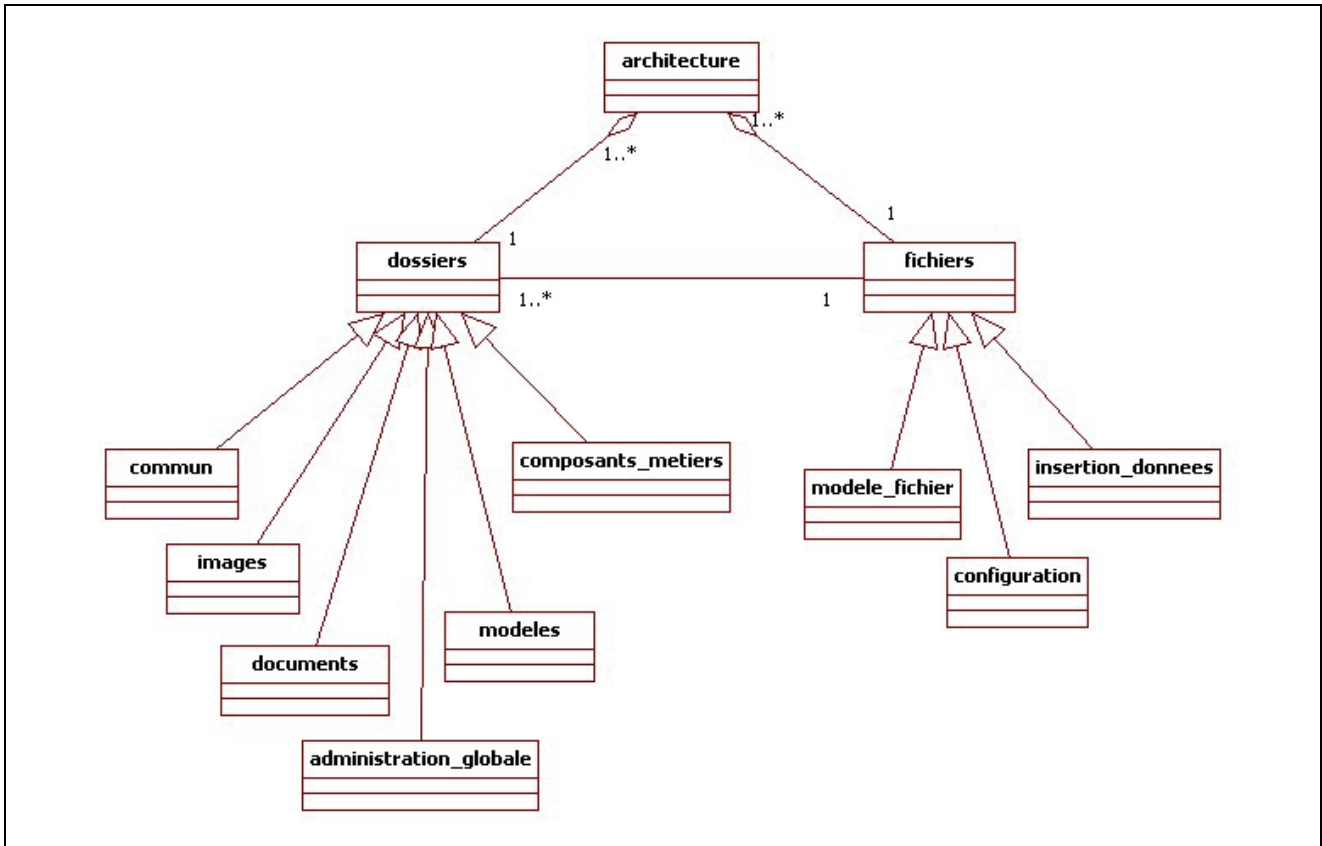


Figure 65 : modèle de l'architecture

Les dossiers peuvent concerner :

- le comportement de bas niveau de l'architecture (« commun ») ;
- les *images* ou les *documents* utilisés dans les fichiers des pages du site web ;
- l'*administration globale* de la base de données propre à l'architecture ou générée à partir des choix de conception ;
- les *modèles* à utiliser pour générer des données à partir des choix de conception ;
- les *composants métiers* intégrés de manière native.

Les fichiers peuvent définir :

- des modèles à utiliser pour générer les fichiers associés aux items de menu ou pour générer les structures de données ;
- les informations propres à la configuration de l'architecture elle-même ou à personnaliser pour le nouveau site web en cours de conception ;
- l'insertion de données dans les zones du modèle des pages web.

5 Conclusion

L'objectif de cette partie était de proposer les différents éléments constitutifs d'une méthode d'aide à la conception de site web qui s'inspirent de nos spécifications et que nous avons appelé WISDOM. Ces éléments, proposés pour la mise en œuvre de chaque fonctionnalité identifiée, sont résumés ci-dessous.

Fonctionnalité « Assister la réalisation du processus de conception ». Pour guider le processus de conception, nous avons proposé :

- un processus de conception ;
- des modèles pour les dimensions reconnues de modélisation d'un site web qui sont « navigation », « présentation », « contenu »

- la prise en compte explicite d'une quatrième dimension « implémentation », pour laquelle un modèle a également été proposé.

Pour assister la réalisation des étapes de ce processus, nous avons proposé des patrons de domaine pour les étapes du processus et nous avons détaillé le processus de chacun des patrons.

Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception ». Pour identifier des solutions de conception sans imposer une expertise dans l'approche patron, nous avons successivement proposé :

- un modèle d'analyse de site web afin de construire une fiche d'analyse de sites web pour chacune des dimensions de modélisation ;
- l'utilisation de cette fiche pour analyser un site web ;
- l'exploitation de ces analyses afin de proposer une liste de départ pour la conception d'un nouveau site web.

Pour faciliter la réutilisation de cette liste de départ, nous avons ensuite présenté les différentes fonctions de l'interface qui permettent l'adaptation de la liste proposée pour la conception d'un site web.

Fonctionnalité « Identifier et faciliter le choix de composants logiciels ». Pour identifier des composants logiciels, nous avons défini un modèle de composants basé sur différentes facettes fonctionnelles et non fonctionnelles et organisées selon les rubriques afin d'identifier des composants. Nous n'avons pas considéré les spécifications liées à la recherche multicritère pour faciliter le choix de composants logiciels car elles nécessitaient la capitalisation de règles métiers qui sortait du cadre de notre étude.

Fonctionnalité « Accélérer la mise en place d'un site web ». Nous avons d'abord proposé la génération de structure de données et de fichiers uniquement pour les dimensions de modélisation « navigation » et « contenu », les deux autres n'étant pas concernées de par la nature du résultat de leurs réalisations, et nous avons ensuite proposé le modèle d'une architecture logicielle d'accueil afin de faciliter le développement d'un nouveau site web.

L'objectif du chapitre suivant est de présenter une mise en œuvre possible de ces propositions au sein d'un outil que nous avons appelé « WISDOM Tool ».

Chapitre V : Mise en œuvre

Ce chapitre présente les mises en œuvre des éléments constitutifs de la méthode WISDOM présentés dans le chapitre IV. Elles constituent l'outil « WISDOM Tool » qui se présente sous la forme d'un site web⁴⁴ développé en utilisant :

- HTML pour l'affichage ;
- PHP pour l'interaction entre l'affichage et la base de données ;
- MySQL pour la base de données.

La page d'accueil du site web, présentée Figure 66, propose l'accès aux modules logiciels détaillés dans ce chapitre :

- « conception d'un site web » (partie 2) ;
- « catalogue d'analyses de sites web » (partie 3) ;
- « catalogue d'analyses de composants » (partie 4) ;



Figure 66 : page d'accueil du WISDOM Tool

1 Conception d'un site web

Cette partie présente la mise en œuvre du processus de conception d'un site web, incluant l'utilisation des patrons de domaine qui ont été formalisés [Cocquebert, 08a] [Cocquebert, 08b]. Le processus implémenté est celui représenté Figure 42 page 73.

D'une manière générale, l'interface associée à chacune des étapes est composée des informations suivantes :

- le nom de l'étape en cours ;
- une interface spécifique à l'étape ;
- un ou plusieurs liens pour revenir vers des étapes précédentes ;
- un lien permettant de revenir à la page d'accueil.

1.1 Etape 1 : « définition du contexte »

Le contexte est défini par l'identification du type de site web à concevoir et par un calcul automatique de la valeur de filtrage, fonction du nombre d'analyses de sites web de même type qui ont été réalisées.

En reprenant les deux catégories de sites web considérées dans cette étude, les types de sites web proposés sont les suivants (Figure 67) :

- « présentation entité », « présentation équipes » et « groupe » dont les sites web se caractérisent par la présentation des entités avec diffusion centralisée de documents (« sites catalogues ») ;
- « événement » dont les sites web se caractérisent par la mise à disposition des informations propres à l'organisation d'une conférence (« sites de présence »).

⁴⁴ <http://www.univ-valenciennes.fr/sp/cocquebert/wisdom/index.php>

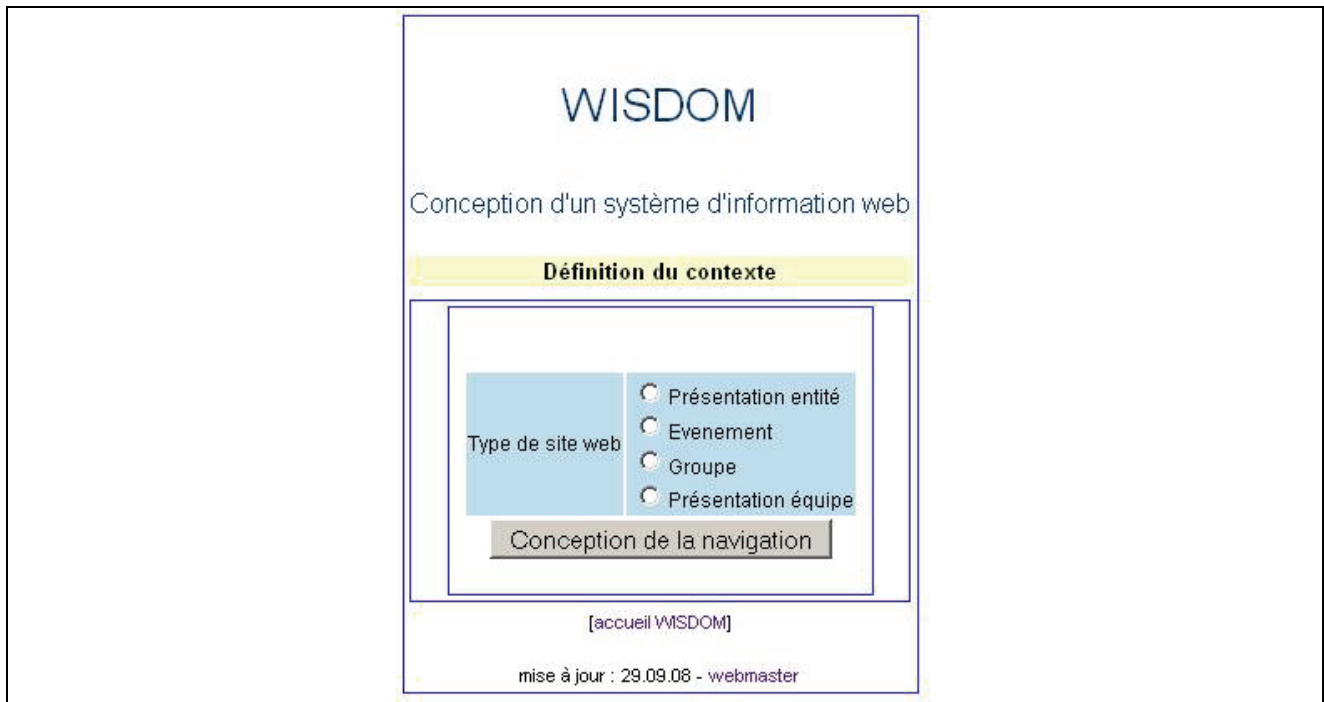


Figure 67 : étape « identification du contexte »

Quand le type de site web est défini, l'interface oriente l'utilisateur vers l'étape suivante, c'est-à-dire l'étape 2 « conception de la navigation » du processus de conception.

1.2 Etape 2 : « conception de la navigation »

Cette étape vise à concevoir (ou modifier) le menu en adaptant les propositions générées à partir des analyses de sites web selon la dimension « navigation ». Basée sur le cahier des charges, elle est réalisée par le concepteur à l'aide de l'interface présentée Figure 68, dont nous présentons maintenant les différentes parties.

- **Partie 1** : elle affiche les informations suivantes :

- nom de l'étape en cours ;
- lien « aide » pour accéder au patron associé ;
- le type de site web en cours de conception (dans la figure : le type est « présentation entité »).

- **Partie 2** : elle propose une liste d'items pour le menu en tenant compte de la valeur de filtrage définie automatiquement à la sortie de l'étape précédente « définition du contexte ». Ces items sont classés par ordre décroissant de leur indice de réutilisation, lequel est indiqué entre parenthèses après le label de l'item. Si la liste est trop longue ou trop courte, le concepteur peut modifier la valeur de filtrage afin d'avoir plus ou moins de propositions d'items. L'adaptation de la liste pour concevoir la navigation consiste à :

- cocher les items qui sont satisfaisants pour le site web en cours de conception ;
- cliquer sur le bouton « Retenir » pour afficher les items retenus dans la partie 3 afin de modifier leur définition.

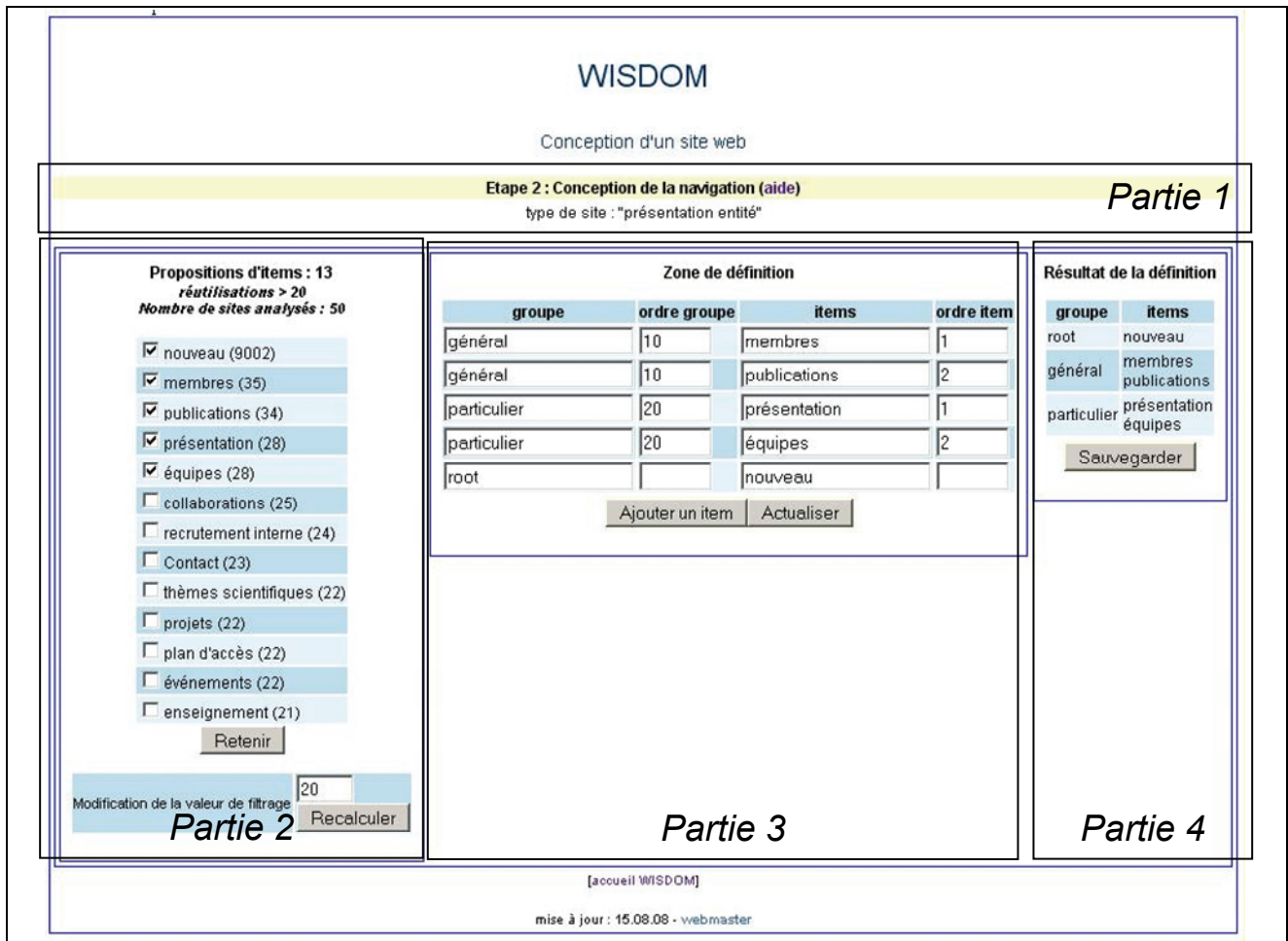


Figure 68 : conception de la navigation

- **Partie 3** : elle présente le menu en cours de conception à l'aide de quatre colonnes :
 - « groupe » : elle définit les différents groupes d'items du menu. Par défaut, un groupe est appelé « root », mais le concepteur peut choisir le nom (dans la figure, le concepteur a défini les groupes « général » et « particulier ») ;
 - « ordre groupe » : elle définit l'ordre d'affichage des groupes dans les pages du site web. Si le concepteur n'utilise pas cette possibilité, les groupes seront affichés par ordre alphabétique ;
 - « items » : elle définit le nom des items de menu en reprenant ceux qui ont été choisis parmi les propositions sélectionnées dans la partie 2 de l'interface (cf. ci-dessus), avec la possibilité de le modifier par le concepteur. Si celui-ci a utilisé le bouton « ajouter un item », le nouvel item est appelé « nouveau » par défaut et il est inséré en haut de la liste de propositions de la partie 2 ;
 - « ordre item » ; elle définit l'ordre d'affichage des items à l'intérieur de leur groupe. Si le concepteur n'utilise pas cette possibilité, les items seront affichés par ordre alphabétique.

Le bouton « Actualiser » permet de visualiser le menu dans la partie 4 de l'interface à partir des items définis dans cette partie 3 de l'interface.

- **Partie 4** : elle affiche le menu en cours de définition à l'aide de deux colonnes :
 - « groupe » : elle indique le nom des groupes en respectant l'ordre d'affichage ;
 - « items » : elle indique le nom des items de chaque groupe en respectant l'ordre d'affichage.

Le bouton « Sauvegarder » permet de :

- mémoriser les informations dans le modèle de conception du site web ;
- générer les tables de définition du menu ;
- générer l'architecture d'accueil et les fichiers associés aux items du menu ;
- commander l'affichage de l'interface associée à l'étape suivante « conception de la présentation ».

1.3 Etape 3 : « conception de la présentation »

Cette étape vise à concevoir (ou modifier) le modèle de présentation de toutes les pages du site web en réalisant deux tâches dont l'ordre d'exécution n'est pas imposé :

- identification des informations communes à toutes les pages ;
- conception du modèle de présentation de toutes les pages.

Elle est basée sur le cahier des charges et exploite les analyses de sites web selon la dimension « présentation ».

1.3.1 Identification des informations communes.

Cette tâche est réalisée en utilisant une interface dont les fonctions sont similaires à celle qui a été détaillée dans l'étape précédente, mais dont le contenu utilise les analyses selon la dimension « présentation ». Ce contenu est adapté de la façon suivante (Figure 69).

- **Partie 2** : elle propose une liste d'informations communes ;
- **Partie 3** : elle permet de définir la position « haut » ou « bas » des items retenus dans la partie 2 ou insérés à l'aide du bouton « ajouter un item ».
- **Partie 4** : elle récapitule par type de position les informations retenues ;
- **Partie 5** : elle contient un lien « modifier la navigation » pour revenir à l'interface de l'étape 2.

WISDOM
Conception d'un site web

Etape 3 : Conception de la présentation (aide)
type de site : "présentation entité"

<p>Propositions d'items : 5 <i>réutilisations > 20</i> Nombre de sites analysés : 50</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> entités d'attache (47) <input checked="" type="checkbox"/> nom complet (44) <input type="checkbox"/> présentation (29) <input checked="" type="checkbox"/> multilingue (26) <input checked="" type="checkbox"/> coordonnées postales (24) <p style="text-align: center;"><input type="button" value="Retenir"/></p> <p>Modification de la valeur de filtrage <input style="width: 50px;" type="text" value="20"/> <input type="button" value="Recalculer"/></p> <p style="text-align: center;">Partie 2</p>	<p style="text-align: center;">Zone de définition</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">items</th> <th style="text-align: left;">position</th> </tr> </thead> <tbody> <tr> <td>entités d'attache</td> <td>haut ▼</td> </tr> <tr> <td>nom complet</td> <td>haut ▼</td> </tr> <tr> <td>multilingue</td> <td>haut ▼</td> </tr> <tr> <td>coordonnées postales</td> <td>bas ▼</td> </tr> </tbody> </table> <p style="text-align: center;"><input type="button" value="Ajouter un item"/> <input type="button" value="Actualiser"/></p> <p style="text-align: center;">Partie 3</p>	items	position	entités d'attache	haut ▼	nom complet	haut ▼	multilingue	haut ▼	coordonnées postales	bas ▼	<p style="text-align: center;">Résultat de la définition</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">position</th> <th style="text-align: left;">items</th> </tr> </thead> <tbody> <tr> <td>bas</td> <td>coordonnées postales</td> </tr> <tr> <td>haut</td> <td>entités d'attache multilingue nom complet</td> </tr> </tbody> </table> <p style="text-align: center;"><input type="button" value="Sauvegarder"/></p> <p style="text-align: center;">Partie 4</p>	position	items	bas	coordonnées postales	haut	entités d'attache multilingue nom complet
items	position																	
entités d'attache	haut ▼																	
nom complet	haut ▼																	
multilingue	haut ▼																	
coordonnées postales	bas ▼																	
position	items																	
bas	coordonnées postales																	
haut	entités d'attache multilingue nom complet																	

[accueil WISDOM] [modifier la navigation]

mise à jour : 15.08.08 - webmaster

Partie 5

Figure 69 : conception de la présentation

Le bouton « Sauvegarder » permet de :

- mémoriser les informations dans le modèle de conception du site web ;
- commander l'affichage de l'interface associée à l'étape suivante « définition de l'implémentation ».

1.3.2 Conception du modèle de présentation.

La conception d'un modèle de présentation est réalisée à l'aide d'un éditeur WYSIWYG type Dremaweaver™. Elle est laissée à la charge du concepteur, mais le patron associé à cette étape présente différents types de zones afin de rationaliser la conception de ce modèle.

Afin de laisser le libre choix de l'outil utilisé pour concevoir le modèle graphique de présentation, il n'y a pas eu d'intégration logicielle entre WISDOM Tool et un éditeur WYSIWYG type Dremaweaver™.

Quand le modèle est défini, le concepteur doit intégrer lui-même les informations communes dans ce modèle de présentation pour en terminer la conception.

1.4 Etape 4 : « définition de l'implémentation »

Cette étape fait suite à l'étape 3 ou à l'étape 6. Dans les deux cas, elle est réalisée en utilisant une interface dont les fonctions sont similaires à celle qui a déjà été détaillée, mais dont le contenu utilise les analyses selon la dimension « implémentation ». Ce contenu est adapté de la façon suivante (Figure 70).

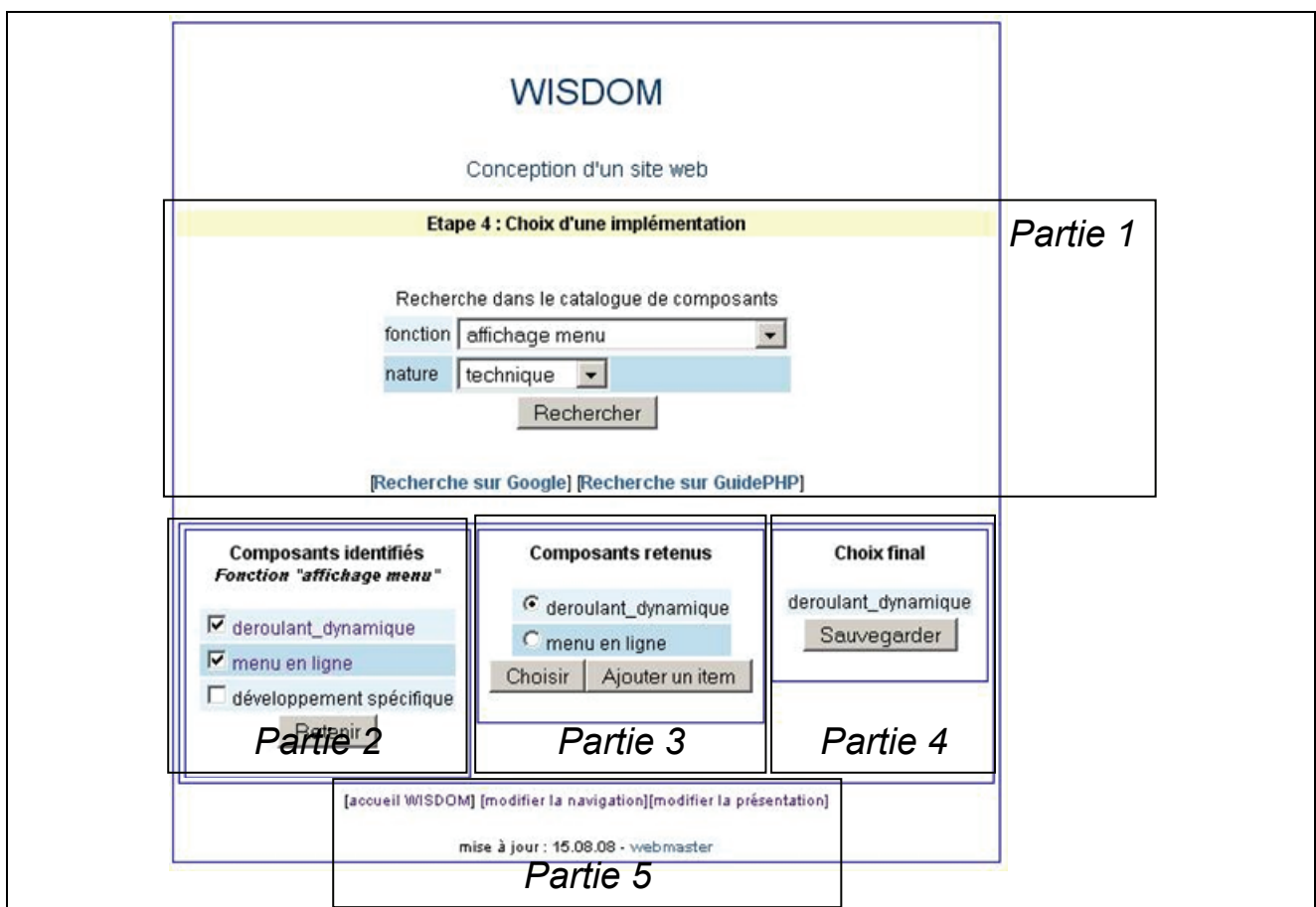


Figure 70 : choix d'une implémentation pour le menu

- **Partie 1** : elle affiche les informations suivantes :

- le nom de l'étape ;
- l'interface de recherche dans le « catalogue de composants » de WISDOM Tool ;
- des liens de recherche sur internet (« Google™ », « GuidePHP⁴⁵ ») pour que le concepteur complète les propositions de la partie 2 s'il le désire.

⁴⁵ Site web pour aider à choisir des composants (<http://www.guidephp.com/>), élaboré dans le cadre du groupe de travail « briques logicielles » de l'AFUP auquel je contribue et dont nous présentons des résultats § 13 page 111.

- **Partie 2** : Elle affiche une liste résultat de la recherche en fonction de l'étape précédente :
 - si l'étape précédente est l'étape 3 : la recherche se fait sur les composants techniques qui implémentent la fonction « affichage menu » (cf. Figure 70) ;
 - si l'étape précédente est l'étape 6 : la recherche dépend du type de contenu associé :
 - type « données » : l'outil cherche des composants techniques qui implémentent la fonction « afficher_données »,
 - type « service » : l'outil cherche des composants techniques ou métiers qui implémentent la fonction du service qui a été définie lors de l'étape 6.

Cette liste résultat de la recherche contient au moins le choix « développement spécifique » si les propositions ne conviennent pas au concepteur ou s'il n'y a pas de propositions.

Chaque proposition est présentée sous la forme d'un lien hypertexte afin de consulter la fiche de renseignements associé dans le catalogue de composants⁴⁶.

- **Partie 3** : Elle affiche la liste des composants retenus dans la partie 2 et donne la possibilité d'ajouter des items (bouton « ajouter un item »). En l'absence d'aide multicritère au choix, le choix est réalisé par le concepteur à partir de la consultation des fiches de renseignements disponibles dans le catalogue de composants.

- **Partie 4** : elle affiche le choix final du concepteur et le bouton « sauvegarder ». Ce dernier permet de :
 - mémoriser l'information dans le modèle de conception du site web ;
 - commander l'affichage de l'interface associée à l'étape suivante « choisir un item de menu » ou d'indiquer la fin du processus si tous les items du menu ont été traités.

Partie 5 : elle contient des liens vers les étapes précédentes.

1.5 Etape 5 : « choisir un item de menu »

Cette étape vise à :

- choisir l'item auquel un contenu va être associé ;
- définir le type de contenu ;
- suivre l'évolution de la conception.

Elle utilise également l'interface des étapes précédentes, dont le contenu est adapté de la façon suivante (Figure 71).

- **Partie 1** : elle rappelle les items du menu définis lors de l'étape 3 « conception de la navigation », triés en respectant l'ordre d'affichage des groupes et l'ordre d'affichage des items dans les groupes. Pour les items qui n'ont pas été traités, l'interface propose les trois types de données possibles (« données », « lien » ou « service ») ; pour les autres :
 - elle indique que la définition est « ok » en indiquant la nature du contenu qui a été défini ;
 - elle propose un lien permettant de modifier ce contenu.

Dans la Figure 71, le message « lien ok » indique que l'item « publications » a été traité, rappelle qu'un lien lui a été associé, donne la possibilité de le modifier par le lien « modifier ».

- **Partie 2** : elle indique au concepteur la marche à suivre pour définir les informations de chacun des items. Quand un item et un type d'information ont été choisis, l'utilisation du bouton « continuer » oriente le concepteur vers l'étape 6 « conception des contenus ».

- **Partir 3** : son contenu est défini lors de la réalisation de l'étape 6 « conception des contenus » (ci-dessous)

- **Partie 4** : elle contient des liens vers les étapes précédentes.

⁴⁶ présenté plus loin

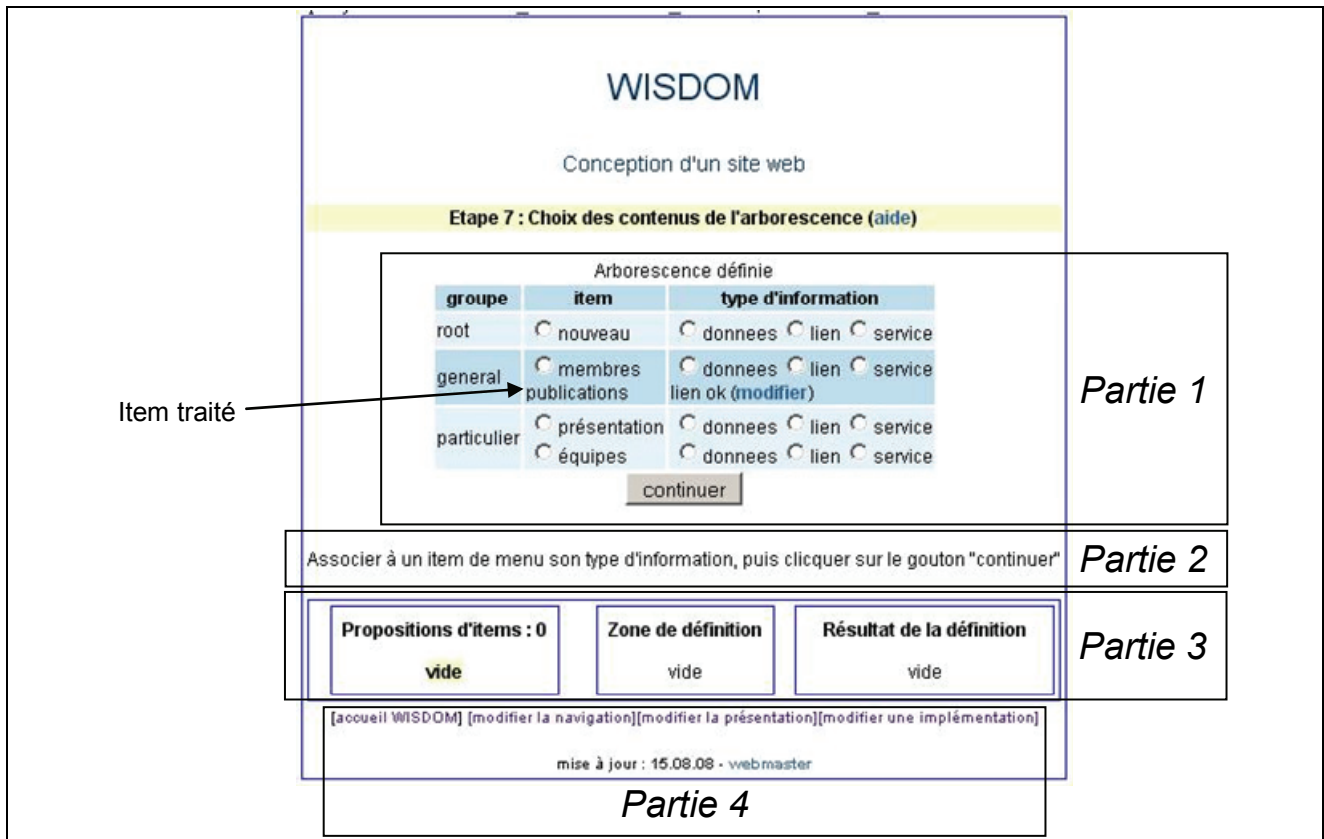


Figure 71 : choix d'un item de menu

1.6 Etape 6 : « conception des contenus »

Cette étape vise à définir le contenu associé à un item de menu, c'est-à-dire :

- les données dynamiques qui seront affichées lors de la sélection de l'item ;
- un lien direct vers un document statique ou une page externe ;
- un lien vers un service (flux RSS, composant métier, etc.).

Elle utilise une interface similaire à celles des étapes précédentes, mais son contenu est adapté au type d'information associé à l'item. La partie 6 de la Figure 72 présente l'interface dans le cas où l'item traité est « membre » et le contenu associé à l'item est un ensemble de données.

- **Partie 2** : elle propose des items de définition de l'item en cours de traitement à partir des analyses de sites web selon la dimension « contenu ». Dans le cas de la Figure 72, il n'y a pas de données concernant l'item en cours de traitement « membres ».

- **Partie 3** : elle affiche les items retenus dans la partie 2 ou ajoutés par le concepteur à l'aide du bouton « ajouter un item ». Les items sont définis par les deux colonnes suivantes :

- « items » : cette colonne affiche le label :
 - des propositions qui ont été retenues dans la partie 2,
 - des items qui ont été ajoutés par le concepteur (dans ce cas, le label par défaut est « nouveau ») ;
- « type » : elle propose les types « principal », « secondaire » ou « interne » pour qualifier les données associées à l'item de menu (ces types sont explicités dans le patron associé à l'étape, accessible à partir du lien « aide » affiché dans la partie 1 de l'interface).

WISDOM

Conception d'un site web

Etape 7 : Choix des contenus de l'arborescence (aide)

Partie 1

Item en cours de traitement →

Arborescence définie

groupe	item	type d'information
root	<input type="radio"/> nouveau	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service
general	<input checked="" type="radio"/> membres publications	<input checked="" type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service lien ok (modifier)
particulier	<input type="radio"/> présentation	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service
	<input type="radio"/> équipes	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service

Partie 6

Propositions d'items : 0

pas de donnees dans la base

Partie 2

Zone de définition

items	type
<input type="text" value="nom"/>	pincipal ▼
<input type="text" value="prénom"/>	pincipal ▼
<input type="text" value="fonction"/>	secondaire ▼
<input type="text" value="téléphone"/>	secondaire ▼
<input type="text" value="nouveau"/>	indéfinie ▼

Résultat de la définition

type	items
indéfinie	nouveau
pincipal	nom prénom
secondaire	fonction téléphone

Partie 4

[accueil WISDOM] [modifier la navigation][modifier la présentation][modifier une implémentation]

mise à jour : 15.08.08 - webmaster

Figure 72 : définition d'un contenu

Partie 4 : elle récapitule les données définies par le concepteur, triés par type de données. Le bouton « Sauvegarder » permet de :

- générer la table en fonction de la liste de données définie par le concepteur en utilisant le méta-modèle de structure de données ;
- modifier la partie 6 (Figure 73) pour présenter :
 - un message indiquant que l'item a été traité,
 - un lien pour modifier la définition des données,
 - un lien pour choisir l'implémentation de l'affichage des données à l'aide de l'étape 4 ;

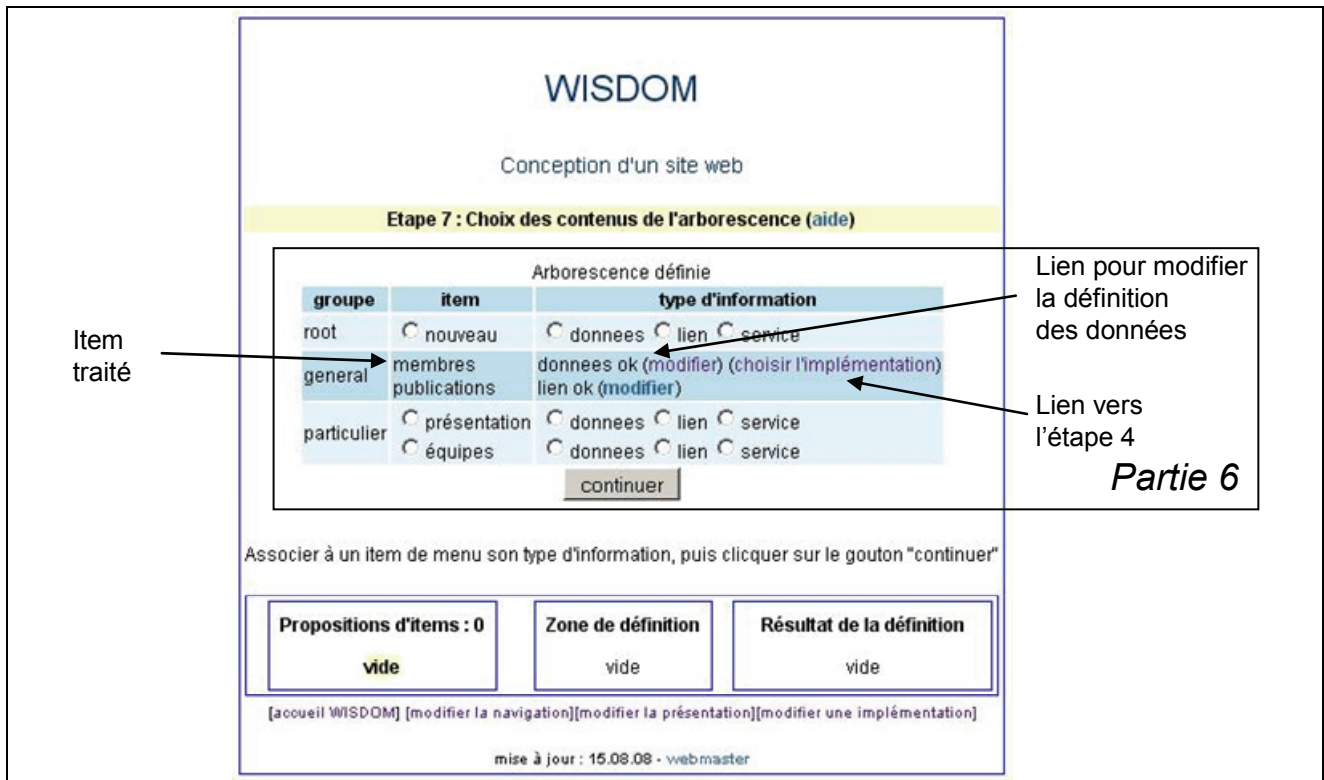


Figure 73 : données définies

La Figure 74 présente l'interface de la même étape dans le cas où un lien est associé à l'item de menu. La différence réside dans la partie 3, où l'interface présente un champ texte pour définir un lien vers un fichier ou vers une page externe.

La Figure 75 présente l'interface de la même étape dans le cas où un service est associé à l'item de menu. La différence réside dans la partie 2, où les propositions de service sont complétées par des liens de recherche (« Google », « GuidePHP », « catalogue de composants de WISDOM »).

La Figure 76 présente l'interface quand tous les items de menus ont été traités et toutes les implémentations définies. La différence réside dans la présence d'un message signalant que le processus est terminé.

WISDOM

Conception d'un site web

Etape 7 : Choix des contenus de l'arborescence (aide)

Arborescence définie

groupe	item	type d'information
root	<input type="radio"/> nouveau	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service
general	<input type="radio"/> membres	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service
	<input checked="" type="radio"/> publications	<input type="radio"/> donnees <input checked="" type="radio"/> lien <input type="radio"/> service
particulier	<input type="radio"/> présentation	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service
	<input type="radio"/> équipes	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service

Propositions d'items : 0

pas de donnees dans la base

Zone de définition

lien

Partie 3

Résultat de la définition

lien

[accueil WISDOM] [modifier la navigation][modifier la présentation][modifier une implémentation]

mise à jour : 15.08.08 - webmaster

Figure 74 : association d'un lien à un item de menu

WISDOM

Conception d'un site web

Etape 7 : Choix des contenus de l'arborescence (aide)

Arborescence définie

groupe	item	type d'information
root	<input checked="" type="radio"/> nouveau	<input type="radio"/> donnees <input type="radio"/> lien <input checked="" type="radio"/> service
general	membres	donnees ok (modifier) (choisir l'implémentation)
	publications	lien ok (modifier)
particulier	<input type="radio"/> présentation	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service
	<input type="radio"/> équipes	<input type="radio"/> donnees <input type="radio"/> lien <input type="radio"/> service

Propositions d'items : 0

réutilisations > 11

Nombre de sites analysés : 50

Recherche sur google

Recherche sur GuidePHP

Recherche dans le catalogue de composants

pas de donnees dans la base

Partie 2

Zone de définition

service

Résultat de la définition

service

[accueil WISDOM] [modifier la navigation][modifier la présentation][modifier une implémentation]

mise à jour : 15.08.08 - webmaster

Figure 75 : association d'un service à un item



Figure 76 ; processus terminé

1.7 Aide à l'utilisateur

Cette aide est constituée par nos patrons de domaine. Sur chaque interface associée aux étapes dédiées à une dimension de modélisation, le patron de domaine correspondant est accessible à partir d'un lien « aide », situé à côté du titre de l'étape dans la partie supérieure de l'interface [Cocquebert, 08b]. Présentés en Annexe 3, ces patrons de domaines ont été formalisés en P-Sigma et se présentent sous la forme de pages statiques.

Après avoir présenté la mise en œuvre du processus de conception, la partie suivante présente la mise en œuvre du catalogue d'analyses de sites web.

2 Analyse de sites web

Cette partie présente la mise en œuvre et l'utilisation des fiches d'analyses de sites web. L'exploitation des fiches n'est pas abordée dans cette partie car elle a été présentée lors du détail des étapes du processus de conception (cf. § 2.1.3 page 90).

2.1 Mise en œuvre des fiches d'analyses

Quelque soit la dimension analysée, une fiche d'analyse propose des rubriques, dont la définition est réalisée à l'aide de l'interface présenté Figure 77, composée de deux parties.

- **Partie 1** : elle présente la liste de toutes les rubriques déjà définies et offre la possibilité de consulter leurs caractéristiques (bouton « afficher »), ou d'en définir une nouvelle (bouton « nouveau »). Dans ce dernier cas, l'utilisation du bouton provoque l'affichage de la partie 2, détaillée maintenant.

- **Partie 2** : elle présente les caractéristiques d'une rubrique qui a été sélectionnée dans la liste de la partie 1. Ces caractéristiques sont :

- « nom » : indique le nom générique qui est affiché dans la fiche d'analyse ;
- « description » : explicite le nom de la rubrique pour que l'utilisateur en comprenne bien la sémantique ;
- « type de site web » : indique le ou les catégories de sites web concernés par la rubrique (dans la Figure 77, la rubrique « nom » est utilisée dans les sites de types « présentation entité ») ;

- « dimension » : indique la dimension de modélisation concernée par la rubrique (dans la Figure 77, la rubrique « nom » est utilisée pour analyser les données) ;
- « donnée définie » : si la rubrique est une sous-rubrique, c'est-à-dire qu'elle définit une autre rubrique, le nom de cette dernière est indiquée par ce champ (dans la Figure 77, la rubrique « nom » est une donnée qui définit la donnée « membres »). Si le label ne définit pas une donnée, ce champ est vide.

Administration externe

Données enregistrées

Donnée définie	membres
Type de site web	<input checked="" type="checkbox"/> Présentation entité <input type="checkbox"/> Evènement <input type="checkbox"/> Groupe <input type="checkbox"/> Présentation équipe
Dimension	<input type="checkbox"/> Navigation <input type="checkbox"/> Organisation <input type="checkbox"/> Implémentation <input checked="" type="checkbox"/> Données <input type="checkbox"/> Service
Nom	Nom
Description	nom d'un membre

Partie 2

Gestion du contenu de "th_rubriques_v2"

<ul style="list-style-type: none"> Abonnement newsletter Acronyme Actes en ligne Actualités Actualités Anciens directeurs Anciens invités Annuaire Appels d'offre Archives
<input type="button" value="Nouveau"/> <input type="button" value="Détails"/>

Partie 1

[\[accueil WISDOM\]](#) [\[accueil catalogue\]](#) [\[accueil admin\]](#) [\[liste tables\]](#)

webmaster - maj : 18.08.08

Figure 77 : définition du contenu d'une fiche

2.2 Utilisation des fiches d'analyses

L'utilisation des fiches d'analyses pour analyser un site web en production nécessite de référencer ce site web dans le catalogue. Cette action est réalisée à l'aide de l'interface présentée Figure 78, composée de deux parties.

- **Partie 1.** Elle présente la liste des sites web déjà référencés, avec la possibilité de consulter les caractéristiques d'un site web (« bouton « détails ») ou d'en ajouter un nouveau (« bouton « nouveau »). Dans ce dernier cas, l'utilisation du bouton provoque l'affichage de la partie 2.
- **Partie 2.** Elle présente les différents champs de définition suivants :
 - « nom » et « url » pour identifier le site web et connaître son adresse internet ;
 - le nom éventuel de l'entité d'attache afin de différencier les sites web des équipes et de leurs entités d'attache ;
 - le type de site web par rapport aux catégories actuellement envisagées.

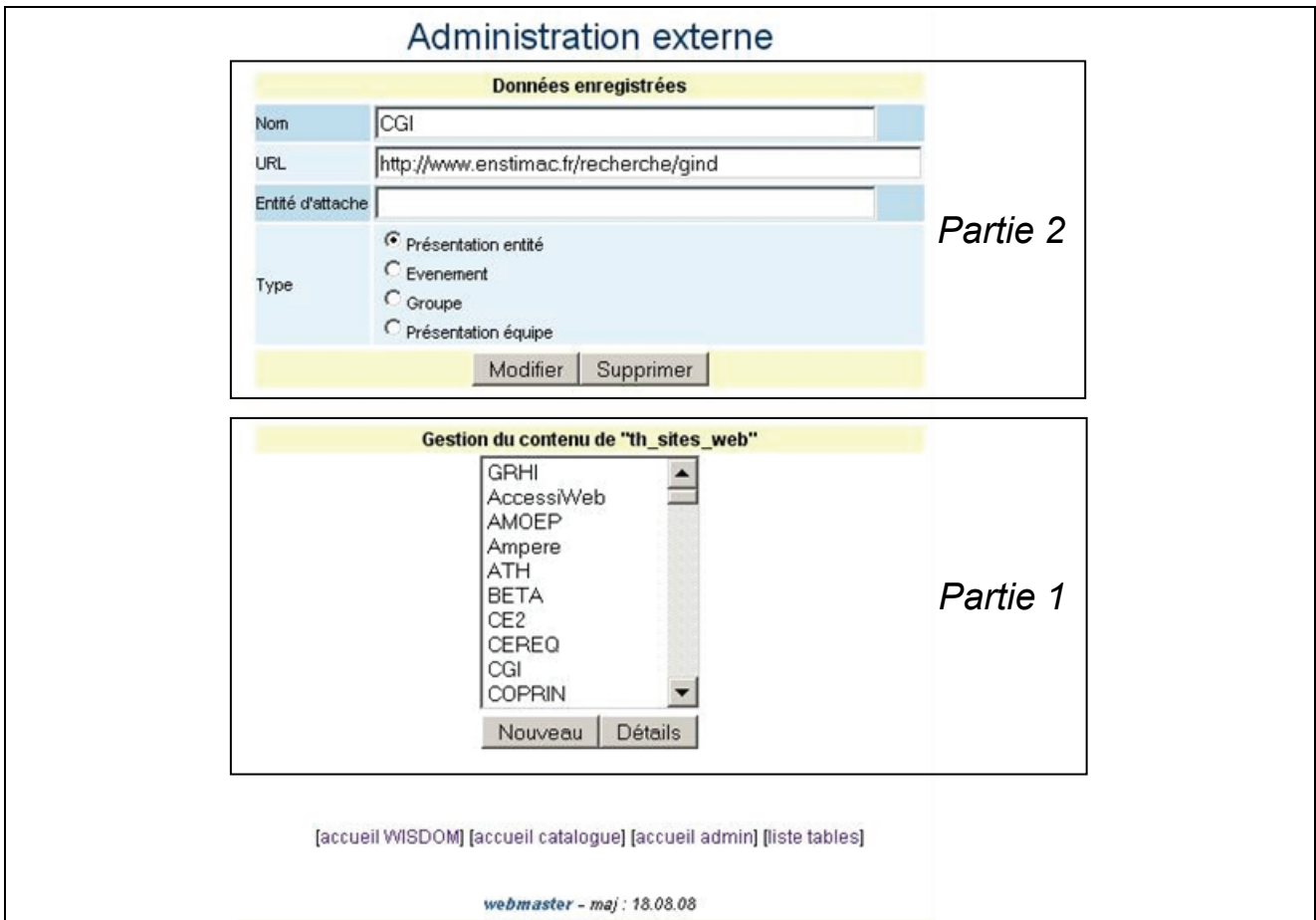


Figure 78 : référencement d'un site web

Quand un site web est référencé, il peut être analysé selon la dimension « navigation » ou « présentation » (Figure 79).

Analyses selon les différentes dimensions		
type : présentation entité		
Ampere	navigation	présentation
BETA	navigation	présentation
CEREQ	navigation	présentation
CGI	navigation	présentation
COPRIN	navigation	présentation
CPI	navigation	présentation
CPNI	navigation	présentation
CRAN	navigation	présentation
CREA	navigation	présentation
CRESTIC	navigation	présentation
D&DT	navigation	présentation

Figure 79 : extrait de la liste de sites web référencés

La Figure 80 présente l'interface pour l'analyse de la dimension « navigation », composée de deux parties.

Analyse de la dimension "navigation"				
Données enregistrées				
Nom	BETA			
URL	http://cournot2.u-strasbg.fr/users/beta/index.p			
Entité d'attache				
Type	présentation entité			

Partie 1

Indiquer la présence d'un item dans la navigation d'un site web en cochant la case correspondante (colonne 1) Définir les données associées si nécessaire à l'aide du lien associé (colonne 2)				
Nom générique	Indice de réutilisation	Contenu associé	Type d'implémentation	Service
<input checked="" type="checkbox"/> Annuaire (?)	8 sites	<input type="radio"/> données (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)	<input type="radio"/> liste (0 site) <input type="radio"/> graphique (0 site) <input type="radio"/> formulaire (0 site)	<input type="text"/>
<input checked="" type="checkbox"/> Appels d'offre (?)	1 site	<input type="radio"/> données (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)	<input type="radio"/> liste (0 site) <input type="radio"/> graphique (0 site) <input type="radio"/> formulaire (0 site)	<input type="text"/>
<input type="checkbox"/> Archives (?)	1 site	<input type="radio"/> données (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)	<input type="radio"/> liste (0 site) <input type="radio"/> graphique (0 site) <input type="radio"/> formulaire (0 site)	<input type="text"/>
<input checked="" type="checkbox"/> Collaborations (?)	25 sites	<input type="radio"/> données (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)	<input type="radio"/> liste (0 site) <input type="radio"/> graphique (0 site) <input type="radio"/> formulaire (0 site)	<input type="text"/>
<input type="checkbox"/> Connexion intranet (?)	16 sites	<input type="radio"/> données (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)	<input type="radio"/> liste (0 site) <input type="radio"/> graphique (0 site) <input type="radio"/> formulaire (0 site)	<input type="text"/>

Partie 2

Figure 80 : mise en œuvre de l'analyse de la navigation

- **Partie 1** : elle rappelle les caractéristiques du site web en cours d'analyse, ou dont l'analyse est en cours de consultation.

- **Partie 2** : elle présente l'interface pour analyser le site web selon la navigation, mais également selon la dimension « contenu » et « implémentation ». Elle est composée des colonnes suivantes :

- « nom générique » : cette colonne est relative à l'analyse de la dimension « navigation ». Elle propose une liste d'items de menu avec une case à cocher pour indiquer la présence de l'item dans le menu en cours d'analyse. Si le label de certains items est en gras surligné, cela signifie qu'une analyse a déjà été réalisée, et que les items surlignés ont été identifiés dans le menu du site web considéré. Le « ? » permet de consulter la description associée à l'item afin d'explicitier le nom ;
- « indice de réutilisation » : indique le nombre de réutilisation de l'item concerné parmi toutes les analyses des sites de même catégorie ;
- « contenu associé » : cette colonne est relative à l'analyse de la dimension « contenu ». Elle précise le type de contenu associé à l'item considéré, en précisant le nombre de réutilisation dans les analyses déjà réalisées ;
- « type d'implémentation » : cette colonne est relative à l'analyse de la dimension « implémentation ». Elle propose différents types d'implémentation pour l'affichage des données ;
- « service : propose une liste de service associé à l'item considéré dans les analyses déjà réalisées.

L'exemple indiqué dans la Figure 80 montre que le site web « BETA » possède au moins les items « annuaire », « appels d'offre » et « collaborations » dans son menu de navigation.

La Figure 81 présente l'interface pour l'analyse de la dimension « présentation », composée de deux parties.

- **Partie 1** : de même que précédemment, elle rappelle les caractéristiques du site web en cours d'analyse, ou dont l'analyse est en cours de consultation.

- **Partie 2** : elle présente l'interface pour analyser la présentation du site web à l'aide des colonnes suivantes :
 - « nom générique » : elle propose une liste d'items, auquel est associé un « ? » pour consulter la description associée à l'item afin d'explicitier le nom ;

- « type d’organisation et indice de réutilisation » : indique les différents types de l’analyse de l’organisation (« toutes pages » si la rubrique est commue à toutes les pages, « page d’accueil » si elle est spécifique à la page d’accueil), en indiquant l’indice de réutilisation de chaque type d’organisation.

Analyse de la dimension "présentation"		Partie 1
Données enregistrées		
Nom	BETA	
URL	http://cournot2.u-strasbg.fr/users/beta/index.p	
Entité d'attache		
Type	présentation entité	

Indiquer si une information est commune à toutes les pages ou spécifique à la page d'accueil		Partie 2
Nom générique	Type de présentation et indice de réutilisation	
Entités d'attache (?)	<input checked="" type="radio"/> toutes pages (35 sites) <input type="radio"/> page accueil (13 sites)	
Nom complet (?)	<input checked="" type="radio"/> toutes pages (33 sites) <input type="radio"/> page accueil (13 sites)	
	<input type="radio"/> toutes pages (0 site)	

Figure 81 : mise en œuvre de l’analyse de la présentation

Dans la figure ci-dessus, le site web « BETA » indique que le nom de l’entité d’attache (« BETA ») et le nom complet de l’entité (« Bureau d’Economie Théorique et Appliquée ») sont indiqués sur toutes les pages du site web.

La partie suivante présente le dernier module logiciel du WISDOM Tool.

3 Analyse de composants

Cette partie présente les interfaces utilisateurs qui permettent d’utiliser une fiche pour analyser un composant logiciel. Par contre, l’exploitation des analyses n’est pas présentée car elle a été abordée lors de la présentation des étapes du processus de conception

Afin d’avoir un référentiel commun de comparaison, il n’y a pas d’interface de création de la fiche : elle est définie de manière statique et son contenu est issu d’une réflexion menée au sein du groupe de travail « Briques logicielles » de l’AFUP⁴⁷ auquel je contribue. L’objectif de notre réflexion a été d’identifier les caractéristiques qui permettent de choisir un composant par rapport à un autre, puis de les implémentées afin de construire un catalogue d’analyse de composants. Ces caractéristiques sont relatives à :

- la facilité de l’installation et les contraintes sur l’environnement d’utilisation (Figure 82) ;
- un retour d’expérience sur l’utilisation (Figure 83) ;
- la qualité du support officiel ou de la communauté (Figure 84) ;
- un retour d’expérience sur l’exploitation en production (Figure 85)
- l’évolution dans le temps (Figure 86).

Elles sont complétées par un avis général (Figure 87) et par une identification du composant analysé et de l’auteur de la fiche (Figure 88).

De manière plus précise, les facettes sont renseignées :

- de manière booléenne (case à cocher) ;
- par une sélection parmi une liste fermée qui contient les choix « insuffisant », « correct », « très bien » ;
- par un texte libre.

⁴⁷ <http://www.afup.org/>

Installation	
Installation automatique	<input type="checkbox"/>
Configuration automatique	<input type="checkbox"/>
Navigation claire dans les sources (arborescence)	<input type="checkbox"/>
Compatible PHP5	<input type="checkbox"/>
Version PHP minimum	<input type="text"/>
Version MySQL minimum	<input type="text"/>
Autre BDD (préciser)	<input type="text"/>

Figure 82 : informations relatives à l'installation

Utilisation	
Prise en main facile	<input type="checkbox"/>
Moins de une heure <	<input type="checkbox"/>
Code source bien structuré	<input type="checkbox"/>
Standards respectés	<input type="checkbox"/>
Internationalisation possible	<input type="checkbox"/>
Interopérable	<input type="checkbox"/>
Outils fonctionnel clef en main	<input type="checkbox"/>
API existante	<input type="checkbox"/>
Absence de langage propriétaire	<input type="checkbox"/>
Développement de plugins possible	<input type="checkbox"/>
Fonctionnalités	<input type="text"/>

Figure 83 : informations relatives à l'utilisation

Support	
Documentation	<input type="text"/>
Documentation en ligne	<input type="checkbox"/>
Caractéristiques en lignes	<input type="checkbox"/>
Exemples clairs	<input type="checkbox"/>
Communauté de développeurs importante	<input type="checkbox"/>
Communauté de développeurs francophones importante	<input type="checkbox"/>
Version(s) de démo en ligne	<input type="checkbox"/>
FAQ	<input type="text"/>
Bonne réactivité sur le forum	<input type="text"/>
Bonne qualité des réponses sur le forum	<input type="text"/>
Activité sur un wiki	<input type="text"/>
Qualité du contenu du wiki	<input type="text"/>

Figure 84 : informations relatives au support

Exploitation	
Bonne montée en charge	<input type="checkbox"/>
Utilisation correcte des ressources serveur	<input type="checkbox"/>
Exécution plutôt rapide	<input type="checkbox"/>
Scripts compilés (moteur de template)	<input type="checkbox"/>
Facilité de mise en production	<input type="checkbox"/>

Figure 85 : informations relatives à l'exploitation

Figure 86 : informations relatives à l'évolution

Figure 87 : informations générales

Figure 88 : identification du composant

A partir de la version actuelle du catalogue, une synthèse des analyses est disponible sur <http://www.guidephp.com/>.

4 Conclusion

L'objectif de ce chapitre était de présenter une mise en œuvre possible des éléments constitutifs de WISDOM au sein du « WISDOM Tool ». Cette mise en œuvre se présente sous la forme d'un site web à partir duquel le concepteur accède aux modules « conception d'un système d'informations web », « catalogue d'analyses de sites web » et « catalogue d'analyses de composants ».

La conception d'un site web est guidée par la réalisation de différentes étapes du processus de la méthode WISDOM, dont la succession est prise en charge par l'informatisation du processus lui-même. Développées de manière générique, les interfaces utilisateurs ont pour tâche de :

- proposer des solutions à l'utilisateur ;
- donner la possibilité d'adapter ces solutions au site web en cours de conception ;
- sauvegarder les données de conception dans le modèle de site web.

Pour chacune des étapes, l'utilisateur a la possibilité de consulter le patron associé afin de disposer d'une assistance à la réalisation de cette étape. Dans l'état actuel de la mise en œuvre, le WISDOM Tool ne permet la création que d'un seul site web à la fois.

Pour analyser un site web, différents interfaces ont été mises en œuvre afin de permettre la création et l'utilisation d'une fiche d'analyse. Ces interfaces ont été développés de manière générique afin d'être réutilisables pour chacune des dimensions d'analyse des sites web.

Pour analyser les composants et bibliothèques, nous avons présenté les interfaces correspondant à la saisie des informations. Les rubriques de ces interfaces sont issues d'une réflexion menée dans le cadre d'un groupe de travail de l'AFUP afin d'avoir un référentiel de comparaison.

Après avoir présenté une mise en œuvre de la méthode WISDOM au sein du WISDOM Tool, le chapitre suivant présente son utilisation pour concevoir un site web, et présente des éléments de la validation des résultats de notre travail.

Chapitre VI : Application et validation

Ce chapitre présente l'utilisation de WISDOM Tool pour la conception d'un site web, puis des éléments de validation de la méthode et de son outil associé.

1 Présentation du cas d'application

Cette partie présente une partie de la conception d'un site web réel à l'aide de WISDOM Tool. Le site est celui du LAMIH (Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines de l'Université de Valenciennes et du Hainaut Cambrésis), entité de recherche UMR CNRS dont nous faisons partie. Le cahier des charges, donné par le directeur, demande la disponibilité des pages suivantes :

- pages publiques pour diffuser des informations sur les objectifs de recherche généraux du laboratoire et par équipes, la composition des équipes, les publications, les événements à venir, les projets de recherche, etc.
- pages privées, constituant un point central d'informations sur les différents documents de fonctionnements administratifs, les logiciels et équipements disponibles, etc.

La suite de cette partie décrit une partie du processus pour la conception du site web du LAMIH selon ce cahier des charges [Cocquebert, 08a] [Cocquebert, 08b], en indiquant des éléments de la validation que cette conception a mis à jour.

2 Conception du site web

2.1 Etape « identification du contexte »

La première étape du processus de conception vise à restreindre l'ensemble des analyses de sites web qui permettront de faire des propositions de solutions de conception. Dans le cas du site du LAMIH, c'est le type « présentation entité » qui doit être choisi (cf. Figure 89)

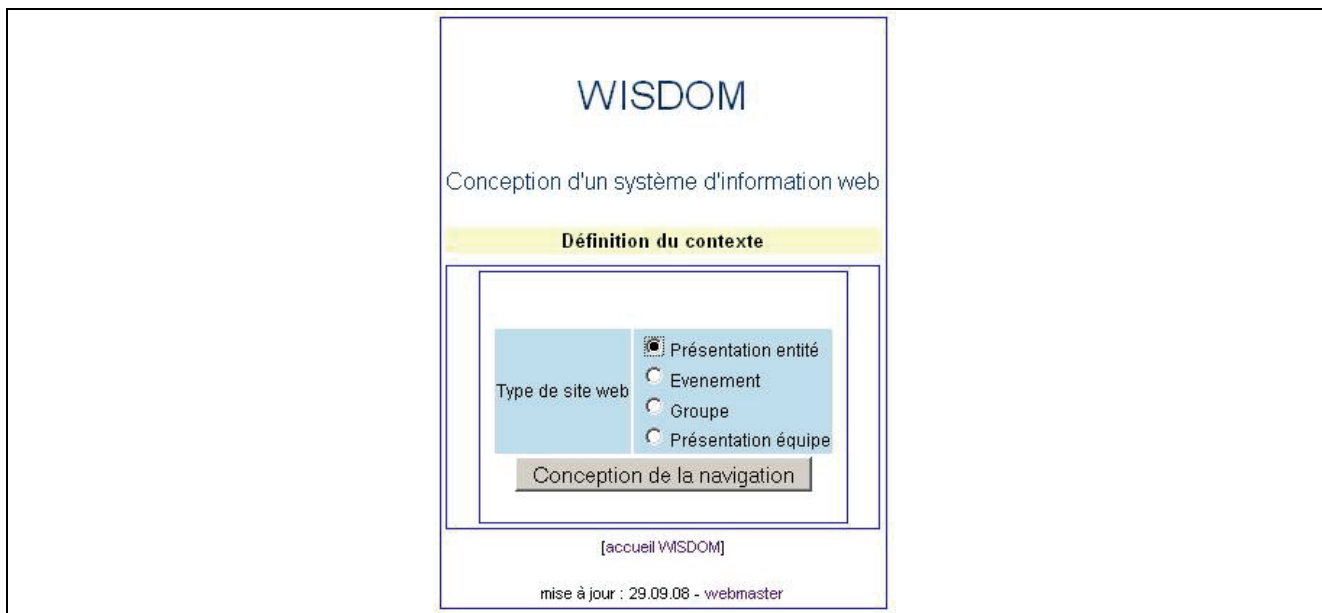


Figure 89 : contexte de la conception du site du LAMIH

Quand le contexte est défini, l'interface oriente l'utilisateur vers l'étape « conception de la navigation ».

2.2 Etape « conception de la navigation »

En utilisant l'interface utilisateur d'exploitation des analyses, nous avons défini l'arborescence du site du LAMIH, dans laquelle nous distinguons trois types d'items (Figure 90) :

- items de type 1, correspondant à des associations directe avec les besoins exprimés dans le cahier des charges ;
- items de type 2, retenus car jugés pertinents pour le site et qui étaient proposés car utilisés dans d'autres sites web de la même catégorie ;

- items de type 3, issus d'une réflexion fondée sur les deux types précédents.

Ce détail nous permet de constater que l'interface répond à trois nécessités :

- formaliser rapidement un élément explicite du cahier des charges par la proposition de labels d'items (items de type 1) ;
- enrichir le cahier des charges initial en y ajoutant des items non envisagés dans le cahier des charges mais :
 - proposés à partir des analyses (items de type 2),
 - issus d'une réflexion qui a conduit à l'insertion de nouveaux items (items de type 3).

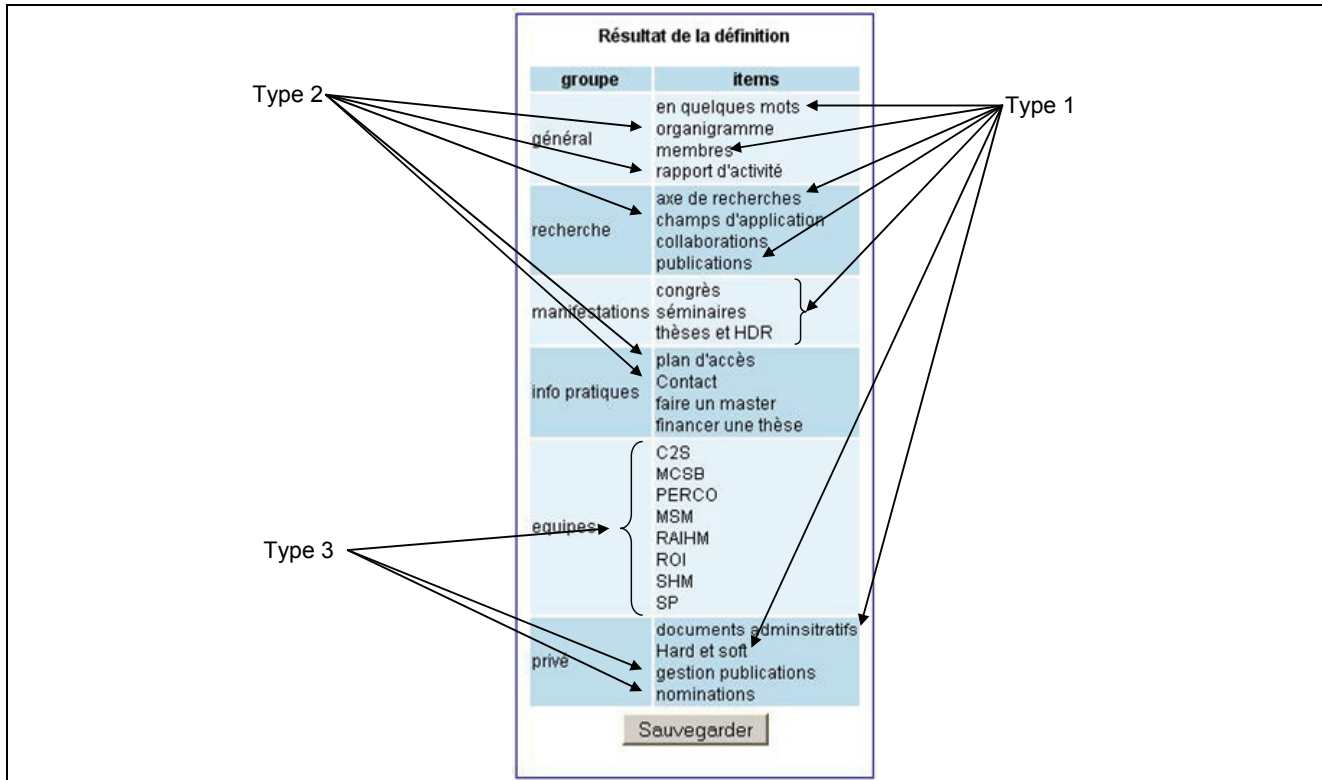


Figure 90 : liste adaptée des items de navigation

Quand l'utilisateur sélectionne le bouton « Sauvegarder » :

- la table correspondant à l'arborescence est générée et initialisée à partir des choix du concepteur ;
- l'architecture d'accueil et les fichiers associés aux items sont générés ;
- le concepteur est orienté vers l'étape « conception de la présentation ».

2.3 Etape « conception de la présentation »

En utilisant l'interface utilisateur de cette étape et en exploitant les analyses selon la dimension « présentation », les informations communes à toutes les pages du site web ont été définies (Figure 91).

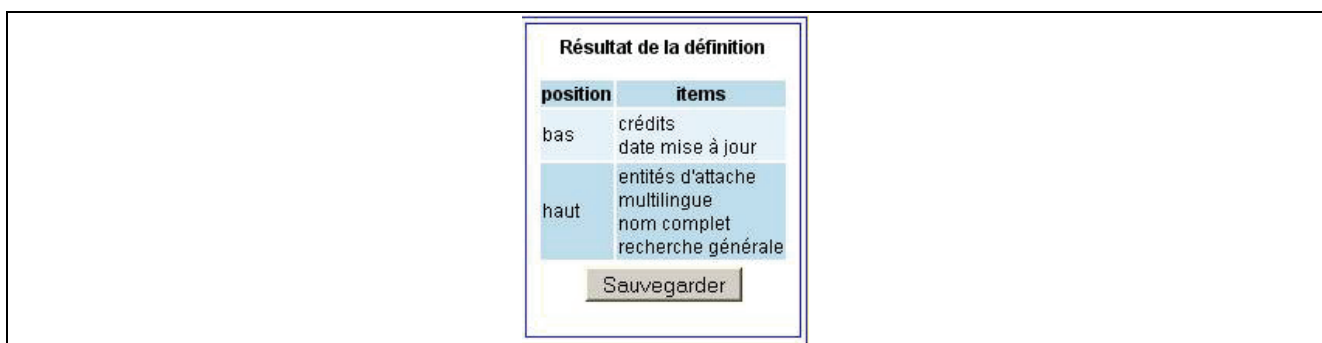


Figure 91 : liste adaptée des informations communes à toutes les pages

Concernant le modèle de présentation des pages du site web, il a été défini en utilisant Dreamweaver™, mais en s'appuyant sur la présentation des différents types de zones dans le patron associé à l'étape. En se référant au cas d'application du patron « présentation », la Figure 92 montre le modèle de présentation du site web du LAMIH dans lequel nous retrouvons les zones suivantes :

- une « zone commune » composée de logos et de la zone de recherche ;
- une « zone accueil » pour afficher les informations spécifiques à la page consultée ;
- une « zone annonce » pour afficher les événements à venir (congrès, séminaires, soutenances, etc.) ;
- des « zones navigation » repérées par des labels : « zone_navigation » pour afficher le menu de navigation, « zone_menu_equipes » pour afficher un menu spécifique aux équipes, et « zone_path » pour afficher la localisation de la page consultée dans l'arborescence.

En associant dans la Figure 92 le cas d'application du patron « présentation » et le modèle de présentation du site du LAMIH, nous montrons l'intérêt des concepts de zones que nous avons défini car ils ont permis de concevoir rapidement ce modèle du site du LAMIH et de le décrire simplement.

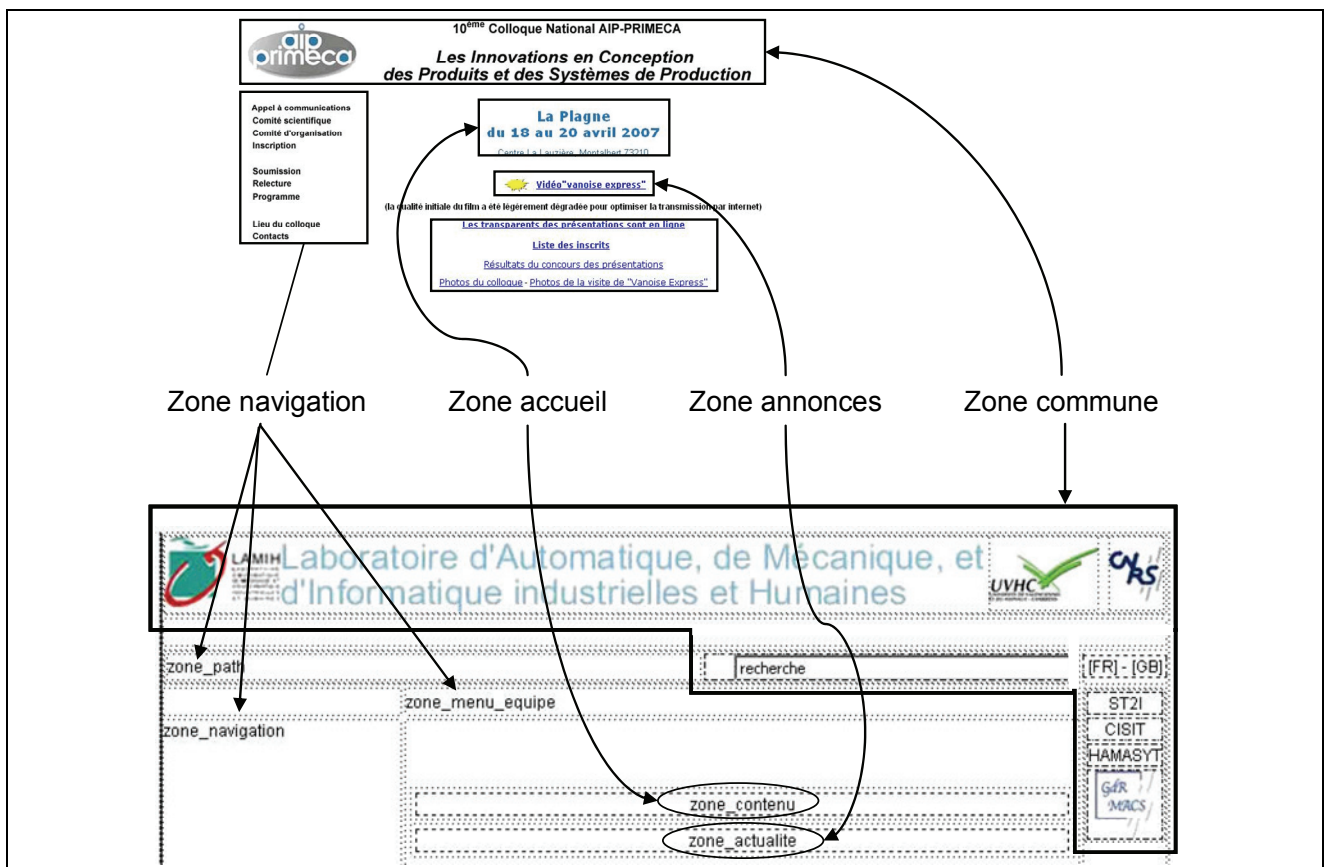


Figure 92 : modèle de présentation du site web du LAMIH

Quand l'utilisateur sélectionne le bouton « Sauvegarder » (cf. Figure 91), il est orienté vers l'étape « définition d'une implémentation » pour la fonction « affichage d'un menu ».

2.4 Etape « définition d'une implémentation » pour l'affichage du menu

L'interface associée à cette étape affiche directement les composants techniques référencés qui répondent à la fonction « affichage d'un menu » (Figure 93), et qui sont en fait des méthodes du framework du WISDOM Tool. Puisque la liste n'est pas vide, nous n'avons ni à effectuer une recherche sur internet ni à spécifier et réaliser un développement spécifique.

WISDOM

Conception d'un site web

Etape 4 : Choix d'une implémentation

Recherche dans le catalogue de composants

fonction

nature

[Recherche sur Google] [Recherche sur GuidePHP]

<p>Composants identifiés <i>Fonction "affichage menu"</i></p> <p><input checked="" type="checkbox"/> deroulant_dynamique</p> <p><input checked="" type="checkbox"/> menu en ligne</p> <p><input type="checkbox"/> développement spécifique</p> <p style="text-align: center;"><input type="button" value="Retenir"/></p>	<p>Composants retenus</p> <p><input checked="" type="radio"/> deroulant_dynamique</p> <p><input type="radio"/> menu en ligne</p> <p style="text-align: center;"><input type="button" value="Choisir"/> <input type="button" value="Ajouter un item"/></p> <p style="text-align: center; font-size: 1.2em;"><i>Partie 2</i></p>	<p>Choix final</p> <p>deroulant_dynamique</p> <p style="text-align: center;"><input type="button" value="Sauvegarder"/></p> <hr/> <p style="text-align: center; font-size: 1.2em;"><i>Partie 3</i></p>
---	---	---

[accueil WISDOM] [modifier la navigation][modifier la présentation]

mise à jour : 15.08.08 - webmaster

Figure 93 : choix de l'implémentation du menu

La consultation des fiches de renseignements permet de savoir :

- « deroulant_dynamique » est relié à la base de données, est situé en vertical sur le côté gauche de la page, est utilisé sur le site web de GISEH⁴⁸, est directement utilisable (copie de ligne de programme) ;
- « menu_en_ligne » est relié à la base de données, est situé en horizontal dans la partie supérieure de la page, est utilisé sur le site web du GDR MACS⁴⁹, n'est pas utilisable directement (généricité à mettre en place dans les lignes de programme concernées).

En considérant que le composant « déroulant_dynamique » est directement utilisable, et que le nombre d'items du site du LAMIH est sujet à évolution, donc que la taille de la zone de navigation risque de varier, notre choix s'est porté sur le composant technique « déroulant_dynamique ». Cependant, son insertion dans l'architecture logicielle a été réalisée manuellement.

Même si son contenu est succinct, l'utilisation de cette interface montre son intérêt pour aider à choisir un composant technique pour répondre à une fonction de base. Elle évite de lancer impérativement une recherche sur internet, ou de spécifier et réaliser un développement spécifique pour implémenter le menu.

Quand l'utilisateur indique son choix (bouton « choisir » de la partie 2), l'utilisation du bouton « sauvegarder » (dans la partie 3) l'oriente vers l'étape « choisir un item de menu ».

2.5 Etape « choisir un item de menu »

L'interface associée à cette étape permet de gérer et de suivre la progression de la conception du site lui-même. Nous détaillons dans le paragraphe suivant la conception des informations associées aux items « membres » et « documents administratifs » (Figure 94).

⁴⁸ <http://www.univ-valenciennes.fr/GDR-MACS/giseh/>

⁴⁹ <http://www.univ-valenciennes.fr/GDR-MACS/>

WISDOM

Conception d'un site web

Etape 7 : Choix des contenus de l'arborescence (aide)

Associer ci-dessous un item de menu et son type d'information, puis cliquer sur le bouton "continuer"

Arborescence définie

groupe	item	type d'information
général	<input type="radio"/> en quelques mots	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> organigramme	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> membres	<input type="radio"/> donnees (2 sites) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> rapport d'activité	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (1 site) <input type="radio"/> service (0 site)
recherche	<input type="radio"/> axe de recherches	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> champs d'application	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> collaborations	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> publications	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
manifestations	<input type="radio"/> congrès	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> séminaires	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> thèses et HDR	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
info pratiques	<input type="radio"/> plan d'accès	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> Contact	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> faire un master	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> financer une thèse	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
equipes	<input type="radio"/> C2S	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> MCSB	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> MSM	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> PERCO	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> RAIHM	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> ROI	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> SHM	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> SP	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
privé	<input type="radio"/> documents administratifs	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> Hard et soft	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> gestion publications	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> nominations	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)

Propositions d'items : 0

Zone de définition

Résultat de la définition

[accueil WISDOM] [modifier la navigation][modifier la présentation][modifier une implémentation]

mise à jour : 29.09.08 - webmaster

Items
détaillés

Figure 94 : choisir un item de menu pour y associer un contenu

2.5.1 Conception associée à l'item « membres »

2.5.1.1 Etape « conception d'un contenu »

Après avoir choisi dans l'interface ci-dessus le type de contenu « donnée » pour l'item « membre », nous avons sélectionné les données de définition de la page « membres » parmi celles qui étaient proposées à partir des analyses de sites web dans une interface similaire à celle de l'étape « conception de la navigation » (Figure 95).

WISDOM

Conception d'un site web

Etape 7 : Choix des contenus de l'arborescence (aide)

type de site : "présentation entité"

Arborescence définie

groupe	item	type d'information
général	<input type="radio"/> en quelques mots	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> organigramme	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input checked="" type="radio"/> membres	<input checked="" type="radio"/> donnees (1 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> rapport d'activité	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (1 site) <input type="radio"/> service (0 site)

Propositions d'items : 4
réutilisations > 1
Nombre de sites analysés : 50

- Fonction (2)
- Téléphone (2)
- Prénom (2)
- Nom (2)

Modification de la valeur de filtrage

Zone de définition

items	type
<input type="text" value="email"/>	secondaire ▼
<input type="text" value="Fonction"/>	secondaire ▼
<input type="text" value="Téléphone"/>	secondaire ▼
<input type="text" value="Prénom"/>	pincipal ▼
<input type="text" value="Nom"/>	pincipal ▼

Résultat de la définition

type	items
pincipal	Prénom Nom
secondaire	email Fonction Téléphone

Figure 95 : liste adaptée des items de navigation

Quand l'utilisateur indique que le choix des données de définition est terminé par un click sur le bouton « Sauvegarder », l'interface Figure 96 indique qu'un contenu a été associé à l'item (message « données ok »), donne la possibilité de modifier les données (message « modifier ») et de choisir l'implémentation de l'affichage de ces données (message « choisir l'implémentation »).

WISDOM

Conception d'un site web

Etape 7 : Choix des contenus de l'arborescence (aide)

Associer ci-dessous un item de menu et son type d'information, puis cliquer sur le bouton "continuer"

Arborescence définie

groupe	item	type d'information
général	<input type="radio"/> en quelques mots	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> organigramme	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> service (0 site)
	<input type="radio"/> membres	donnees ok (modifier) (choisir l'implémentation)
	<input type="radio"/> rapport d'activité	<input type="radio"/> donnees (0 site) <input type="radio"/> lien (1 site) <input type="radio"/> service (0 site)

Figure 96 : traitement terminé pour l'item « membre »

2.5.1.2 Etape « définition d'une implémentation » pour l'affichage d'un contenu

L'interface associée à cette étape présente directement une liste de composants et de bibliothèques dont la fonction est d'afficher un contenu selon différentes manières (Figure 97).

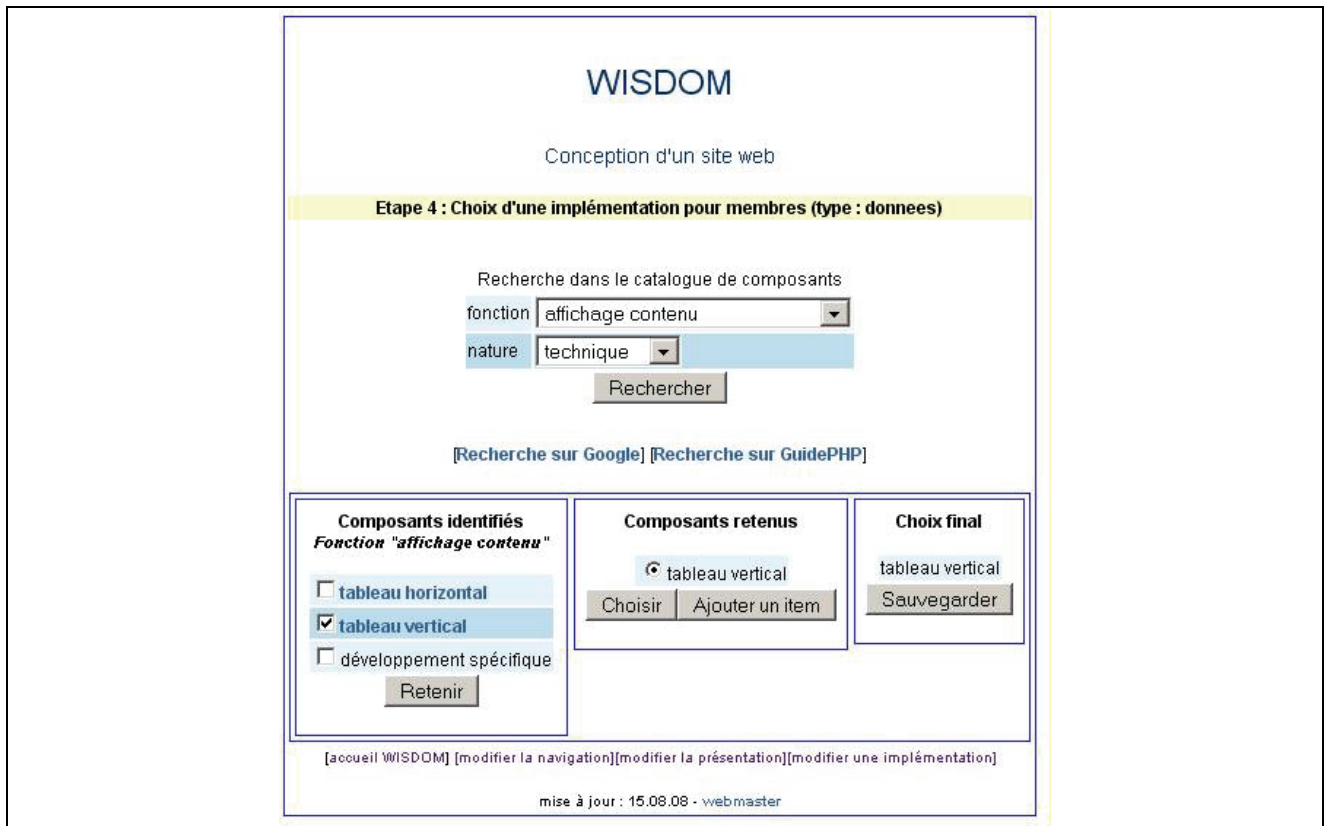


Figure 97 : composants techniques pour la fonction « affichage de données »

Ces composants techniques correspondent également à des méthodes du framework du WISDOM Tool. Les données affichées étant le nom et le prénom des membres du LAMIH, notre choix se porte sur un affichage sous la forme d'un tableau vertical car plus habituel pour ce type d'informations.

Quand l'utilisateur indique que le choix est réalisé en utilisant le bouton « Sauvegarder », il est orienté à nouveau vers l'étape « choix d'un item de menu » (Figure 98), et l'interface indique que l'implémentation des données de l'item « membre » a été choisi par le message « modifier l'implémentation ». Cependant, la programmation de l'utilisation du composant technique (c'est-à-dire l'appel de la méthode du framework du WISDOM Tool) est laissée à l'initiative du concepteur.



Figure 98 : traitement terminé du choix de l'implémentation

2.5.2 Conception associée à l'item « documents administratifs »

Après avoir indiqué dans l'interface de l'étape « choisir un item de menu » que le type de contenu de l'item « documents administratifs » est « service », l'utilisation du bouton « continuer » affiche le résultat de la recherche dans le catalogue de composants métiers pour la fonction « gestion de documents » (Figure 99).

The screenshot shows a web interface for selecting components. At the top, there are radio buttons for selecting the type of content: 'documents administratifs' (selected), 'Hard et soft', 'gestion publications', 'nominations', 'donnees (0 site)', 'lien (0 site)', and 'service (0 site)'. Below this are 'Rechercher' and 'continuer' buttons. The main search area has 'fonction' set to 'gestion de documents' and 'nature' set to 'métier', with a 'Rechercher' button. Below the search area are links for '[Recherche sur Google]' and '[Recherche sur GuidePHP]'. The results section is titled 'Propositions d'items : 4' and 'Fonction "gestion de documents"'. It lists four items: 'DocMan', 'LAMIHDoc', 'OpenDocMan', and 'développement spécifique', each with a checkbox. A 'Retenir' button is at the bottom of the list. To the right of the list are two empty boxes labeled 'Composants retenus' and 'Choix final'. At the bottom of the interface are navigation links: '[accueil VMSDOM]', '[modifier la navigation]', '[modifier la présentation]', and '[modifier une implémentation]'. The footer indicates 'mise à jour : 29.09.08 - webmaster'.

Figure 99 : composants techniques pour implémenter le service « gestion de documents »

En consultant les fiches de renseignements associés à chacun des composants proposés, nous observons que :

- « DocMan⁵⁰ » est un logiciel open source, mais c'est un module de Joomla⁵¹ et nécessite son installation ;
- « LAMIHDoc⁵² » est un logiciel open source, dédié à la gestion de documents, n'apporte pas de fonctions non souhaitées, est simple à configurer et dispose d'une interface en langue française ;
- « OpenDocMan⁵³ » est très complet du point de vue de la gestion d'informations (documents, liens, etc.) mais il est complexe à configurer, et son interface utilisateur est en langue anglaise.

Compte tenu de ces renseignements, notre choix s'est porté sur la proposition « LAMIHDoc ».

En indiquant que le composant est choisi, l'interface oriente l'utilisateur à nouveau sur l'étape « choisir un item de menu » en indiquant (Figure 100) :

- que le traitement de l'item « documents administratifs » est terminé ;
- un lien pour modifier le choix du service.

⁵⁰ http://www.joomlafrance.org/telecharger/select/Docman/par_titre/1.html

⁵¹ <http://www.joomla.fr/>

⁵² Composant développé dans le cadre d'un projet étudiant

⁵³ <http://www.opendocman.com/>

privé	documents administratifs	service ok (modifier le service)		
	<input type="radio"/> Hard et soft <input type="radio"/> gestion publications <input type="radio"/> nominations	<input type="radio"/> donnees (0 site) <input type="radio"/> donnees (0 site) <input type="radio"/> donnees (0 site)	<input type="radio"/> lien (0 site) <input type="radio"/> lien (0 site) <input type="radio"/> lien (0 site)	<input type="radio"/> service (0 site) <input type="radio"/> service (0 site) <input type="radio"/> service (0 site)
<input type="button" value="continuer"/>				
Propositions d'items : 0		Zone de définition		Résultat de la définition
[accueil WMSDOM] [modifier la navigation] [modifier la présentation] [modifier une implémentation]				
mise à jour : 29.09.08 - webmaster				

Figure 100 : traitement terminé du service « gestion publications »

2.6 Résultat

La Figure 101 montre la page d'accueil, basé sur le modèle définissant les zones d'affichage et intégrant l'implémentation du menu ; la Figure 102 présente la page « membre », également basé sur le modèle ; la Figure 103 montre l'interface du composant « LAMIHDoc », lancé à partir du site web.

 Laboratoire d'Automatique, de Mécanique, et d'Informatique industrielles et Humaines		 							
<table border="1"> <tr><td>Général</td></tr> <tr><td>Recherche</td></tr> <tr><td>Manifestations</td></tr> <tr><td>Info pratiques</td></tr> <tr><td>Equipes</td></tr> <tr><td>Privé</td></tr> <tr><td>admin</td></tr> </table>		Général	Recherche	Manifestations	Info pratiques	Equipes	Privé	admin	<input type="text" value="recherche"/> [FR] - [GB]
Général									
Recherche									
Manifestations									
Info pratiques									
Equipes									
Privé									
admin									
Bienvenue sur le site du LAMIH									
Actualités : seminaire : 13/03/08 : Ingénierie des Facteurs Humains pour la Survei... -									
ST2I CISIT HAMASYT 									
© LAMIH-SP - Crédits et mentions légales									

Figure 101 : page d'accueil du site web du LAMIH

 Laboratoire d'Automatique, de Mécanique, et d'Informatique industrielles et Humaines		 							
accueil >> membres		<input type="text" value="recherche"/> [FR] - [GB]							
<table border="1"> <tr><td>Général</td></tr> <tr><td>Recherche</td></tr> <tr><td>Manifestations</td></tr> <tr><td>Info pratiques</td></tr> <tr><td>Equipes</td></tr> <tr><td>Privé</td></tr> <tr><td>admin</td></tr> </table>		Général	Recherche	Manifestations	Info pratiques	Equipes	Privé	admin	ST2I CISIT HAMASYT 
Général									
Recherche									
Manifestations									
Info pratiques									
Equipes									
Privé									
admin									
Cocquebert Etienne									
Fonction : ingénieur d'études Telephone : 03 27 51 13 21 Email : etienne.cocquebert@univ-valenciennes.fr									
3 inscrits au 17-09-2008									
 Cocquebert Etienne  Tahon Christian  Trentesaux Damien									
© LAMIH-SP - Crédits et mentions légales									

Figure 102 : page « membre » du site web du LAMIH



Figure 103 : composant « LAMIHDoc »

2.7 Conclusion sur la conception du site web du LAMIH

D'un point de vue global, la méthode WISDOM et le WISDOM Tool ont rempli à la fois les rôles de support de communication et d'outil de développement pour la mise en œuvre de la structure d'une nouvelle version du site web du LAMIH.

De manière plus précise, la conception du site web du LAMIH en utilisant le WISDOM Tool a consisté à :

- concevoir la navigation, la présentation, les contenus et choisir les implémentations ;
- générer une architecture d'accueil et les structures de données ;
- personnaliser les fichiers associés aux items de la navigation (interaction avec la base de données, préparation des données et génération de l'affichage).

Ce processus a été réalisé sur une durée équivalente à une semaine à plein temps de travail.

Comme le résultat est un site web prêt à être alimenté, il est nécessaire de réaliser les tâches suivantes avant de le mettre en ligne :

- utilisation du module de gestion de contenu (intégré au site web du LAMIH de par l'utilisation de WISDOM Tool pour son développement) pour alimenter les différentes rubriques ;
- transfert et adaptation d'une partie du contenu de l'ancien site web du LAMIH ;
- amélioration du graphisme en collaboration avec un spécialiste du domaine ;
- connexion logicielle entre le site web du LAMIH et des composants spécifiques du système d'information de l'Université de Valenciennes et du Hainaut Cambrésis (base de données, authentification centralisée, etc.).

Après avoir présenté un exemple d'utilisation du WISDOM Tool pour concevoir un site web, la partie suivante évalue la validation de la méthode et de l'outil.

3 Validation

Cette partie est basée sur notre analyse de la conception des sites web suivants :

- « CAO-Calcul'08⁵⁴ » (journée technique organisées sous l'égide de l'AIP-PRIMECA sur la liaison entre la CAO et le calcul), dont la page d'accueil est présentée en Annexe 5 ;
- « GISEH⁵⁵ » (groupe de travail du GDR CNRS « MACS⁵⁶ ») , dont la page d'accueil est présentée en Annexe 6 ;
- nouvelle version du site du LAMIH, appelé dans la suite de cette partie « LAMIH_new⁵⁷ »

D'un point de vue général, une validation complète doit considérer :

⁵⁴ <http://www.univ-valenciennes.fr/sp/CAOCalcul/>

⁵⁵ <http://www.univ-valenciennes.fr/GDR-MACS/giseh/>

⁵⁶ <http://www.univ-valenciennes.fr/GDR-MACS/>

⁵⁷ http://www.univ-valenciennes.fr/sp/cocquebert/wisdom/LAMIH_new/ (lien non définitif)

- le processus de développement, en utilisant en tant que critères de validation les objectifs que nous nous sommes fixés pour les fonctionnalités à mettre en œuvre au sein de WISDOM ;
- le site web en tant que produit logiciel, en utilisant les critères définis par la norme ISO 9126 (cf. Annexe 4), définissant les caractéristiques qualités principales pour évaluer un produit logiciel ;
- l'utilisation du site web résultat, en utilisant les critères sur l'efficacité de la navigation, la qualité de la présentation, la pertinence du contenu, etc., en utilisant des listes de critères définies par FORMIST⁵⁸, l'INSA de Lyon⁵⁹, l'université de Montréal⁶⁰, etc.

Dans ce mémoire, nous nous focalisons sur la première catégorie de critères, c'est-à-dire que nous examinons pour chacune des fonctionnalités énoncées dans le chapitre 1 si les objectifs que nous nous sommes fixés ont été mis en œuvre dans la méthode et dans l'outil [Cocquebert, 08b].

3.1 Fonctionnalité « Assister la réalisation du processus de conception »

Concernant cette fonctionnalité, les critères de validation sont les suivants :

- capacité à guider le concepteur vers les étapes valides du processus qui s'offrent à lui, compte tenu du résultat de l'étape qu'il vient de terminer ;
- capacité à aider à la réalisation des étapes en indiquant pour chacune des informations sur son objectif, ses entrées, ses résultats, ses méthodes, ses outils supports, etc.

L'apport de notre étude concerne les propositions suivantes :

- un processus de conception qui s'appuie sur un processus réel ;
- la modélisation d'un site web selon les dimensions « navigation », « présentation », « contenu » et « implémentation ». Parmi ces dimensions :
 - « navigation » et « présentation » s'inspirent directement de l'état de l'art ;
 - « contenu » reprend la dimension « donnée » de l'état de l'art mais intègre les possibilités d'associer un lien ou un service à un item ;
 - « implémentation » reprend la considération de cette dimension par quelques méthodes de conception de sites web, mais intègre la possibilité d'implémenter une fonction par un composant logiciel ou un service.

L'analyse de cet apport est basée sur l'utilisation de la méthode et de l'outil pour la conception des sites web « CAO-Calcul'08 », « GISEH » et « LAMIH_new ».

3.1.1 Capacité à guider le concepteur

3.1.1.1 Validation par rapport à la méthode

Par rapport à notre proposition de processus, l'utilisation de WISDOM par un concepteur débutant pour concevoir le site web de « GISEH » a montré l'intérêt de définir un processus de conception qui explicite les différentes dimensions de modélisation d'un site web. L'absence de la méthode aurait conduit ce concepteur à une ou plusieurs des possibilités suivantes :

- faire des recherches pour identifier un processus ;
- concevoir le site par une suite d'itérations « essais-erreurs » ;
- reprendre le processus traditionnel de conception, dont nous avons vu qu'il était différent par rapport au processus de conception d'un site web.

Le temps de conception aurait certainement été plus long.

Par rapport à nos propositions de modélisation d'un site web, notre utilisation de WISDOM lors de la conception de « CAO-Calculs '08 » a montré que la prise en compte de la dimension « implémentation » a conduit à la proposition de gérer les inscriptions en utilisant un composant logiciel. Si le bénéfice de ce point

⁵⁸ http://formist.enssib.fr/documents/Grille_d_analyse_de_sites_Web-n-1029-r-26-t-theme.html#

⁵⁹ <http://docinsa.insa-lyon.fr/sapristi/index.php?rub=1004>

⁶⁰ <http://www.santemontreal.qc.ca/pdf/documentations/pubs/grilleanal-sitesweb.pdf>

est mineur pour un concepteur confirmé, il est majeur pour un concepteur débutant car la méthode apporte la connaissance sur la prise en compte des composants logiciels pour implémenter une fonction.

3.1.1.2 Validation par rapport à l'outil

Par rapport à l'informatisation du processus, l'utilisation du WISDOM Tool lors de la conception du site de « GISEH » par le concepteur débutant lui a permis de concevoir le site web en deux semaines à partir d'un cahier des charges exprimé par des membres du groupe de travail sous la forme d'une liste informelle d'items de menus et de contenus associés. Dans ce délai et à partir des analyses de sites web de groupes de travail (cf. § 3.2), le concepteur a été guidé par le WISDOM Tool pour concevoir le site web selon les différentes dimensions de modélisation.

Si l'outil ne l'avait pas aidé, le concepteur aurait eu les possibilités suivantes :

- concevoir le site web « from scratch » en faisant des recherches afin d'acquérir l'expertise minimale pour la conception (le temps de conception aurait certainement été plus longs et le site de qualité inférieure car un apprentissage aurait été nécessaire) ;
- rechercher un outil de gestion de contenu parmi la liste disponible sur « CMSMatrix¹⁴ » dans lequel il n'aurait eu qu'à adapter les labels des menus (cette recherche aurait consommé du temps pour faire un choix objectif en consultant les caractéristiques fonctionnelles, les contraintes d'installation logicielle, etc. et l'étape suivante aurait consisté à faire l'apprentissage de l'outil) ;
- demander conseils sur des listes de diffusion spécialisées (cette solution évitait de faire des recherches, mais elle nécessitait d'analyser les réponses. La synthèse n'aurait pas forcément conduit à un choix unique et une recherche aurait été nécessaire pour approfondir ces propositions).

Le bénéfice de l'outil est donc d'éviter de consommer du temps en recherche par rapport à la manière de concevoir un site web et par rapport à un outil support de cette conception.

Par rapport à la gestion des différentes dimensions de modélisation dans les conceptions de « CAO-Calcul'08 », « GISEH » et « LAMIH_new », elle est validée par le succès de l'utilisation de l'outil sans avoir à les modifier manuellement.

Ainsi, que ce soit par rapport à la méthode ou par rapport à l'outil, ces analyses montrent que l'objectif d'aider le concepteur dans la conception d'un site web est atteint par nos propositions, et qu'elles ont été mises en œuvre au sein du WISDOM Tool.

3.1.2 Aider à la réalisation des étapes

Par rapport à l'état de l'art, l'apport de la méthode et de l'outil est de proposer des patrons de domaine par rapport aux différentes dimensions de modélisation d'un site web.

Formalisés au sein de la méthode et mis en œuvre au sein de l'outil, ils apportent encore un fois une aide majeure au concepteur débutant par les explications fournies sur l'objectif, le processus, la fonction, les supports et le résultat des étapes du processus de conception.

Nous pensons donc que leur présence et le contenu de leur formalisation est validée de par le bénéfice apporté au concepteur débutant lors de la conception du site de « GISEH ».

3.2 Fonctionnalité « Identifier et faciliter la réutilisation de solutions de conception »

Concernant cette fonctionnalité, les critères de validation sont les suivants :

- capacité à identifier et proposer des solutions de conception déjà utilisées dans des sites web en production ;
- capacité à faciliter la réutilisation des solutions de conception retenues par le concepteur parmi les propositions.

Ces deux points sont ici considérés simultanément étant donné le lien étroit existant entre eux.

L'apport de notre travail par rapport à l'état de l'art est :

- la définition, l'utilisation et l'exploitation d'un modèle d'analyse pour capitaliser et proposer l'utilisation de solutions de conception ;
- la formalisation de concepts dans l'étape « conception de la présentation » (types de zones) et dans l'étape « conception de contenu » (types de données) pour assister leur réalisation et la communication sur l'état de leur avancement.

L'analyse est basée sur l'utilisation de la méthode et de l'outil pour l'identification de solutions de conception lors de la conception des sites web « GISEH » et « LAMIH_new ».

3.2.1 Validation par rapport à la méthode

Dans le cadre de la conception du nouveau site du « LAMIH_new » et du site web de « GISEH », notre retour d'expérience est positif car nous avons constaté que l'utilisation de ces concepts lors de réunions de travail ont facilité la réalisation, la présentation et l'amélioration des choix réalisés sur :

- la navigation ;
- le modèle de présentation des pages du site web ;
- les données associées aux items du menu.

Si ces concepts n'avaient pas été définis :

- un vocabulaire commun aurait dû être défini au sein des groupes de travail ;
- la définition des choix aurait été liée à la consultation de sites similaires et à l'expression de ces consultations ;
- la communication et la prise de décision aurait été plus fastidieuse et/ou plus longue à mener.

3.2.2 Validation par rapport à l'outil

Concernant l'exploitation des solutions de conception proposées à partir des analyses de sites web, elle est validée de par le bénéfice apporté par les propositions elles-mêmes et par l'interactivité de l'outil.

Dans le cas de la conception du site web du LAMIH, cette exploitation a permis de supporter visuellement le résultat des discussions. En l'absence de l'outil, cela aurait obligé le groupe de travail sur le site web à

- identifier des solutions de conception pour la navigation, la présentation et les données en dehors des réunions ;
- adapter ces solutions pendant les réunions.

Le délai de l'adaptation des solutions de conception aurait été plus important.

Dans le cas de la conception du site de «GISEH », cette exploitation a participé à la rapidité avec laquelle la conception a été réalisée par un concepteur débutant.

Cette capacité de l'outil à proposer des solutions de conception en fonction de la catégorie du site à concevoir nous a conduits à définir des architectures d'accueil dédiées aux familles de sites web pour l'organisation des conférences « LaPlagne », « PENTOM », « journées thématiques AIP-PRIMECA » et « Journées semestrielles STP ». L'intérêt de ces architectures est de proposer dès leur installation :

- un menu de navigation ;
- un modèle de présentation ;
- les structures de données associées aux items de navigation ;
- l'intégration de services spécifiques (gestion de la relecture, inscription en ligne).

L'utilisation de l'architecture d'accueil dédiée aux sites web « LaPlagne » nous a permis de mettre en œuvre le site de « LaPlagne2009 » en cinq heures de travail. En l'absence de cette architecture dédiée, une conception « from scratch » aurait nécessité :

- la conception de la structure de données de la navigation (afin de faciliter la modification ou l'insertion de labels dans le menu) ;
- la conception du modèle de présentation en s'appuyant sur celles des conférences antérieures ;
- la conception des différents fichiers des items du menu en s'appuyant sur le modèle de présentation.

Ainsi, nous pensons que ces analyses apportent une première validation par rapport à nos propositions et notre mise en œuvre pour identifier et faciliter la réutilisation de solutions de conception.

3.3 Fonctionnalité « Identifier et faciliter le choix de composants logiciels »

Concernant cette fonctionnalité, les critères de validation sont les suivants :

- capacité à identifier des composants logiciels existants déjà utilisés dans des sites web en production ;
- capacité à faciliter le choix d'un composant utilisateur par la mise à disposition d'informations sur les fonctions des composants et sur des retours d'expérience d'utilisateurs.

Ces deux points sont également considérés simultanément étant donné le lien étroit existant entre eux.

L'apport de notre travail par rapport à l'état de l'art est :

- la proposition d'une caractérisation fonctionnelle et non fonctionnelle des composants logiciels ;
- la proposition d'un catalogue pour utiliser ces caractérisations.

L'analyse est basée sur l'utilisation de la méthode et de l'outil pour l'identification de solutions de conception lors de la conception du site web de « CAO-Calcul'08 ».

3.3.1 Validation par rapport à la méthode

Les caractéristiques ont été définies dans le cadre d'un groupe de travail regroupant des concepteurs actifs. Afin d'obtenir leur validation, nous avons envoyé des appels à contribution sur des listes de discussion spécialisées pour demander à des concepteurs externes d'utiliser notre fiche d'analyse pour caractériser des composants qu'ils connaissaient et de commenter le choix des caractéristiques. Or, malgré nos appels et les demandes régulières de retours d'expérience sur ces listes de discussion (cf. paragraphe suivant), nous n'avons eu que peu de retours et il n'y a eu que très peu d'analyses qui ont été réalisées. Nous pouvons donc dire que notre ensemble de caractéristiques n'a pas été validé.

Concernant la proposition du catalogue, elle est validée par les demandes régulières de retours d'expérience sur des composants logiciels dans les listes de discussion spécialisées. Dans l'hypothèse où ce catalogue serait disponible sur internet, cela conduirait les concepteurs s'ils ne l'utilisaient pas à :

- faire des recherches pour identifier des composants ;
- demander des retours d'expérience sur des listes de diffusion spécialisées ;
- réutiliser constamment les mêmes composants déjà identifiés lors de projets antérieurs ;
- réaliser des développements spécifiques.

Sa présence au sein de la méthode est validée par cette réalité de la majorité des concepteurs.

3.3.2 Validation par rapport à l'outil

En plus de son utilisation pour le choix d'un composant « gestion de documentation » pour le site web du LAMIH (cf. § 2.5.2 page 124), nous avons utilisé l'outil pour choisir un composant afin de mettre en œuvre la gestion des inscriptions en ligne sur le site web de la journée « CAO-Calcul '08 ». La recherche sur mots-clés dans le catalogue de composants a identifié les trois produits suivants, dont la consultation des fiches de renseignements a permis de constater que :

- « Confman » est développé en Python, il est anglophone, la fonction « inscription » fait partie d'une des nombreuses autres fonctions qui ne sont pas utiles pour l'événement ;
- « LAMIHRegister » est « open source », il est en langue française, il n'apporte pas de fonctions non souhaitées ;
- « START V2 » n'est pas open source, il n'est disponible que sur Windows et Unix, ce qui n'est pas compatible avec l'équipement de l'université, il est en langue anglaise, la fonction « inscription » fait également partie d'une des nombreuses autres fonctions qui ne sont pas utiles pour ce type de site « événement ».

Compte tenu de ces renseignements et du contexte d'utilisation, le choix s'est porté sur « LAMIHRegister ».

Si le catalogue n'avait pas été utilisé, cela aurait conduit à suivre la situation actuelle de la majorité des concepteurs telle que nous l'avons exposé ci-dessus : faire des recherches, demander des retours d'expérience sur des listes de diffusion spécialisées ou réaliser des développements spécifiques.

Ainsi, cette analyse montre que, contrairement à celle des caractéristiques des composants, la présence et l'utilisation du catalogue a été validée. Cependant, il serait souhaitable d'augmenter le nombre de composants qu'il référence afin d'avoir des propositions significatives de choix.

3.4 Fonctionnalité « Accélérer la mise en place d'un site web »

Concernant cette fonctionnalité, les critères de validation sont les suivants :

- capacité à générer automatiquement des structures de données à partir de choix de conception ;
- capacité à définir une architecture logicielle d'accueil afin de faciliter la conception d'un nouveau site web.

Ces objectifs étant orienté implémentation, nos propositions dans la méthode ne peuvent pas être validés. Nous examinons donc uniquement la validation par rapport à l'outil.

L'apport de notre travail par rapport à l'état de l'art est :

- La génération de structures de données au sein du modèle de site web pour la navigation et le contenu ;
- La définition d'une architecture d'accueil.

L'analyse est basée sur l'utilisation de l'outil pour la mise en œuvre des sites web « CAO-Calcul'08 », « GISEH » et « LAMIH_new ».

3.4.1 Générer une structure de données

Pour « GISEH » et « LAMIH_new », l'utilisation du WISDOM Tool a permis de générer les structures de données à partir des choix de conception. En l'absence de l'outil, les structures de données auraient dû être définies manuellement. Cette capacité est donc validée par le gain de temps constaté pour générer rapidement les données en fonction des choix de l'utilisateur.

3.4.2 Faciliter la conception d'un site web

Pour « CAO-Calculs '08⁶¹ », le cahier des charges était

- une énumération d'items (planning, objectif, partenaires, accès) ;
- d'avoir la possibilité de s'inscrire en ligne.

En utilisant une architecture d'accueil dédié au sites web de « journées thématiques AIP-PRIMECA », la conception et l'implémentation de la structure du site a pris 4h30. L'essentiel du travail a été d'adapter :

- le label des items de menu ;
- le contenu des fichiers sans modifier leurs structures ;
- la configuration du composant logiciel « gestion des inscriptions ».

En l'absence de la méthode et de l'outil, des concepteurs web confirmés ont estimé qu'à partir du même cahier des charges, ils auraient passés entre une journée et demi et deux jours pour le concevoir. Le gain de temps est donc significatif car la conception consiste surtout à les adapter au nouveau site web.

Pour le site « LAMIH_new », la conception et l'implémentation a duré l'équivalent d'une semaine de travail. Le délai un peu plus important s'explique par la taille plus importante du site (« LAMIH_new » est un site catalogue nécessitant la conception d'une structure de données plus complexe que pour un site de présence tel que « CAO-Calculs '08 »).

Pour un cahier des charges similaires à celui de « CAO-Calculs '08 », la conception et l'implémentation du site web de « GISEH » ont été réalisées en deux semaines. Le délai plus important par rapport aux deux précédents s'explique par le fait que le concepteur était débutant en conception web, et que l'appropriation d'un framework n'est pas une tâche facile. Compte tenu de ces considérations, nous estimons que ce délai de

⁶¹ <http://www.univ-valenciennes.fr/sp/CAOCalcul/>

conception est relativement rapide, et que l'absence de l'outil aurait entraîné une durée de développement beaucoup plus longue et un site de qualité inférieure.

Ainsi, même si le framework de classe doit être amélioré quant à son appropriation, nous pensons que cette capacité à faciliter la conception d'un site web est validée par ce gain de temps que nous avons constaté.

4 Conclusion

L'objectif de cette partie était d'examiner si les propositions et leurs mises en œuvre répondaient aux objectifs que nous nous étions fixés pour chacune des fonctionnalités énoncées dans le chapitre 1. Nous résumons ci-dessous les conclusions par rapport à chacune des fonctionnalités [Cocquebert, 08b].

Concernant la fonctionnalité « *Assister la réalisation du processus de conception* », le processus défini dans la méthode n'est pas validé car il dépend du contexte de son utilisation (habitudes de développement, taille de l'équipe, domaine d'application). Par contre :

- la prise en compte de la dimension « implémentation » pour modéliser un site web est validée par son lien naturel avec la conception de contenu dans le processus ;
- l'implémentation du processus dans l'outil est validée par sa capacité à avoir guidé un concepteur débutant dans la réalisation d'un site web ;
- la proposition des modèles pour les dimensions de modélisation est validée par l'utilisation du WISDOM Tool pour concevoir des sites web réels.

Quand à l'assistance de l'utilisateur par des patrons de domaine, elle est également validée par leur capacité à avoir guidé le concepteur débutant dans la réalisation d'un site web.

Concernant la fonctionnalité « *Identifier et faciliter la réutilisation de solutions de conception* », elle est validée dans la méthode par les concepts définis pour faciliter la communication dans la conception de la présentation et des contenus, Sa mise en œuvre dans l'outil est également validée de par sa capacité à proposer :

- des solutions de conception et à faciliter leur adaptation par des possibilités interactives ;
- des « architectures d'accueil dédiées » pour des familles de sites web spécifiques.

Concernant la fonctionnalité « *Identifier et faciliter le choix de composants logiciels* », l'identification des caractéristiques des composants n'a pas été validée car nous n'avons pas réussi à avoir des retours d'autres concepteurs. Cependant, la présence du catalogue est validée de par sa capacité à éviter (tout au moins dans un premier temps) de faire des recherches sur internet sur l'existence de composants logiciels et sur leurs caractérisations fonctionnelles et non fonctionnelles.

Concernant la fonctionnalité « *Accélérer la mise en place d'un site web* », elle ne peut pas être validée dans la méthode car elle est orienté implémentation, mais elle est validée dans l'outil de par le gain de temps constaté pour la mise en place des structures de données et des sites web examinés.

Conclusion générale

En partant du constat que les méthodes et outils de conception pour le développement de sites web ne répondaient pas totalement aux attentes des concepteurs de sites en termes d'assistance et de réutilisation, nous avons proposé dans ce mémoire de thèse une méthode d'aide à la conception de site web appelée WISDOM. Cette méthode permet de réaliser les fonctionnalités suivantes :

- guider le concepteur de sites web dans le processus de conception, plus particulièrement dans la réalisation des étapes « conception globale », « conception détaillée » et « implémentation » de ce processus ;
- réutiliser une expérience de conception capitalisée à partir d'une analyse des solutions utilisées dans des sites web en production ;
- faciliter l'identification et le choix de composants logiciels métiers ou techniques à partir d'une capitalisation de leurs caractéristiques fonctionnelle et non fonctionnelles ;
- faciliter le développement du site lui-même en proposant une génération de structures de données en fonction des choix de conception et d'une architecture d'accueil à personnaliser par le concepteur

Une implémentation de cette méthode a été proposée dans ce mémoire sous la forme d'un site web appelé WISDOM Tool. Ce site web implémente :

- le processus de conception afin de guider le concepteur dans le processus de conception et lui apporter une aide à la réalisation des étapes du processus sous la forme de patrons de domaine ;
- une exploitation des analyses de sites web afin de proposer des solutions de conception et faciliter leur adaptation ;
- une exploitation de la caractérisation des composants logiciels
- une génération de structures de données et d'une architecture d'accueil afin de faciliter la mise en œuvre d'un nouveau site web.

Issus de la conception de onze sites web, WISDOM et WISDOM Tool ont été évalués à partir d'une analyse de leur utilisation pour concevoir trois sites web : « GISEH », « CAO-Calculs '08 » et une nouvelle version du site du LAMIH. Focalisée par rapport aux fonctionnalités que nous avons spécifiées et mises en œuvre, cette évaluation a montré que :

- le concepteur était effectivement guidé dans le processus de conception d'un site web et que les patrons de domaine lui apportaient une aide dans la réalisation des étapes de ce processus ;
- les solutions de conception étaient proposées au concepteur pour chacune des dimensions de modélisation, et que celui-ci avait la possibilité de les adapter au site en cours de conception ;
- les composants logiciels métiers ou techniques étaient identifiés à partir de fonctions à mettre en œuvre et évitaient une recherche fastidieuse pour avoir des informations sur les fonctionnalités et retours d'expérience des utilisateurs ;
- la génération de structures de données à partir de choix de conception et d'une architecture d'accueil personnalisable permettaient un gain de temps significatif dans la mise en œuvre d'un nouveau site web, d'autant plus si cette architecture était celle d'une famille de site web.

De cette évaluation, nous en avons conclu que les résultats obtenus sont très encourageants, de par la capacité de WISDOM et du WISDOM Tool à satisfaire notre objectif initial de guider le concepteur dans le processus de conception et de faciliter la réutilisation de solutions de conception et de composants logiciels.

D'un point de vue général, l'originalité de notre approche réside dans le fait de relier le processus de conception avec :

- une formalisation de l'expérience de conception qui s'appuie sur des sites web réels ;
- une caractérisation de composants logiciels qui permet de considérer à la fois des facettes fonctionnelles et des facettes non fonctionnelles.

Cependant, les premiers éléments de validation de cette originalité ont uniquement concerné les fonctionnalités par rapport à la conception des sites de catégorie « site catalogue » et « site de présence ».

Notre première perspective de recherche serait donc d'élargir cette validation :

- aux autres catégories de site web (« sites orientés services » et « systèmes d'information web ») ;
- aux autres catégories de critères de validation en considérant :
 - le site web comme un produit logiciel en évaluant la fiabilité, la maintenabilité, la portabilité, etc.
 - l'utilisation du site web résultat en évaluant l'efficacité de la navigation dans le site web, du respect des critères d'ergonomie dans la conception de la présentation, etc.

D'un point de vue plus technique, notre contribution concerne la mise à disposition d'un catalogue d'analyses de sites web et d'un catalogue de caractérisations fonctionnelle et non fonctionnelle de composants logiciels, et la génération d'une architecture d'accueil.

Concernant les catalogues, notre évaluation de l'outil et de la méthode ont montré qu'ils remplissaient leur rôle auprès du concepteur, mais il serait souhaitable d'augmenter le nombre d'analyses qu'ils contiennent. Pour cela, notre seconde perspective de recherche serait d'automatiser cette caractérisation des sites web et des composants logiciels de la manière suivante :

- par rapport au catalogue d'analyses de site web, cette automatisation consisterait à utiliser des méthodes et outils proposés par le « web sémantique » ;
- par rapport au catalogue de caractérisation des composants, cette automatisation consisterait à consulter les bases de données de sites de références afin d'y extraire les caractérisations fonctionnelles et les forums afin d'en extraire les retours d'expérience des utilisateurs.

Dans les deux cas, cette automatisation permettrait d'augmenter de manière significative le nombre d'analyses, et d'augmenter ainsi la pertinence des propositions de solutions de conception et de composants logiciels.

Concernant l'architecture d'accueil, notre évaluation a montré qu'elle permettait un gain significatif de temps dans la mise en œuvre d'un site web si le concepteur connaissait le framework, mais que sa prise en main par une personne externe au développement n'était pas facile. Ainsi, notre troisième perspective, plus technique, consisterait à améliorer la prise en main du framework par une intégration de composants techniques standards, par un examen des programmes du framework pour améliorer leur lisibilité et par l'écriture d'un tutoriel de prise en main.

D'une manière globale, ces trois perspectives sont un exemple du travail à faire pour améliorer WISDOM et WISDOM Tool en tant que méthode et d'outil d'aide à la conception qui favorise la réutilisation de solutions de conception et de composants logiciels.

Références bibliographiques

- [Bordage 08] Bordage S., Thévenon D., Brousse F., Dupaquier L. *Conduite de projet Web*, Eyrolles, 2008.
- [Borne, 99] Borne I., Revault N., « Comparaison d'outils de mise en œuvre de design patterns », *L'Objet*, 5(2), 1999.
- [Buschmann, 96] Fowler M., *Analysis Patterns - Reusable Object Models*, Addison-Wesley, 1997
- [Cauvet, 01] Cauvet C., *Ingénierie des systèmes d'informations*, Paris, Hermès, 2001.
- [Ceri, 00] Ceri S., Fraternali P., Bongio A., Maurino A., "Modeling data entry and operations in WebML", *WebDB 2000*. Dallas, 2000.
- [Clark, 04] Clark J., Clarke C., De Panfilis S., Granatella G., Predonzani P., Sillitti A., Succi G., Vernazza T., « Selecting components in large COTS repositories », *Journal of Systems and Software*, vol. 73, issue 2, 2004, p. 323-331.
- [Cnkovic, 02] Crnkovic, I.; Larsson, M. « Building Reliable Component-Based Software Systems ». Artech House Publisher., 2002.
- [Coad, 95] Coad P., *Object Models : Strategies, Patterns an Applications*, Yourdon Press Computing Series, 1995.
- [Cocquebert, 07a] Cocquebert E., Trentesaux D., Tahon C. : « Proposition d'un système patron pour la conception de sites web ». *JD-JN-MACS 2007, 9-11 juillet 2007, Reims, France*
- [Cocquebert, 07b] Cocquebert E., Trentesaux D., Tahon C. « Proposition d'un système patron pour la conception de sites web ». *E-Sta*, Vol. 5, pp. 28 (2008). (ce numéro est l'un des deux numéros spéciaux sur les JD-MACS 2007, contenant une sélection de 15 communications sur les 75 qui ont été présentées).
- [Cocquebert, 08a] Cocquebert E., Trentesaux D., Tahon C. : « Conception d'un SIW par réutilisation conceptuelle et logicielle ». *ERTSI 2008, conjoint à INFORSID 2008*.
- [Cocquebert, 08b] Cocquebert E., Trentesaux D., Tahon C., *Méthode d'aide à la conception basée sur la réutilisation conceptuelle et logicielle*, Numéro spécial « Objets, Composants et Modèles dans l'Ingénierie des Systèmes d'Information » de la revue « Ingénierie des systèmes d'information », Vol 3/2008.

- [Conte, 01a] Conte A., Fredj M., Giraudin J.-P., Rieu D., « P-Sigma : un formalisme pour une représentation unifiée de patrons », *INFORSID 2001*, Genève, 29 mai-1^{er} juin 2001, p. 67-86.
- [Conte, 01b] Conte A., Hassine I., Giraudin J.-P., Rieu D., « AGAP : un Atelier de Gestion et d'Application de Patrons », *INFORSID 2001*, Genève, 29 mai-1^{er} juin 2001, p. 142-159.
- [Coplien, 96] Coplien J. O., *Advanced C++ Programming Styles and idioms*, Addison-Wesley, 1992.
- [Coulanges, 96] Coulange B., *Réutilisation du logiciel*, Masson, 1996
- [Couturier, 04] Couturier V., « Patterns de coopération de systèmes d'information », *INFORSID 2004*, Biarritz, 25-28 mai 2004, p. 495-510.
- [De Troyer, 98] 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.
- [Favre, 06a] Favre J.M., Estublier J., Blay-Fornarino M., *L'ingénierie dirigée par les modèles*, Lavoisier, 2006.
- [Favre, 06b] Favre J.-M., Musset J., « Rétro-ingénierie dirigée par les métamodèles », *2^{ème} journée sur l'Ingénierie Dirigée par les Modèles IDM '06*, Lille, 26-28 juin 2006, p. 51-66.
- [Fonte, 97] Front, Agnès. Développement de systèmes d'information à l'aide de patrons. Application aux bases de données actives. Thèse de doctorat, Université Grenoble 1, 1997.
- [Fowler, 97] Fowler M., *Analysis Patterns – Reusable Object Models*, Addison-Wesley, 1997.
- [Gamma, 95] Gamma E., Helm R., Johnson R., Vlissides J. *Design pattern, Elements of Reusable Object-Oriented Software*, Addison Wesley, 1995
- [Gnahou, 00] Gnaho C., Définition d'un cadre méthodologique pour l'Ingénierie des Systèmes d'Information Web Adaptatifs, Thèse de doctorat, Université de Paris I, 2000.
- [Guerrero, 01] Guerrero L. A., Fuller D. A., « A pattern system for the development of collaborative applications », *Information and Software Technology*, vol. 43, issue 7, p. 457-467.
- [Gzara 00] Gzara L., Les patterns pour l'ingénierie des systèmes d'informations produit, Thèse de doctorat, INPG, 2000.
- [Halin, 05] Halin G., « De la conception d'hypermédia à la conception d'application Web », *STICEF*, vol. 12, 2005.
- [Hassine, 02] Hassine I., Rieu D., Bousnaas F., Seghrouchni O., « Symphony : un modèle conceptuel de composants métier ». In C. Cauvet (Ed.), *Connaissances métier en ingénierie des systèmes d'information*, Hermès science, Paris, pp. 35-59, 2002
- [IEC, 99] International Organisation for Standardization (ISO), International Electrotechnical Commission (IEC), Topic Maps, International Standard ISO/IEC 13250:1999, 19 April 1999.
- [Isakowitz, 95] Isakowitz T., Stohr A. & Balasubramanian E., *RMM : A methodology for structured hypermedia design*, Communications of the ACM 38(8), pp. 34-44, August 1995.

- [Jezequel, 99] J-M. Jezequel, S. Lorcy, N. Plouzeau, « Un patron pour la gestion de la qualité de service d'applications réparties », *L'objet : Patrons Orientés Objet*, Hermès, 1999
- [Jrad, 08] Jrad Z., Anli A., Aufaure M.-A. « Modèle contextuel pour la personnalisation », *EGC 2008*, Sophia-Antipolis, 29 janvier- février 2008.
- [Khayati, 05] Khayati O., Modèles formels et outils génériques pour la gestion et la recherche de composants, Thèse de doctorat, Université de Grenoble, 2005.
- [Lang, 02] Lang, M., « Hypermedia Systems Development: Do We Really Need New Methods? », *Proceedings of the Informing Science + IT Education Conference*, Cork, Ireland, June 19-21, pp. 883-891.
- [Le, 01] Le T. L., Rolland C., « Functional matching in COTS-based development context », *INFORSID 2001*, Genève, 29 mai-1^{er} juin 2001, p. 87-110.
- [Lepreux, 05] Lepreux, S.. Approche de Développement centré décideur et à l'aide de patrons de Systèmes Interactifs d'Aide à la Décision. . Thèse de doctorat, Université de Valenciennes, 2005.
- [Leung, 02] Leung K. R. P. H., Leung H. K. N., « On the efficiency of domain-based COTS product selection method », *Information and Software Technology*, vol. 44, issue 12, 2002, Pages 703-715.
- [Licea, 00] Licea, J. Favela, « An extensible platform for the development of synchronous groupware », *Information and Software Technology*, vol. 42, issue 6, 2000, Pages 389-406.
- [Lukosch, 06] Lukosch S., Schümmer T., « Groupware development support with technology patterns », *International Journal of Human-Computer Studies*, vol. 64, issue 7, 2006, Pages 599-610.
- [Meinadier, 98] Meinadier J.-P., « Ingénierie et intégration des systèmes », Hermès, 1998.
- [Michel, 06] Michel R., Système d'informations pour l'évaluation des composants logiciels sur étagère (COTS components), Thèse de doctorat, Université de Pau et des Pays de l'Adour, 2006.
- [Muller, 04] Muller P.-A., Gaertner N., *Modélisation objet avec UML*, Eyrolles, 2004.
- [Muller, 05] Muller P.-A., Studer P., Fondement F., Bezivin J., « Platform independent Web application modeling and development with Netsilon », *Software and System Modeling*, vol. 4, num. 4, 2005, p. 424-442.
- [Oussalah, 99] Oussalah C., *Génie objet*, Hermès, 1999.
- [Parson, 06] Parsons D., Rashid A., Telea A., Speck A., « An Architectural Pattern for Designing Component-Based Application Frameworks », *Software : Practice & Experience*, vol. 36, issue 2, 2006, p. 157-190.
- [Pujalte, 04] Puljate V., Ramadour P., Cauvet C., « Recherche de composants réutilisables : une approche centrée sur l'assistance à l'utilisateur ». *INFORSID 2004*, Biarritz, 25-28 mai 2004, p. 211-227.
- [Rieu, 07] Rieu, D. La recherche en SI : cœur du domaine, Assises du GDR I3, 2007
- [Saidane, 05] Saidane M., Formalisation de Familles d'Architectures Logicielles Coopératives : Démarches, Modèles et Outils, Thèse de Doctorat, 2005.

- [Schmit, 96] Schmidt, Douglas C; Fayad, Mohamed; Johnson, Ralph E. Software patterns. *Communications of the ACM* (1996) **39, 10**
- [Schwabe, 96] Schwabe D., Rossi G., Barbosa S., "Systematic Hypermedia Application Design with OOHDM", *Proceedings of ACM Hypertext*, 1996.
- [Seibel, 05] Seibel J., A version-to-version upgrade decision support model for commercial off-the-shelf productivity applications, PhD Thesis, 2005.
- [Studer, 98] Studer R., Benjamins V.R., Fensel D., "Knowledge engineering: principles and methods", *IEEE Transactions on Data and Knowledge Engineering*, 25(1&2), 1998, pp.161-197.
- [Valetto, 95] Valetto G., Kaiser G.E. "Enveloping Sophisticated Tools into Computer-Aided Software Engineering Environments," 40-48. *Proceedings of 7th IEEE International Workshop on CASE*. Toronto, Ontario, Canada, July 10-14, 1995. Los Alamitos, CA: IEEE Computer Society Press, 1995.
- [Villanova-Oliver, 02] Villanova-Oliver M., Adaptabilité dans les systèmes d'informations sur le web : Modélisation et mise en œuvre de l'accès progressif, Thèse de doctorat, Université de Grenoble, 2002
- [W3C, 99] World Wide Web Consortium, Resource Description Framework (RDF) Model and Syntax Specifications, W3C Recommendation, 22 February 1999.

Annexe 1. Modèle des données d'un site web

Le modèle présenté Figure 104 rassemble sur un seul schéma les modèles des différentes dimensions de modélisations détaillées pages 75 et suivantes.

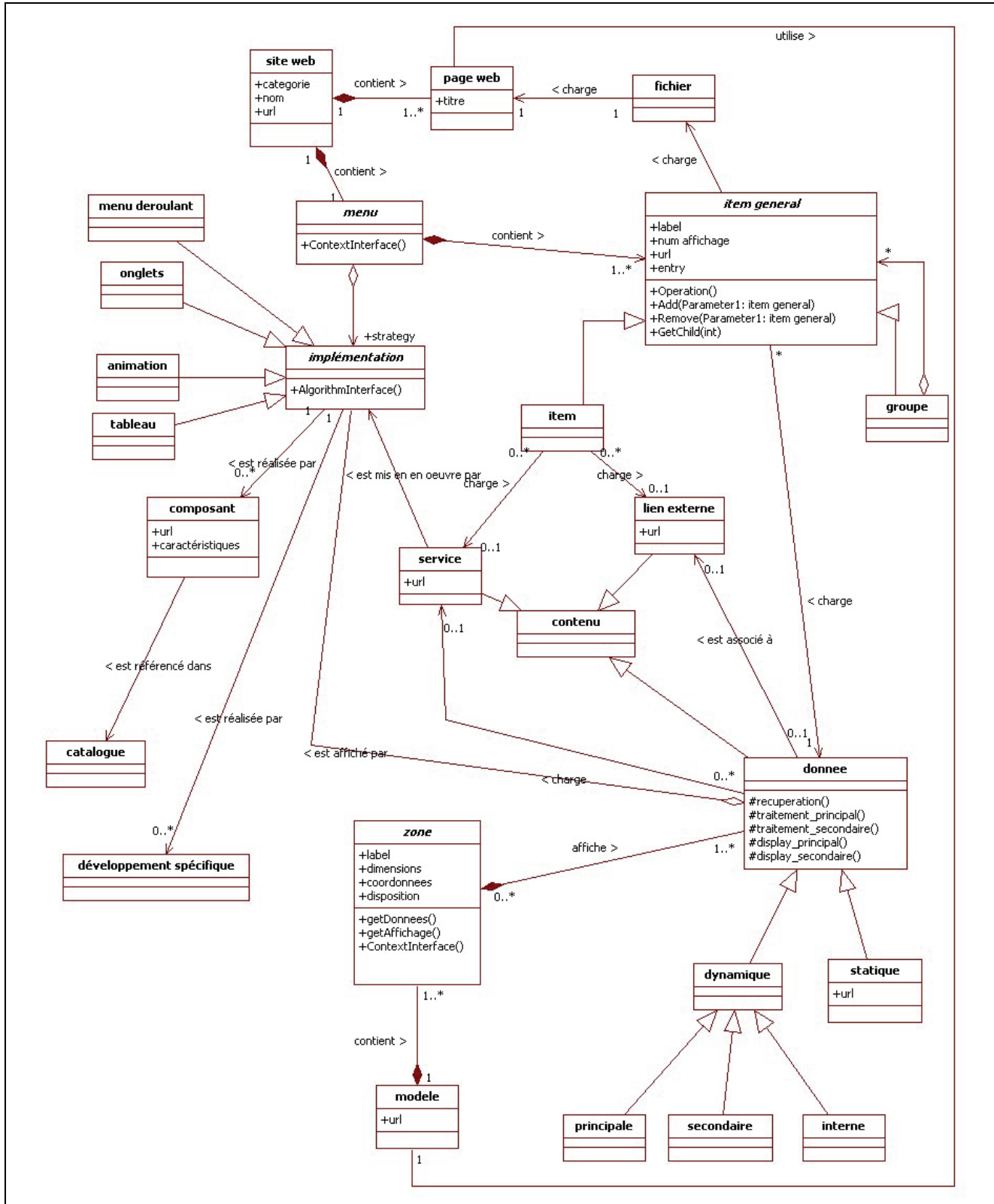


Figure 104 : modèle des données d'un site web

Annexe 2. Formalisme P-Sigma

4.1 Structure générale

Le formalisme P-Sigma est constitué de trois parties [Conte, 01a] :

- « Interface » : elle contient tous les éléments qui permettent la sélection d'un patron ;
- « Réalisation » : elle exprime la solution d'un patron en termes de solution modèle et de solution démarche ;
- « Relation » : elle permet d'organiser les relations entre patrons, donc d'organiser les catalogues de patrons.

Comme le montre la Figure 105 : chacune de ces parties regroupe un ensemble de rubriques, chacune de ces rubriques est composée d'un ou de plusieurs champs typés (de type texte, diagramme UML, expression logique de mots-clés, etc.).

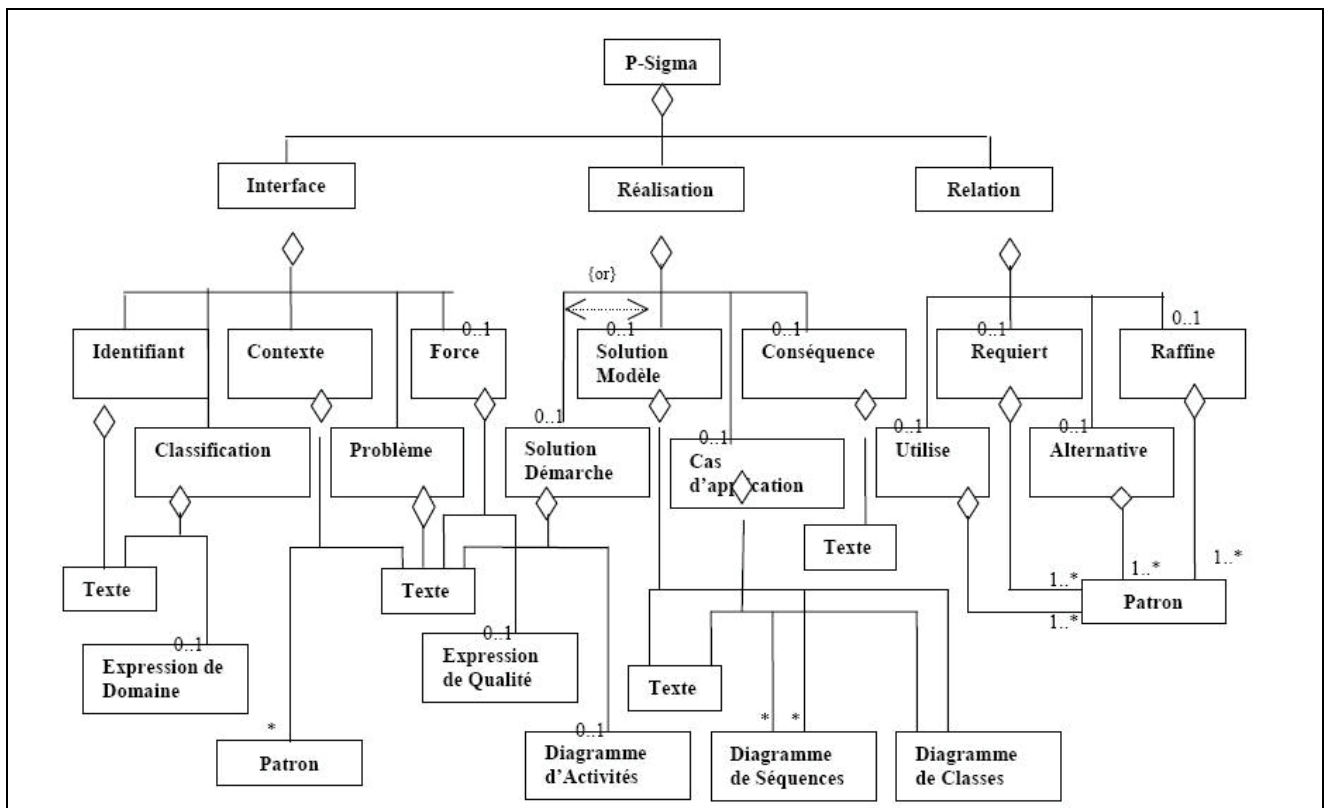


Figure 105 : structure générale de P-Sigma ([Conte, 01a])

4.2 La partie Interface

L'interface d'un patron est composée de cinq rubriques servant à la sélection d'un patron.

Rubriques	Définition	Champs
Identifiant	Définit le couple problème/solution à partir duquel le patron pourra être référencé. Constitue la clé principale de communication entre le concepteur de patrons et les utilisateurs.	– Un champ textuel
Classification	Définit la fonction du patron par un ensemble de mots-clés du domaine (termes du domaine). Donne intuitivement la classification du domaine.	– Un champ textuel – Eventuellement un champ formel (expression logique de mots-clés du domaine)

Contexte	Décrit la pré-condition pour l'application du patron. Peut être obtenu en appliquant la solution modèle d'un ou de plusieurs patrons ; les noms des patrons correspondants constituent alors le champ formel.	<ul style="list-style-type: none"> – Un champ textuel – Eventuellement un champ formel : {Patron}
Problème	Définit le problème résolu par le patron.	– Un champ textuel
Force	Définit les apports induits par l'application du patron.	<ul style="list-style-type: none"> – Un champ textuel – Eventuellement un champ formel : expression logique des critères de qualité associés à une technologie

Tableau 27 : partie « interface » d'un patron selon le du formalisme P-Sigma

4.3 La partie Réalisation

La partie Réalisation d'un patron comprend quatre rubriques présentées dans le tableau 3.

Rubriques	Définition	Champs
Solution, démarche	Indique la solution du problème en terme de processus à suivre. Un diagramme d'activités permet éventuellement de représenter la démarche sous la forme : [garde] Patron.	<ul style="list-style-type: none"> – Un champ textuel – Eventuellement un champ formel de type diagramme d'activités
Solution conseil	Décrit la solution en terme des produits attendus après l'application du patron.	<ul style="list-style-type: none"> – Un champ textuel – Un champ de type diagramme de classes – Eventuellement un champ de type {diagramme de séquence}
Cas d'application	Décrit des exemples d'imitation de la Solution Modèle. Rubrique optionnelle mais fortement conseillée pour faciliter la compréhension de la solution du patron.	<ul style="list-style-type: none"> – Un champ textuel – Un champ de type diagramme de classes – Eventuellement un champ de type {diagramme de séquence}
Conséquences d'application	Présente les limites et les bénéfices de l'application de la solution. Peut inclure un nouvel ensemble de problèmes faisant apparaître la nécessité d'appliquer de nouveaux patrons.	– Un champ textuel

Tableau 28 : partie « réalisation » d'un patron selon le du formalisme P-Sigma

4.4 La partie Relation

La partie Relation est composée de quatre rubriques correspondantes aux quatre types de relations possibles entre les patrons. Dans le formalisme P-Sigma, chaque relation est exprimée par une rubrique mono-champ donnant l'ensemble des patrons qui sont liés au patron en question par la relation concernée (cf Figure 3) : à chaque rubrique est donc associé un seul champ formel exprimé sous la forme {Patrons}. La signification de chacune de ces relations est exprimée dans le Tableau 4 et s'appuie principalement sur les rubriques de la partie Interface.

Rubrique	Définition
Utilise	Si un patron P1 utilise un patron P2, alors : <ul style="list-style-type: none"> – La solution démarche de P1 doit être exprimée en utilisant le patron P2. – La classification de P2 peut être enrichie par rapport à celle de P1 : de nouveaux mot clés sont éventuellement ajoutés dans la classification de P2. – Le contexte de P2 peut être enrichi par rapport à celui de P1.
Raffine	Si un patron P1 raffine un patron P2, alors : <ul style="list-style-type: none"> – Le problème de P1 doit être une spécialisation de celui de P2. – La classification de P1 peut être enrichie par rapport à celle de P2. – La force de P1 peut être enrichie par rapport à celle de P2. – Le contexte de P1 peut être enrichi par rapport à celui de P2.
Requiert	Si un patron P1 requiert un patron P2, alors : <ul style="list-style-type: none"> – L'application de P2 doit être un pré-requis à l'application de P1. – P2 doit apparaître dans le contexte de P1.

Alternative	Un patron P1 est une alternative d'un patron P2 si les deux patrons se différencient par leur force qui justifie deux solutions différentes au même problème : <ul style="list-style-type: none">- P1 et P2 ont la même classification, le même contexte et le même problème.- Seule la rubrique « Force » des deux patrons est différente.
-------------	--

Tableau 29 : partie « relations » d'un patron selon le du formalisme P-Sigma

Annexe 3. Patrons de domaine de WISDOM

Les patrons de domaine sont à utiliser conjointement au WISDOM Tool. Leur objectif est d'aider à utiliser cet outil selon la dimension de modélisation qui les concerne. Le teste fiat donc référence à l'utilisation du WISDOM Tool.

4.5 Patron « navigation »

4.5.1 La partie Interface

Rubriques	Définition
Identifiant	Conception/modification de la navigation
Classification	Navigation d'un site web, items et groupe d'items, famille de site web, dimension « navigation ».
Contexte	Paramètres du contexte de la conception, complétés par : <ul style="list-style-type: none"> – le cahier des charges donné par le client sur les items potentiels, les fonctions à mettre en œuvre et les contenus à afficher sur le site (cas d'une nouvelle conception) ; – le modèle « navigation » de la conception en cours ou issu d'une sauvegarde antérieure (cas de la modification)
Problème	Conception ou modification du menu du site web, afin d'assurer la consultation des informations disponibles ou l'utilisation d'applicatifs spécifiques.
Force	<ul style="list-style-type: none"> – identification et proposition de solutions de conception selon la dimension « navigation » en tenant compte de la valeur de filtrage et de la catégorie du site web ; – gestion de l'interface utilisateur pour en permettre l'adaptation.

4.5.2 La partie Réalisation

Rubriques	Définition
Solution, démarche	<p>Le processus défini par ce patron est présenté Figure 106.</p> <pre> graph TD Start{ } -- menu non défini -.-> 1a(1a. proposition du menu de navigation) Start -- menu déjà défini --> 1b(1b. récupération du menu défini) 1a -.-> MP[menu proposé] 1b -.-> ME[menu existant] MP -.-> 2(2. adaptation du menu de navigation) ME -.-> 2 2 -.-> MA[menu adapté] MA -.-> 3(3. génération des données associées) 3 -.-> SD[structure de données du menu, fichiers associés aux items de menu] </pre>
	<p align="center">Figure 106 : processus du patron navigation</p> <p>Si le menu du site en cours de conception n'est pas encore défini, le processus débute par l'étape 1a ; si le menu est déjà défini, le processus comment à l'étape 1b.</p>

- c. « **Proposition du menu de navigation** » (cf. détails dans Tableau 30) : elle propose l'ensemble des items de menu qui ont été identifiés dans les analyses des sites web de la même catégorie.

Données entrée	en	Cahier des charges sur les items potentiels du site web, catégorie du site web en cours de conception et valeur de filtrage.
Fonction		<ul style="list-style-type: none"> – construction dynamique de la liste des items de menu à partir de l'analyse des menus des sites web selon la dimension « navigation » ; – calcul de l'indice de réutilisation de chacun des items de la liste (l'indice de réutilisation d'un item étant le nombre de fois qu'il a été identifié dans les analyses des sites web).
Donnée sortie	en	Liste des items de menus dont les indices de réutilisation sont supérieurs à la valeur de filtrage, ordonnée de manière décroissante selon les indices de réutilisation.
Etape suivante		Etape 2 pour adapter le menu de navigation.

Tableau 30 : détails pour la proposition de menu de navigation

- d. « **récupération du menu défini** » (cf. détails dans Tableau 31) : elle permet de récupérer les données d'un menu sauvegardé.

Donnée entrée	en	Nom du site web auquel le menu est associé.
Fonction		<ul style="list-style-type: none"> – récupération de l'identificateur du site web ; – récupération des données dans la table associée à l'identificateur.
Donnée sortie	en	Menu existant.
Etape suivante		Etape 2 pour adapter le menu de navigation.

Tableau 31 : détails pour la récupération d'un menu sauvegardé

- Etape 3 « adaptation du menu de navigation »** (cf. détails dans Tableau 32) : elle permet d'adapter au site en cours de conception le menu proposé.

Donnée entrée	en	Liste générée à l'étape 1a ou 1b.
Fonction		Adaptation de la liste à l'aide des fonctions suivantes : <ul style="list-style-type: none"> – sélection et modification éventuelle des items pour le site en conception ; – insertion d'items qui n'ont jamais été définis ; – définition de groupes d'items si nécessaire ; – définition de l'ordre d'affichage des groupes et des items dans un groupe ou dans le menu.
Donnée sortie	en	Menu adapté, sauvegardé dans le modèle de site web.
Etape suivante		Etape 3 pour générer les données d'implémentation en fonction des choix de conception.

Tableau 32 : détails pour l'adaptation du menu de navigation

- Etape 4 « génération des données associées »** (cf. détails dans Tableau 33) : elle permet de générer automatiquement les données d'implémentation en fonction des choix de conception du menu.

Donnée en entrée		Menu adapté.
Fonction		<ul style="list-style-type: none"> – génération de la structure de données d'implémentation du menu ; – génération de l'architecture d'accueil ; – génération des fichiers pour chacun des items auxquels des données sont associées

Données en sortie	<ul style="list-style-type: none"> - structure de données du menu générée en base de données ; - architecture d'accueil générée ; - fichiers des items de menus créés dans l'architecture d'accueil du site web.
Etape suivante	Aucune (fin du processus).

Tableau 33 : détails pour générer des données associées au menu de navigation

Solution conseil

La modélisation associée à la dimension « navigation » est présentée Figure 107.

Selon ce modèle, instancié lors de l'étape 2 « conception/modification de la navigation », un *site web* est caractérisé par sa catégorie, son nom et son adresse internet (url) et contient un *menu*.

Ce menu est défini à l'aide d'une adaptation du patron « composite » [Gamma, 95] afin de représenter la navigation sous la forme d'un arbre sans limiter la profondeur du menu. Cela nous a conduit à relier *menu* à *item général*, classe issue de cette adaptation.

L'implémentation de ce menu est définie à l'aide d'une adaptation du patron « stratégie » [Gamma, 95] afin de la rendre indépendante de la définition du menu. Nous avons ainsi relié *menu* à deux exemples de types d'affichage (*onglets* et *menu déroulant*) à travers *implémentation*, classe issue de cette adaptation.

Le type de *contenu* « donnée » est associé à *item général* afin de pouvoir l'associer à un item ou à un groupe d'item (dans ce dernier cas, pour afficher par exemple sur une seule page les nouveautés des items du groupe). Par contre, les types de *contenu* « service » et « lien externe » sont associés uniquement à *item* car ils y sont spécifiques.

Un *item général* est associé à un *fichier*, définissant une *page web* du *site web*.

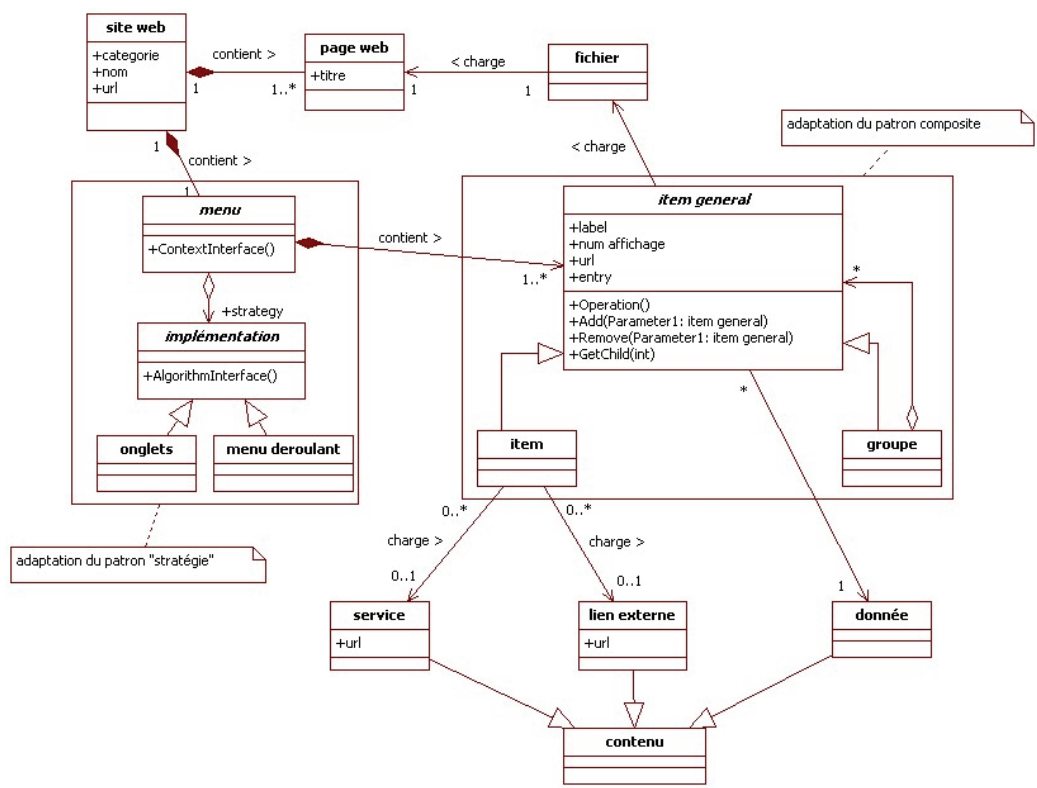


Figure 107 : modèle associé à la dimension « navigation »

Cas d'application

La Figure 108 présente un diagramme de classe pour implémenter un menu. En application de la

	<p>« règle des trois clics⁶² », nous ne considérons qu'un seul niveau de profondeur⁶³.</p> <p>Un <i>groupe_item</i> est défini par :</p> <ul style="list-style-type: none"> – un identifiant ; – un label ; – son adresse internet (url). <p>Un <i>menu_item</i> est défini par :</p> <ul style="list-style-type: none"> – un label ; – son adresse internet (url). <p>Il est attaché à son groupe par <i>id_groupe</i>.</p>
	<pre> classDiagram classe groupe_item { +id_groupe +label +url } classe menu_item { +label +url +id_groupe } groupe_item "1" *-- "1..*" menu_item </pre>
	Figure 108 : diagramme de classe pour générer l'implémentation du menu
Conséquences d'application	Le bénéfice de l'utilisation conjointe du patron et du WISDOM Tool est de concevoir l'application en s'inspirant des navigations sur des sites réels et de générer automatiquement la structure de données à partir de la conception de la navigation.

4.5.3 La partie Relation

L'utilisation ou la définition de ce patron ne dépend pas de celle d'un autre patron.

4.6 Patron « présentation »

4.6.1 La partie Interface

Rubriques	Définition
Identifiant	Conception/modification de la présentation
Classification	Présentation des pages d'un site web, famille de sites web, règles générales de présentation des pages d'un site web, dimension « présentation ».
Contexte	Paramètres du contexte de la conception, complétés par : <ul style="list-style-type: none"> – le cahier des charges fourni par le client sur la technologie support de conception du site web (FLASH, « graphique », « web 2.0 », « standard », etc.) (cas d'une nouvelle conception) ; – le modèle « présentation » de la conception en cours ou issu d'une sauvegarde antérieure (cas de la modification).
Problème	Conception ou modification du modèle de présentation des pages du site web ⁶⁴ .
Force	<ul style="list-style-type: none"> – identification et proposition de solutions de conception selon la dimension « présentation » en tenant compte de la valeur de filtrage et de la catégorie du site web ; – gestion de l'interface utilisateur pour en permettre l'adaptation.

4.6.2 La partie Réalisation

Rubriques	Définition
Solution, démarche	<p>Le processus défini par ce patron est présenté Figure 109.</p> <p>Si le modèle de présentation du site en cours de conception n'est pas encore défini, le processus débute</p>

⁶² « Trois clics maximum pour accéder à une information dans un site web ».

⁶³ 1^{er} click : groupe de menu pour connaître les items ; 2^{ème} click : item de menu pour afficher la page ; 3^{ème} click : sur un lien au sein de la page affichée si elle en dispose.

⁶⁴ Cf. par exemple : <http://www.cs.vu.nl/~martijn/patterns/index.html> et http://www.mit.edu/~jtiddwell/common_ground.html

par l'opération 1a ou 1b ; si le modèle est déjà défini, le processus débute par l'étape 1c.

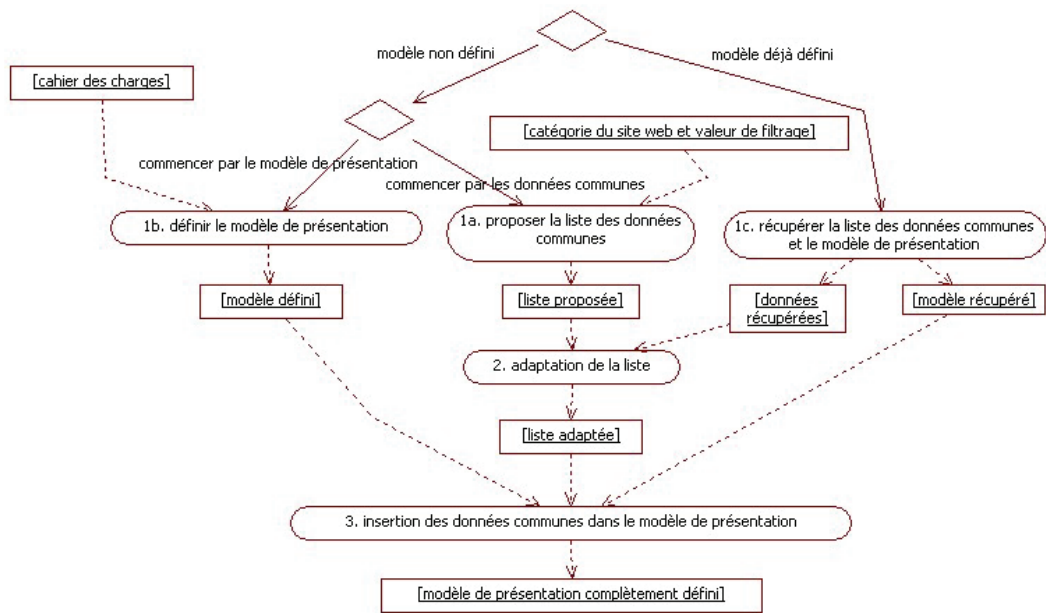


Figure 109 : processus du patron « présentation »

Etape 1

- a. « proposer la liste des données communes » (cf. détails dans Tableau 34) : elle propose la liste des informations communes à toutes les pages.

Données entrée	en	Catégorie du site web en cours de conception et valeur de filtrage.
Fonction		– construction de la liste des informations communes à partir de l'analyse des pages des sites web selon la dimension « présentation » ; – calcul de l'indice de réutilisation de chacun des items de la liste.
Donnée sortie	en	Liste des informations communes dont les indices de réutilisation sont supérieurs à la valeur de filtrage, ordonnée de manière décroissante selon les indices de réutilisation.
Etape suivante		Etape 2 pour adapter la liste des propositions.

Tableau 34 : détails pour proposer des données communes

- b. « définir le modèle de présentation » (cf. détails dans Tableau 35) : elle permet de définir le modèle de présentation des pages.

Donnée entrée	en	Cahier des charges sur le type d'interface utilisateur (Flash, web 2.0, HTML standard, graphique interactif, etc.).
---------------	----	---

Fonction	Définir le modèle de présentation des pages en définissant des zones d'affichage à l'aide d'un éditeur WYSIWYG. Pour faciliter la définition de ce modèle, utiliser les types suivants de zones proposés (cf. Figure 111)) : <ul style="list-style-type: none"> – « zone commune » pour afficher les informations communes à toutes les pages, – « zone accueil » pour afficher un contenu spécifique à la page d'accueil, – « zone navigation » pour afficher le menu défini à l'aide du patron « navigation » (la définition de cette zone peut utiliser les standards <i>de facto</i> de la conception des sites web⁶⁵), – « zone liens » pour afficher un ou plusieurs liens pour accéder rapidement à une page interne, – « zone annonces » pour afficher un rappel d'événements à venir si le site web contient ce type d'information ou pour afficher les modifications du site web ; – « zone contenu » pour afficher le contenu associé à un item du menu. Cette zone est généralement située au centre de la page.
Donnée en sortie	Modèle de présentation défini dans un fichier.
Etape suivante	Etape 3 pour insérer les données communes à toutes les pages dans le modèle.
Tableau 35 : détails pour définir le modèle de présentation	
c. « <i>recupérer la liste des données communes et le modèle de présentation</i> » (cf. détails dans Tableau 36) : elle permet de récupérer les données de la liste adaptée et le modèle de présentation sauvegardé.	
Donnée en entrée	Nom du site web auquel les données sont associées.
Fonction	Les données et le modèle sont récupérés dans la table associée au site web.
Donnée en sortie	Données et modèle récupérés.
Etape suivante	Selon les cas : <ul style="list-style-type: none"> – étape 2 pour adapter la liste des propositions ; – étape 3 pour insérer les données communes dans le modèle de présentation s'il a été modifié.
Tableau 36 : détails pour récupérer la liste des données communes et le modèle de présentation	
Etape 2 « adaptation de la liste » (cf. détails dans Tableau 37) : elle permet d'adapter au site en cours de conception la liste des données communes à toutes les pages.	
Donnée en entrée	Liste proposée des données communes (si l'étape précédente était l'étape 1a), ou liste existante des données communes (si l'étape précédente était l'étape 1c).
Fonction	Adaptation de la liste à l'aide des fonctions suivantes : <ul style="list-style-type: none"> – sélection et modification éventuelle des items qui correspondent au site en cours de conception ; – insertion d'items qui n'ont jamais été définis.
Donnée en sortie	Liste adaptée.
Etape suivante	Etape 3 pour insérer les données communes à toutes les pages dans le modèle.
Tableau 37 : détails pour l'adaptation de la liste	
Etape 3 « <i>insertion des données communes dans le modèle de présentation</i> » (cf. détails dans Tableau 38) : elle permet de finaliser le modèle de présentation de toutes les pages web du site.	
Données en entrée	Liste des données communes et modèle de présentation.
Fonction	Les données communes sont insérées à l'aide de l'éditeur WYSIWYG utilisé pour concevoir le modèle de présentation.

⁶⁵ Par exemple : <http://www.cs.vu.nl/~martijn/patterns/index.html> et http://www.mit.edu/~jtiddwell/common_ground.html

Donnée en sortie	Modèle de présentation finalisé.
Étape suivante	Aucune (fin du processus).

Tableau 38 : détails pour l'insertion de données communes

Solution conseil

La modélisation associée à la dimension « présentation » est présentée Figure 110. Selon ce modèle, – toutes les instances de *page web* sont associées à un *modèle* de présentation ; – un *modèle* est composé d'une ou de plusieurs *zones* d'affichage de données ; – une *zone* affiche une ou plusieurs données ; – Une *donnée* peut être associée à un *lien externe* ou à un *service* (par exemple : « HAL⁶⁶ »).

Afin également de rendre l'affichage du contenu de la zone indépendant de sa définition, nous avons intégré à notre diagramme de classe une autre adaptation du patron « stratégie » [Gamma, 95]. De cette manière, nous avons relié *donnée* à deux exemples d'affichage (*animation* et *tableau*) à travers *implémentation*, classe issue de l'adaptation de ce patron.

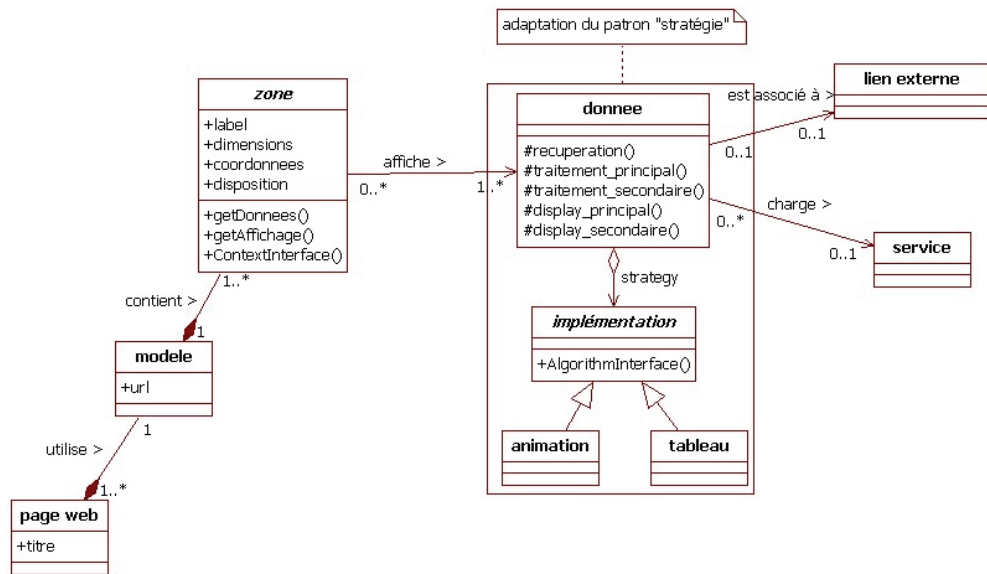


Figure 110 : modèle associé à la dimension « présentation »

Cas d'application

Illustration Figure 111 de différents types de zones dans de présentation d'une page web.



Figure 111 : Exemples de zones sur une page d'accueil (LaPlagne2007)

⁶⁶ <http://hal.archives-ouvertes.fr/>

Conséquences d'application	<p>Le bénéfice de l'utilisation du patron est de disposer du concept de zone et de différents types afin de faciliter la présentation et l'appréhension du problème de la conception de la présentation d'une page.</p> <p>La limite actuelle est de ne pas avoir d'interopérabilité entre le WISDOM Tool et l'éditeur graphique utilisé pour la conception du modèle de présentation et la nécessité d'insérer manuellement les informations communes dans le modèle de présentation.</p>
----------------------------	--

4.6.3 La partie Relation

Rubrique	Définition
Utilise	non applicable
Raffine	non applicable
Requiert	Ce patron nécessite la définition préalable de la navigation du site web (cf. patron « navigation »).
Alternative	non applicable

4.7 Patron « contenu »

4.7.1 La partie Interface

Rubriques	Définition
Identifiant	Conception/modification d'un contenu
Classification	Contenu d'une page web, service, données dynamiques, données statiques, lien internet, dimension « contenu ».
Contexte	Paramètres du contexte, complétés par l'item du menu sans contenu associé (nouvelle conception), ou avec les données du contenu associé (modification de la conception en cours ou reprise d'une conception sauvegardée).
Problème	Associer un contenu à un item du menu ou de modifier ce contenu
Force	<ul style="list-style-type: none"> – identification et proposition de contenu selon la dimension « contenu » en tenant compte de la valeur de filtrage ; – gestion de l'interface utilisateur pour en permettre la conception ou l'adaptation.

4.7.2 La partie Réalisation

Rubriques	Définition
<i>Solution, démarche</i>	<p>Selon les cas, le contenu est un ensemble de données, un lien externe ou un service. Le processus est présenté Figure 112.</p> <p>Le point d'entrée dans le processus est l'item auquel des données doivent être associées. Si des données n'y sont pas encore associées, le processus débute à l'étape 1a ; si des données y sont déjà associées, le processus débute à l'étape 1b.</p>

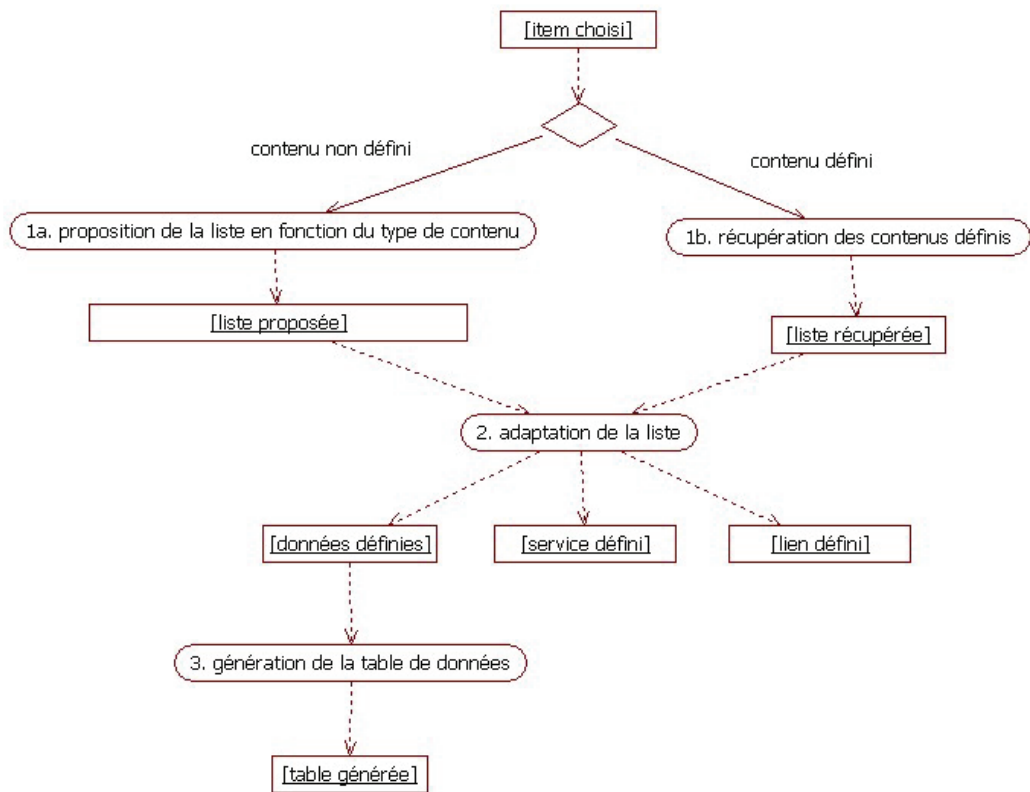


Figure 112 : processus du patron « contenu »

Etape 1

d. « *proposition de la liste des contenus associés* » (cf. détails dans Tableau 39) : elle permet de proposer la liste en fonction du type de contenu choisi.

Donnée en entrée	Nom de l'item concerné par l'étape.
Fonction	<ul style="list-style-type: none"> – construction de la liste en fonction du type de contenu et à partir de l'analyse des pages des sites web selon la dimension « contenu » ; – calcul de l'indice de réutilisation de chacun des items de la liste.
Donnée en sortie	Liste des contenus associés à cet item dont les indices de réutilisation sont supérieurs à la valeur de filtrage, ordonnée de manière décroissante selon les indices de réutilisation.
Etape suivante	Etape 2 pour adapter la liste des propositions.

Tableau 39 : détails pour la proposition de la liste des contenus associés

e. « *récupération des contenus définis* » (cf. détails dans Tableau 40) : elle permet de récupérer la liste des contenus déjà associés à l'item.

Donnée en entrée	Nom de l'item concerné par l'étape.
Fonction	Récupération des contenus dans le modèle de site web.
Donnée en sortie	Contenus récupérés.
Etape suivante	Etape 2 pour adapter la liste des propositions.

Tableau 40 : détails pour la récupération des contenus définis

	<p>Etape 2 « <i>adaptation de la liste</i> » (cf. détails dans Tableau 41) : elle permet d’adapter au site en cours de conception la liste des contenus associés à l’item.</p> <table border="1" data-bbox="335 280 1444 750"> <tr> <td data-bbox="335 280 590 369">Donnée en entrée</td> <td data-bbox="590 280 1444 369">Liste proposée des contenus de l’item (si l’étape précédente était l’étape 1a), ou liste existante des contenus déjà définis (si l’étape précédente était l’étape 1c).</td> </tr> <tr> <td data-bbox="335 369 590 683">Fonction</td> <td data-bbox="590 369 1444 683">Définir les contenus associés (données, service, lien). Dans le premier cas, les données peuvent être définies selon l’un des trois types suivants : <ul style="list-style-type: none"> – donnée statique, qui nécessite d’identifier le fichier correspondant, – donnée dynamique, qui peut être (cf. Figure 115) : <ul style="list-style-type: none"> – une donnée principale si elle différencie les informations entre elles au sein d’une page, – une donnée secondaire si elle est affichée à partir d’une sélection d’une donnée principale et lui est complémentaire, – une donnée interne si elle n’est pas affichée mais est utile à la gestion de l’affichage des deux types précédents. </td> </tr> <tr> <td data-bbox="335 683 590 716">Donnée en sortie</td> <td data-bbox="590 683 1444 716">Liste adaptée.</td> </tr> <tr> <td data-bbox="335 716 590 750">Etape suivante</td> <td data-bbox="590 716 1444 750">Etape 3 pour générer les données en fonction des choix de conception.</td> </tr> </table> <p style="text-align: center;">Tableau 41 : détails pour l’adaptation de la liste</p> <p>Etape 3 « <i>génération de la table de données</i> » (cf. détails dans Tableau 42) : elle permet de générer la table de données en fonction des choix de conception de l’étape précédente.</p> <table border="1" data-bbox="335 896 1444 1120"> <tr> <td data-bbox="335 896 558 963">Donnée en entrée</td> <td data-bbox="558 896 1444 963">Liste adaptée.</td> </tr> <tr> <td data-bbox="335 963 558 1019">Fonction</td> <td data-bbox="558 963 1444 1019">Génération de la table à partir de la liste adaptée, en se basant sur un schéma générique de table de données défini dans le patron.</td> </tr> <tr> <td data-bbox="335 1019 558 1086">Donnée en sortie</td> <td data-bbox="558 1019 1444 1086">Table générée.</td> </tr> <tr> <td data-bbox="335 1086 558 1120">Etape suivante</td> <td data-bbox="558 1086 1444 1120">Aucune (fin du processus).</td> </tr> </table> <p style="text-align: center;">Tableau 42 : détails pour la génération de la table de données</p>	Donnée en entrée	Liste proposée des contenus de l’item (si l’étape précédente était l’étape 1a), ou liste existante des contenus déjà définis (si l’étape précédente était l’étape 1c).	Fonction	Définir les contenus associés (données, service, lien). Dans le premier cas, les données peuvent être définies selon l’un des trois types suivants : <ul style="list-style-type: none"> – donnée statique, qui nécessite d’identifier le fichier correspondant, – donnée dynamique, qui peut être (cf. Figure 115) : <ul style="list-style-type: none"> – une donnée principale si elle différencie les informations entre elles au sein d’une page, – une donnée secondaire si elle est affichée à partir d’une sélection d’une donnée principale et lui est complémentaire, – une donnée interne si elle n’est pas affichée mais est utile à la gestion de l’affichage des deux types précédents. 	Donnée en sortie	Liste adaptée.	Etape suivante	Etape 3 pour générer les données en fonction des choix de conception.	Donnée en entrée	Liste adaptée.	Fonction	Génération de la table à partir de la liste adaptée, en se basant sur un schéma générique de table de données défini dans le patron.	Donnée en sortie	Table générée.	Etape suivante	Aucune (fin du processus).
Donnée en entrée	Liste proposée des contenus de l’item (si l’étape précédente était l’étape 1a), ou liste existante des contenus déjà définis (si l’étape précédente était l’étape 1c).																
Fonction	Définir les contenus associés (données, service, lien). Dans le premier cas, les données peuvent être définies selon l’un des trois types suivants : <ul style="list-style-type: none"> – donnée statique, qui nécessite d’identifier le fichier correspondant, – donnée dynamique, qui peut être (cf. Figure 115) : <ul style="list-style-type: none"> – une donnée principale si elle différencie les informations entre elles au sein d’une page, – une donnée secondaire si elle est affichée à partir d’une sélection d’une donnée principale et lui est complémentaire, – une donnée interne si elle n’est pas affichée mais est utile à la gestion de l’affichage des deux types précédents. 																
Donnée en sortie	Liste adaptée.																
Etape suivante	Etape 3 pour générer les données en fonction des choix de conception.																
Donnée en entrée	Liste adaptée.																
Fonction	Génération de la table à partir de la liste adaptée, en se basant sur un schéma générique de table de données défini dans le patron.																
Donnée en sortie	Table générée.																
Etape suivante	Aucune (fin du processus).																
<p><i>Solution conseil</i></p>	<p>La modélisation associée à cette dimension est présentée Figure 113 : modèle associé à la dimension « contenu ». Selon ce modèle, instancié lors de l’étape 6 « <i>conception/modification des données</i> », un contenu peut être un <i>service</i>, un <i>lien externe</i> ou une <i>donnée</i>.</p> <p>Dans les deux premiers cas, <i>service</i> et <i>lien externe</i> sont définis par une adresse internet (url).</p> <p>Dans le troisième cas, une <i>donnée</i> est <i>statique</i> ou <i>dynamique</i>. Si elle est dynamique, elle est :</p> <ul style="list-style-type: none"> – <i>principale</i> si elle permet de différencier un enregistrement dans une table ; – <i>secondaire</i> si elle précise la définition d’une donnée principale ; – <i>interne</i> si elle n’est pas affichée. <p>Les méthodes de gestion de ces données (récupération, traitement et affichage des données principales et secondaires) sont définies de manière protégée dans la classe <i>donnée</i> afin de :</p> <ul style="list-style-type: none"> – les appeler de manière générique dans un fichier modèle associé aux items ; – définir le corps de ces méthodes dans les classes 																

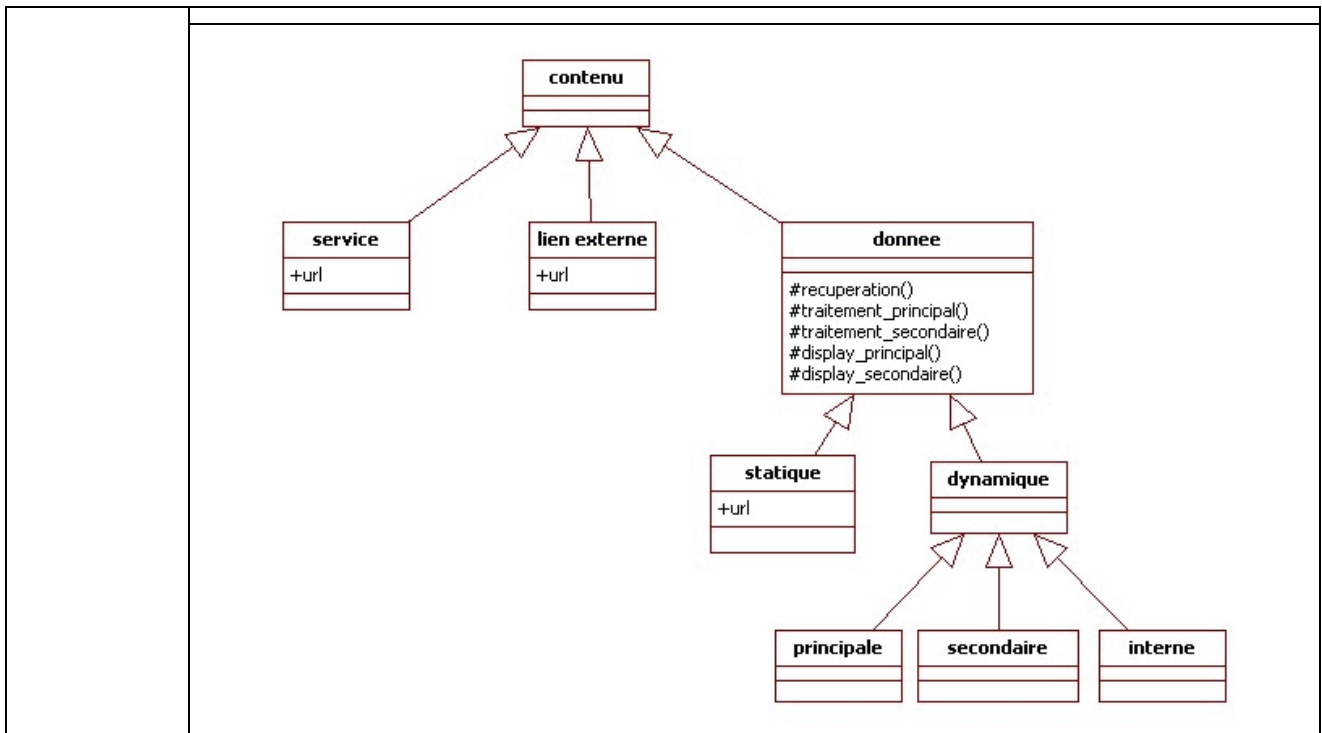


Figure 113 : modèle associé à la dimension « contenu »

Cas d'application

Illustration Figure 113 du concept de « donnée principale » et de « donnée secondaire ».

Figure 114 : Exemple de types de données sur un extrait de la page « conférence » du GDR MACS

Conséquences d'application

Le bénéfice de l'utilisation est de définir le type de contenu et de concevoir la structure de donnée associée en s'inspirant des analyses de sites web réels.

4.7.3 La partie Relation

Ce patron n'a pas de relations avec un autre patron.

Rubrique	Définition
Utilise	non applicable
Raffine	non applicable
Requiert	Ce patron nécessite la définition préalable de la navigation du site web (cf. patron « navigation ») et de définir la présentation des pages du site web (cf. patron « présentation »).
Alternative	non applicable

4.8 Patron « implémentation »

4.8.1 La partie Interface

Rubriques	Définition
Identifiant	Définition/modification d'une implémentation
Classification	Implémentation de l'affichage d'un menu,, de l'affichage de données ou d'un service, aide au choix de composants techniques ou métiers, dimension « implémentation ».
Contexte	Paramètres du contexte et l'élément à implémenter (menu, données, service) dans le cas de la définition d'une implémentation, complétés par les données sur l'implémentation de l'élément du site en cours de conception ou issu d'une conception sauvegardée dans le cas de la modification d'une implémentation.
Problème	Fait suite à l'étape 3 pour implémenter un menu, ou à l'étape 6 pour implémenter un affichage de données ou d'un service. Il permet de réaliser ou de modifier le choix d'un composant métier pour implémenter un service, ou d'un composant technique pour implémenter un menu ou l'affichage de données.
Force	<ul style="list-style-type: none"> - identification et proposition de composants techniques ou métiers, - gestion de l'interface utilisateur pour les proposer à l'utilisateur

4.8.2 La partie Réalisation

Rubriques	Définition		
Solution, démarche	<p>Le processus défini par ce patron est présenté Figure 115.</p> <div style="text-align: center;"> <pre> graph TD A["[fonction à implémenter]"] -.-> B("1. identifier les composants") B -.-> C["[liste identifiée]"] C -.-> D{ } D -- "il y a des composants dans le catalogue" --> E("2a. choisir un composant") D -- "il n'y a pas de composants dans le catalogue" --> F("2b. rechercher ou réaliser un développement spécifique") E -.-> G["[composant choisi, ou implémentation réalisée]"] F -.-> G </pre> </div> <p>Figure 115 : processus du patron « identification du type d'implémentation »</p> <p>Le processus débute à l'étape 1.</p> <p>Etape 1 « identifier les composants » (cf. détails dans Tableau 43) : elle permet d'identifier un composant à partir du catalogue en fonction du besoin à remplir (implémentation d'un menu, affichage de données, mise en œuvre d'un service).</p> <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td style="width: 25%;">Donnée en entrée</td> <td>Fonction à implémenter (« affichage d'un menu », « affichage de données », service à implémenter).</td> </tr> </table>	Donnée en entrée	Fonction à implémenter (« affichage d'un menu », « affichage de données », service à implémenter).
Donnée en entrée	Fonction à implémenter (« affichage d'un menu », « affichage de données », service à implémenter).		

Fonction	Identification des composants dans le catalogue à partir de la fonction « affichage d'un menu », de la fonction « affichage de données », ou de la fonction du service à implémenter.
Donnée en sortie	Résultat de la recherche.
Etape suivante	S'il y a des composants dans le catalogue, l'étape suivante est l'étape 2a ; s'il n'y a pas de composant dans le catalogue pour mettre en œuvre la fonction recherchée, l'étape suivante est l'étape 2b.

Tableau 43 : détails pour identifier les composants

Etape 2

f. « choisir un composant » (cf. détails dans Tableau 44) : elle permet de choisir un composant parmi la liste proposée.

Donnée entrée	en	Résultat de la recherche de l'étape précédente.
Fonction		Mise à disposition de possibilités interactives pour aider à choisir un composant à partir de leur caractérisation dans le catalogue.
Donnée sortie	en	Composant choisi.
Etape suivante		Aucune (fin du processus).

Tableau 44 : détails pour choisir un composant

g. « rechercher ou réaliser un développement spécifique » (cf. détails dans Tableau 45) : elle permet de trouver une solution pour palier l'absence de composant dans le catalogue.

Donnée entrée	en	Aucune, car la recherche est à la charge du concepteur.
Fonction		Recherche d'un composant, laissé à l'initiative de l'utilisateur (moteur de recherche pour la recherche sur internet ; outil de modélisation et de développement pour le développement spécifique).
Donnée sortie	en	Implémentation réalisée.
Etape suivante		Aucune (fin du processus).

Tableau 45 : détails pour rechercher ou réaliser un développement spécifique

Solution conseil

La modélisation de cette dimension est présentée Figure 116: modélisation selon la dimension « implémentation ». Selon ce modèle qui est instancié lors de l'étape 4 « définition/modification de l'implémentation », une *implémentation* affiche une *donnée* (dans une zone) et met en œuvre un *service*. Une *implémentation* est réalisée par un *composant* référencé dans un *catalogue* ou par un *développement spécifique*.

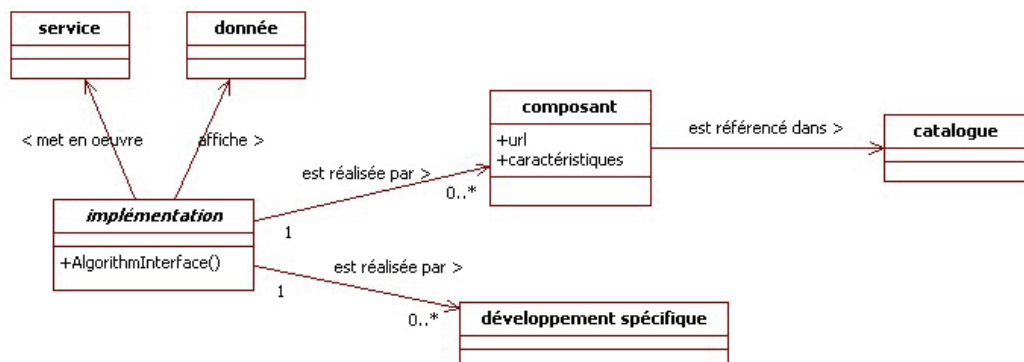


Figure 116: modélisation selon la dimension « implémentation »

Cas

Décrit des exemples d'imitation de la Solution Modèle. Rubrique optionnelle mais fortement

d'application	conseillée pour faciliter la compréhension de la solution du patron.
Conséquences d'application	Le bénéfice de l'utilisation de ce patron est de disposer de conseil sur le choix de l'implémentation d'un menu, de d'affichage de données ou d'un service. Ces conseils sont issus des fiches de renseignements des composants techniques et métiers du catalogue de composants logiciels.

4.8.3 La partie Relation

Rubrique	Définition
Utilise	non applicable
Raffine	non applicable
Requiert	Ce patron fait suite au patron « présentation » pour implémenter un menu, ou à l'étape « contenu » pour implémenter un affichage de données ou d'un service.
Alternative	non applicable

5 Génération de l'architecture d'accueil

Elle a été mise en œuvre sous la forme d'un dossier autonome dont les fichiers sont développés en PHP 4. Cette architecture contient entre autres :

- différents dossiers afin de répartir :
 - les images,
 - les fichiers communs relatifs aux classes publiques du framework, aux composants logiciels et aux librairies intégrés dans l'architecture,
 - les documents ;
 - les méthodes de génération automatique de données pour les étapes « conception de la navigation » et « conception de contenus » ;
- un fichier de configuration générale pour définir les chemins des différentes méthodes, classes, composants et librairies intégrés dans l'architecture ;
- un fichier de démarrage pour coder un nouvel item qui est intégré à l'architecture, et offrant ainsi le bénéfice de toutes les méthodes, fonctions, etc. ;
- un fichier modèle de pages HTML à partir duquel toutes les pages web seront affichées.

Annexe 4. Norme ISO 9126

Cette norme pose les caractéristiques qualités principales d'un produit logiciel et organise leur classification afin d'en faciliter son utilisation lors de processus d'évaluation logiciel. Elle a été réalisée par le comité technique JTC 1 de l'ISO/CEI. Les évolutions de cette norme ont été intégrées dans la norme SquaRE (Software product Quality Requirement and Evaluation), ou ISO 25000. Les caractéristiques décrites s'appliquent au logiciel (progiciels, développements spécifiques, firmware...) que l'on doit considérer comme un système de transformation avec des données d'entrée et des données de sortie. Elles sont représentées Figure 117⁶⁷.

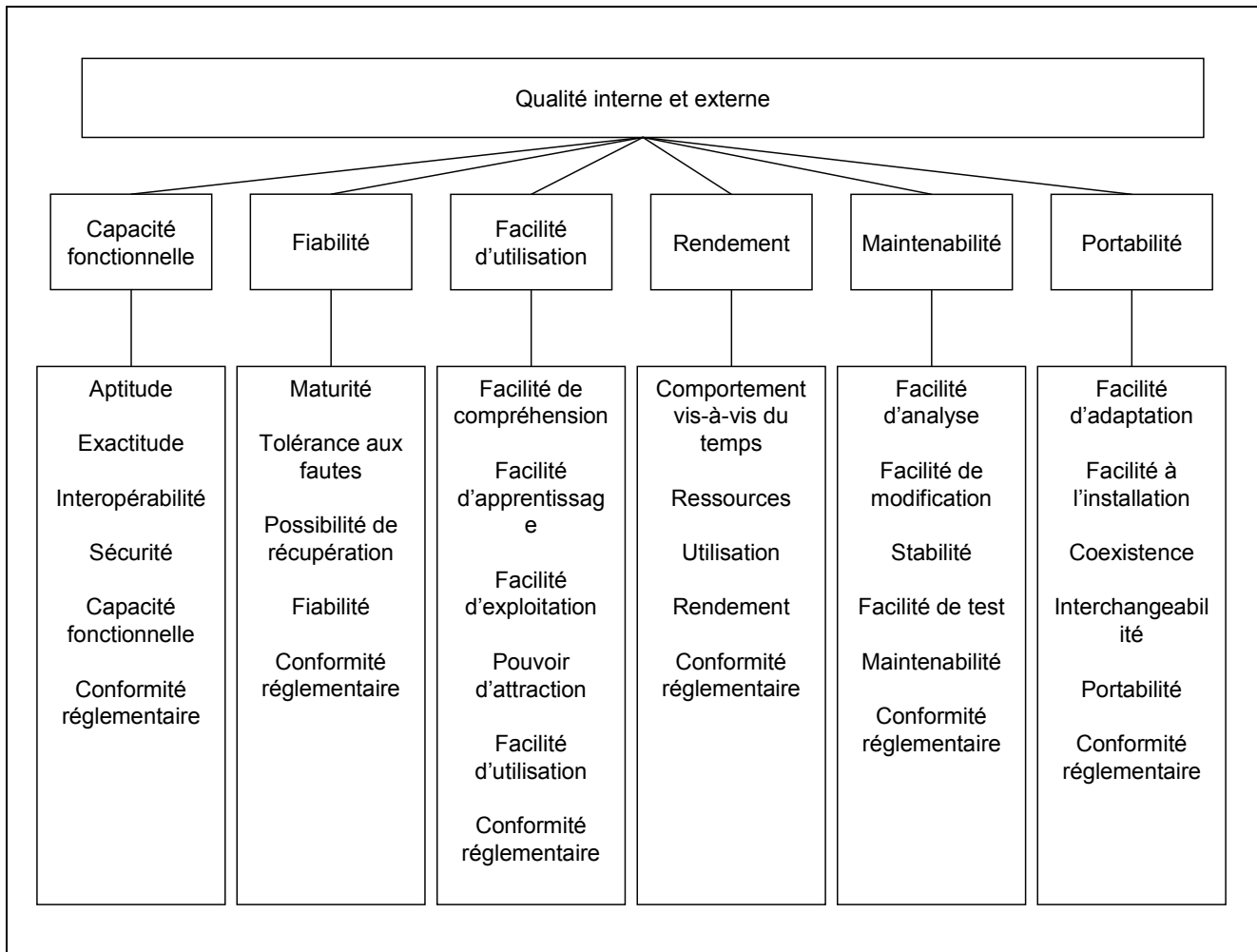


Figure 117 : rubriques de la norme ISO 9126

⁶⁷: http://lil.univ-littoral.fr/~oumoumsack/qualite/ISO9126_3.html et <http://membres.lycos.fr/benoitouellet/La%20norme%20ISO%209126.doc>

Annexe 5. page d'accueil de CAO-Calculs '08

The image shows the homepage of the CAO-Calculs '08 website. At the top left, there are logos for ISTA, AFM, and aip oriméca. The main title is 'Journée Liaison CAO-Calcul' in blue. Below this, the date 'Jeudi 22 mai 2007' is displayed. The location is 'ENSIAME Université de Valenciennes Le Mont Houy 59313 Valenciennes CEDEX 9'. There are links for 'Présentation', 'Programme', 'Inscription', 'Organisation', 'Accès', and 'Contacts'. Below these links, there are links for 'Présentations de la journée', 'Photos de la journées', and '47 inscriptions au 30-09-2008'. At the bottom, there are logos for UVHC, ENSIAME, and EVS CACHAN. A small 'ESTAT'PERSONS' logo is in the bottom left corner. The copyright notice at the bottom center reads '© LAMIH-SP - Crédits et mentions légales'.

ISTA
AFM
aip oriméca

Journée Liaison CAO-Calcul

Jeudi 22 mai 2007

ENSIAME
Université de Valenciennes
Le Mont Houy
59313 Valenciennes CEDEX 9

Présentations de la journée
Photos de la journées
47 inscriptions au 30-09-2008

UVHC
LAMIH
ENS I
CACHAN
EVS
CACHAN

ESTAT'PERSONS

© LAMIH-SP - Crédits et mentions légales

Figure 118 : page d'accueil du site « CAO-Calculs '08 »

Annexe 6. page d'accueil de GISEH



Figure 119 : page d'accueil du site « GISEH »

Résumé

Depuis leur apparition en 1996, la nature et les acteurs d'un site web ont considérablement évolués : les premiers sites diffusaient de l'information statique et étaient réalisés de manière artisanale par peu d'acteurs, alors que désormais, ils sont devenus un vecteur privilégié d'informations dynamiques (sites marchands, institutionnels, de marques, etc.), de dialogue entre les utilisateurs (sites communautaires, intranet, collaboratifs, etc.), et sont réalisés par des équipes projets faisant intervenir de multiples compétences (codeurs, ergonomes, référenceurs, chefs de projet, graphistes, etc.). Du point de vue de la réalisation elle-même, cette évolution a conduit les concepteurs à faire face à un ensemble de critères de plus en plus contraignants pour répondre de manière toujours plus rapide à la mise en œuvre de fonctionnalités interactives évoluées.

Pour éviter une augmentation de coût liée à celle du temps de développement et des exigences de qualité, de sécurité, etc., un nombre élevé de travaux ont été menés et ont proposé des outils et méthodes d'aide à la conception à caractère académique, commercial ou issus du monde libre. Cependant, notre analyse de l'état de l'art montre que ces méthodes et outils de l'ingénierie du web, de la réutilisation ou centrés sur les modèles ne répondent pas totalement aux attentes des concepteurs.

Spécifiée à la fois à partir de l'état de l'art et de notre expérience de conception de sites web, nous proposons la méthode WISDOM, dont les objectifs sont de guider le concepteur dans le processus de conception d'un site web, de formaliser et de réutiliser une expérience de conception issue de l'analyse de sites web en production, de caractériser des composants logiciels par des informations fonctionnelles et non fonctionnelles pour aider à les choisir, et de proposer une architecture logicielle d'accueil pour faciliter le développement du site lui-même. Implémentée au sein du WISDOM Tool, cette méthode a été utilisée pour concevoir la nouvelle version du site web du LAMIH, mais sa validation et celle de l'outil sont examinés par rapport aux nombreux sites web pour la conception desquels (et à partir desquels) la méthode a été conçue et utilisée.