



Une approche épiphyte pour la conception de systèmes conseillers

Bruno Richard

► To cite this version:

Bruno Richard. Une approche épiphyte pour la conception de systèmes conseillers. Autre [cs.OH]. Université du Maine, 2008. Français. NNT : . tel-00491735

HAL Id: tel-00491735

<https://theses.hal.science/tel-00491735>

Submitted on 14 Jun 2010

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.

Une approche épiphyte pour la conception de systèmes conseillers

THÈSE

soutenue le 2 avril 2008

pour l'obtention du titre de

Docteur de l'Université du Maine
(spécialité informatique)

par

Bruno RICHARD

Composition du jury

<i>Rapporteurs :</i>	M. Michel Crampes	Ecole des Mines d'Ales
	M. Serge Garlatti	ENST Bretagne, Brest
<i>Examineurs :</i>	Mme Nathalie Aussenac-Gilles	IRIT, Université Paul Sabatier, Toulouse
	M. Gilles Kassel	LARIA, Université d'Amiens
<i>Directeur :</i>	M. Pierre Tchounikine	LIUM, Université du Maine, Le Mans

Laboratoire d'Informatique de l'Université du Maine



Remerciements

Mes remerciements vont, en premier lieu, à M. Pierre Tchounikine sans qui je ne serais sans doute pas docteur à ce jour. Un grand merci pour m'avoir incité à me lancer dans cette aventure puis pour ses conseils, ses relectures, sa patience et ses encouragements.

Je remercie MM. Crampes et Garlatti qui m'ont fait l'honneur d'être les rapporteurs de cette thèse. Merci également à Mme Aussenac et M. Kassel d'avoir accepté d'être membres du jury de ma soutenance.

Je remercie les membres du LIUM.

Enfin, merci à ma femme, à mes parents, à ma famille et à mes amis pour leurs encouragements constants durant ces cinq années.

Table des matières

1	Introduction	12
1.1	Contexte général	12
1.1.1	Difficultés des utilisateurs du Web	12
1.1.2	Conséquences	14
1.2	Problématique et Méthodologie	14
1.3	Travail réalisé	15
1.4	Organisation du mémoire	17
2	Les systèmes conseillers et systèmes voisins	20
2.1	Définition Générale	20
2.1.1	Définition	20
2.1.2	Caractéristiques	21
2.2	Dimension 1 : types d'aide proposés	25
2.2.1	Apporter directement de l'information supplémentaire à l'utilisateur	26
2.2.2	Simplifier la navigation de l'utilisateur	27
2.2.3	Guider l'utilisateur	27
2.3	Dimension 2 : techniques de présentation des informations/conseils	28
2.3.1	Afficher une page adjacente	29
2.3.2	Ajouter/modifier/supprimer des liens au sein d'une page .	29
2.3.3	Adapter le contenu d'une page	30
2.4	Dimension 3 : techniques de gestion des données manipulées . . .	30
2.4.1	Utilisation de la sémantique des informations contenues dans les pages	31
2.4.1.1	“Content based comparison”	31
2.4.1.2	“Knowledge based recommendations”	34
2.4.1.3	Création de modèles cognitifs	36
2.4.2	Utilisation d'informations relatives au comportement des utilisateurs	38
2.4.2.1	Techniques de “collaborative filtering”	39
2.4.2.2	“Web usage mining”	42

2.4.3	Utilisation de méthodes mixtes	44
2.5	Synthèse	47
3	Problématique	49
3.1	Présentation	49
3.1.1	Contexte	49
3.1.2	Problématique détaillée	50
3.1.3	Spectre des travaux	55
3.2	Domaines scientifiques abordés	55
3.2.1	Domaines scientifiques auxquels nous avons contribué . .	55
3.2.2	Domaines scientifiques en relation avec nos travaux . . .	56
3.2.3	Domaines scientifiques dans lesquels nous avons puisé des techniques et/ou dont nous nous sommes inspirés . . .	56
3.3	Problématique et état de l'art	58
3.3.1	Apporter directement de l'information supplémentaire à l'utilisateur	58
3.3.2	Simplifier la navigation de l'utilisateur	60
3.3.3	Guider l'utilisateur	61
3.4	Déroulement de la recherche	62
3.5	Terrains applicatifs	62
3.5.1	Educasource	63
3.5.2	Apache	63
3.6	Synthèse	63
4	Approche générale proposée : une architecture épiphyte prenant en compte la tâche de l'utilisateur	66
4.1	Principes généraux	66
4.1.1	Définitions	67
4.1.2	Une architecture épiphyte	75
4.1.3	Des modèles de tâches	80
4.1.3.1	Modéliser la tâche	80
4.1.3.2	Identifier les tâches considérées par le système conseiller	80
4.1.4	Des modèles d'utilisation	81
4.1.4.1	Création des conseils	81
4.1.4.2	Utilisation des modèles	84
4.2	Processus d'élaboration d'un système conseiller	85
4.2.1	Création/adaptation/amélioration d'un système conseiller générique	86
4.2.2	Identification et création des modèles de tâches	87
4.2.3	Création des conseils	88

4.2.4	Positionnement des conseils sur les modèles	89
4.2.5	Instanciation du système conseiller générique	89
4.2.6	Analyse de l'utilisation et de l'efficacité du système conseiller et évolutions	89
4.3	Synthèse	90
5	Présentation détaillée de l'architecture proposée : le système conseiller générique Marco	91
5.1	Introduction	91
5.2	Présentation générale	94
5.2.1	Modules et notions	96
5.2.2	Données manipulées	99
5.2.3	Exemple de fonctionnement du système conseiller	100
5.3	Détails de l'architecture technique	102
5.3.1	Noyau du système de conseiller	102
5.3.2	Module réseau	103
5.3.3	Module HTTP	107
5.3.4	Module Proxy	112
5.3.5	Serveur Web et cache	114
5.3.6	Module de diagnostic	115
5.3.7	Module d'élaboration des conseils	116
5.4	Synthèse	118
6	Assister la création des modèles d'utilisation	119
6.1	Assister l'identification et la création des modèles de tâches (étape 1)	120
6.1.1	Assister l'analyse des usages du site (étape 1.a)	120
6.1.2	Assister la création des modèles de tâches (étape 1.c) . . .	123
6.1.2.1	Approche 1 - Utilisation de Protégé 2000	124
6.1.2.2	Approche 2 - Environnement ad hoc	125
6.2	Assister la création des conseils (étape 2)	131
6.2.1	Approche 1 - Utilisation de Protégé 2000	131
6.2.2	Approche 2 - Environnement ad hoc	133
6.2.2.1	Principe	133
6.2.2.2	Définitions	135
6.3	Assister le positionnement des instances de conseil sur les arcs du modèle de tâches (étape 3)	142
6.3.1	Approche 1 - Utilisation de Protégé 2000	142
6.3.2	Approche 2 - Environnement ad hoc	143
6.4	Assister l'instanciation d'un système conseiller générique (étape 5)	144
6.4.1	Approche 1 - Utilisation de Protégé 2000	144

6.4.2	Approche 2 - Environnement ad hoc	144
6.4.3	Définition des propriétés des modèles	144
6.5	Synthèse	147
7	Mise en oeuvre	148
7.1	Premier terrain applicatif : Educasource	149
7.1.1	Présentation du site	149
7.1.2	Analyse du site (étape 1.a)	152
7.1.3	Identification des tâches (étape 1.b)	153
7.1.4	Création des modèles de tâches (étape 1.c)	154
7.1.5	Création des conseils (étape 2)	155
7.1.6	Positionnement des conseils sur les modèles de tâches (étape 3)	156
7.1.7	Instanciation d'un système conseiller générique (étape 4) .	156
7.2	Second terrain applicatif : Apache	159
7.2.1	Présentation du site	159
7.2.2	Identification et création des modèles de tâches (étape 1) .	159
7.2.2.1	Analyse du site (étape 1.a)	159
7.2.2.2	Identification des tâches (étape 1.b)	163
7.2.2.3	Création des modèles de tâches (étape 1.c) . . .	164
7.2.3	Création des conseils (étape 2)	166
7.2.4	Positionnement des conseils sur les modèles de tâches (étape 3)	167
7.2.5	Instanciation d'un système conseiller générique (étape 4) .	167
8	Conclusions et perspectives	171
8.1	Bilan des travaux	171
8.2	Positionnement des travaux	172
8.3	Limites	174
8.4	Perspectives	175
A	Notre approche vue sous l'angle de l'approche méthodologique Platon- 1 [TBB⁺04]	177
B	Pré-traitements effectués sur les logs	181
C	Exemples de statistiques effectuées sur les logs	199
D	Exemple de parcours du site Educasource et de fonctionnement du système conseiller	204

Table des figures

3.1	Exemple de modèle d'utilisation	54
4.1	Exemple de modèle de tâches (l'ordre des transitions est indiqué par les chiffres). En second plan, un exemple de graphe de site. . .	68
4.2	Exemple de conseil	70
4.3	Exemple de conseil	71
4.4	Exemple de modèle d'utilisation.	72
4.5	Fusion de noeuds du modèle d'utilisation.	73
4.6	Exemple de conseil nécessitant l'ouverture d'une page spécifique. . .	74
4.7	Schéma simplifié de l'architecture proposée.	77
4.8	Extension d'un site Web minimaliste par ajout d'un système conseiller . .	79
4.9	Exemple de modèle d'utilisation	83
5.1	Schéma général de fonctionnement du système conseiller générique Marco	95
5.2	Exemples de modifications à la volée effectuées par le système conseiller	110
5.3	Modifications d'une page HTML permettant l'ouverture d'une fenêtre conseil	113
6.1	Les classes génériques	124
6.2	Gestion de modèles d'utilisation avec Protégé 2000	126
6.3	Création manuelle d'un noeud	126
6.4	Les différents noeuds d'un modèle d'utilisation	127
6.5	Création d'un noeud par simple "Drag and Drop"	129
6.6	Exploration automatique du site cible	130
6.7	Exemple de conseil créé avec Protégé 2000	133
6.8	Instanciations des conseils élémentaires	134
6.9	Élaboration d'un conseil	136
6.10	Conseils élémentaires, conseils génériques et instances	137
6.11	Créer un conseil - 1	139
6.12	Créer un conseil - 2	140

6.13	Créer un conseil - 3	141
6.14	Positionnement d'un conseil sur un arc avec Protégé 2000	142
6.15	Positionnement d'un conseil sur un arc avec l'environnement ad hoc	143
6.16	Extrait de la page de garde du système conseiller	146
7.1	Page d'accueil du site Educasource	150
7.2	Exemple de descripteur proposé par le site Educasource	151
7.3	Le modèle de tâches construit pour le site Educasource et correspondant à la tâche "Utilisation des moteurs de recherche".	154
7.4	Exemple de conseil proposé par le système conseiller	157
7.5	Exemple de conseil proposé par le système conseiller	158
7.6	Navigations entre les projets WWW, HTTPD, Jakarta, Cocoon, Lenya et Forrest	161
7.7	Sortie proposée par l'outil de recherche des navigations du type "flimsy navigation"	163
7.8	Copie d'écran partielle du modèle "Utilisation des outils de la fondation Apache pour publier sur le Web"	165
7.9	Exemple de conseils proposés à un utilisateur du modèle "Je veux publier sur le Web"	168
7.10	Exemple de conseils proposés à un utilisateur du modèle "Je veux publier sur le Web"	169
C.1	Sites utilisés comme entrée	200
C.2	Sites parcourus ensemble lors d'une même session	201
C.3	Sites parcourus après www.apache.org	202
C.4	Sites parcourus avant cocoon.apache.org	203

Liste des tableaux

1.1	Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Résultat de la recherche	18
1.2	Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Cycle de vie de la recherche	19
2.1	Systèmes conseillers, aide apportée et techniques utilisées	48
3.1	Notre problématique vue sous l'angle de l'approche méthodolo- gique Platon-1 - Définition du projet de recherche	64
3.2	Notre problématique vue sous l'angle de l'approche méthodolo- gique Platon-1 - Cadre théorique de la recherche	65
8.1	Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Résultat de la recherche	173
A.1	Notre problématique vue sous l'angle de l'approche méthodolo- gique Platon-1 - Définition du projet de recherche	178
A.2	Notre problématique vue sous l'angle de l'approche méthodolo- gique Platon-1 - Cadre théorique de la recherche	178
A.3	Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Résultat de la recherche	179
A.4	Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Cycle de vie de la recherche	180

Liste des algorithmes

1	Algorithme du noyau du système conseiller	104
2	Algorithme du module de diagnostic	117

Chapitre 1

Introduction

1.1 Contexte général

Depuis 2004, internet a doublé de volume. Ainsi d'après une étude du cabinet Netcraft [Net], le nombre de sites internet dans le monde a dépassé, en mars 2007, le chiffre de 110 millions. Dans un même temps, le nombre d'utilisateurs du Web continue de croître. Ainsi, après qu'en mars 2006, comscore [Com] eut annoncé 694 millions d'utilisateurs d'internet, en mars 2007, le cabinet Research and Markets [RaM] a annoncé que ce chiffre devrait passer en 2007 à 1,35 milliard d'internautes.

Cependant, ces chiffres éloquentes vont de pair avec d'autres chiffres, moins glorieux, qui montrent que bon nombre de ces internautes rencontrent des difficultés dans leurs recherches d'informations sur le Web et abandonnent avant d'avoir atteint leur objectif.

1.1.1 Difficultés des utilisateurs du Web

Naviguer sur internet, y rechercher des informations, n'est pas chose facile : ainsi on sait que 50% des navigations échouent : pour deux utilisations d'internet, une seule permettra à l'utilisateur d'obtenir ce qu'il cherche [TR04] [KBP00] [CVG06] !

Les difficultés que rencontrent les internautes peuvent être analysées à deux niveaux : au niveau, très général, du Web et au niveau, plus précis, des sites Web.

Ces difficultés ont des causes multiples :

- Au niveau du Web :

1. L'utilisation généralisée des moteurs de recherche : la recherche d'information sur le Web consiste généralement à (1) localiser le site Web jugé intéressant et (2) rechercher l'information dans ce site [LBW00]. Pour effectuer la première tâche (*i.e.*, localiser le site qui va contenir les informations qu'ils cherchent) les utilisateurs n'ont à leur disposition qu'un outil : le moteur de recherche.
Or, l'utilisation d'un moteur de recherche qui propose une liste (souvent très longue) de pages susceptibles d'intéresser l'utilisateur, présente plusieurs inconvénients :
 - (a) Les moteurs de recherche sont plus difficiles à utiliser pour des utilisateurs novices [LBW00].
 - (b) Les moteurs de recherche nécessitent l'utilisation de mots clés censés représenter l'objet recherché [CJ03] ; or, quand l'utilisateur a peu de connaissances du domaine dans lequel se situe sa recherche, trouver de tels mots clés n'est pas trivial [Bel00] [SKZ03] [LLS02] [BHY96].
 - (c) Les moteurs de recherche fonctionnent bien quand la tâche de recherche reste relativement simple. Cependant, le Web est souvent utilisé pour effectuer des recherches complexes [JvO05].
 - (d) Les moteurs de recherche étant incapables de proposer des listes de liens relativement courtes, les utilisateurs ont adapté leurs styles de navigations. Ainsi [WOHM06] a montré que la durée de plus de 50% des visites d'un site ne dépassait pas 10 secondes.
2. La grande variété des fonctionnalités que les utilisateurs peuvent rencontrer sur le Web [AT06].

- Au niveau des sites Web :

1. La non prise en compte de la notion d'utilisabilité lors de la phase de développement des sites Web [Nie05] [Che05] [CVG06].
2. La non prise en compte des caractéristiques des utilisateurs [JvO05] [JH05] [PE98] [BM02].
3. L'évolution anarchique des sites [AT06].
4. La difficile gestion du but de la recherche de l'utilisateur : on sait que le but évolue au fur et à mesure de la navigation et que sa représentation doit constamment être en mémoire de travail de l'utilisateur. Or, la

richesse des pages, des fonctionnalités proposées par les hypermédias imposent des efforts de compréhension et d'analyse qui requièrent, eux aussi, une part de la mémoire de travail. En cas de saturation de la mémoire de travail (on parle alors de surcharge cognitive) la représentation du but risque d'en être chassée, laissant l'utilisateur avec un sentiment de désorientation [AT06] [TR04] [TPDB98].

1.1.2 Conséquences

Les difficultés présentées ci-dessus et rencontrées par les utilisateurs du Web ont des conséquences sur le ressenti des utilisateurs. Ceux-ci peuvent avoir deux types de sentiments : [AT06] [TR04] [TPDB98]

- incompréhension : c'est le sentiment souvent éprouvé par les utilisateurs de sites présentant des pages trop complexes, trop riches, trop longues [KGX⁺03].
- désorientation : c'est le sentiment souvent éprouvé par les utilisateurs de sites trop complexes avec de multiples liens et une arborescence compliquée. De plus, comme expliqué dans le paragraphe ci-dessus, l'analyse de pages compliquées peut entraîner une surcharge cognitive laissant l'utilisateur "dans un état où il ne sait plus où aller, ni d'où il vient, ni comment y retourner [...]" [AT06] [JH05] [KGX⁺03].

1.2 Problématique et Méthodologie

Les systèmes conseillers sont des systèmes dont l'objectif est d'assister les utilisateurs de systèmes informatiques. Les systèmes conseillers qui nous intéressent plus particulièrement sont les systèmes conseillers fondés sur une architecture **épiphyte**¹. Ces systèmes peuvent être greffés sur un système existant (un serveur Web par exemple) sans qu'il soit nécessaire de le modifier. Ils présentent plusieurs avantages : d'une part, leur fonctionnement n'impose pas de modification du système cible et, d'autre part, la séparation, imposée par cette architecture, entre les données du site cible et celles du système conseiller facilite la mise à jour et le développement du système conseiller et du site cible.

Quelle que soit l'architecture sur laquelle ils sont fondés, les systèmes conseillers ont pour objectif de guider, d'assister les utilisateurs au cours de leurs navi-

1. Le terme épiphyte vient de la botanique. Il désigne des plantes qui utilisent d'autres plantes comme support. Ce ne sont pas des parasites : elles ne puisent pas leur nourriture dans la plante support.

gations sur le Web ou dans des sites Web particuliers. Cette aide peut s'opérer de différentes manières : il est possible de proposer à l'utilisateur un supplément d'information, de simplifier sa navigation ou encore de le guider. C'est à ce dernier type d'aide, **le guidage**, que nous nous sommes intéressés plus particulièrement.

Par ailleurs, nous avons choisi de cibler les sites Web dits **complexes**, *i.e.*, des sites qui présentent un nombre important de pages, une structure large et profonde et de nombreuses fonctionnalités. Ces caractéristiques en font (1) des sites difficiles à appréhender et (2) des sites extrêmement riches de par les tâches qu'ils permettent de réaliser.

Enfin, nous avons choisi de proposer une aide **fondée sur la tâche de l'utilisateur**. Un site Web, notamment un site Web complexe, permet la réalisation de nombreuses tâches. Ces tâches, lorsqu'elles sont complexes, doivent être décomposées en sous tâches. Cette décomposition reste cependant souvent à la charge de l'utilisateur. Nous proposons d'assister l'utilisateur dans la réalisation de tâches complexes en prenant en compte cette décomposition, c'est-à-dire en assistant, en guidant l'utilisateur du début à la fin de la réalisation de la tâche.

Notre problématique peut donc se résumer ainsi : **comment élaborer un système conseiller qui prenne en compte la tâche de l'utilisateur pour le guider dans un site Web complexe ?**

Dans cette thèse, nous avons :

1. Cherché à dégager une approche qui permette la construction de systèmes conseillers fondés sur une architecture épiphyte.
2. Étudié comment représenter les tâches des utilisateurs de façon à pouvoir leur associer des conseils.
3. Recherché comment guider l'utilisateur en fonction des modèles de tâches construits.
4. Déterminé les différentes étapes d'élaboration d'un tel système conseiller.
5. Construit deux prototypes pour à la fois, illustrer notre approche et tester sa faisabilité.
6. Réitéré ces étapes pour modifier et améliorer l'ensemble.

1.3 Travail réalisé

Les résultats des travaux présentés dans ce mémoire sont les suivants :

1. Un état de l'art des différents travaux sur les systèmes conseillers, que nous avons analysés en fonction du type de l'aide apportée.

2. Une architecture générique permettant de construire des systèmes conseillers par instantiation de l'architecture, *i.e.*, la définition de modèles d'usage (qui vont servir de référence à l'analyse des actions de l'utilisateur) et de conseils.
3. Une approche et des outils facilitant l'instanciation de cette architecture, *i.e.*, facilitant la construction et la manipulation des modèles et des conseils.

Pour obtenir ces résultats nous avons tout d'abord étudié différents travaux sur les systèmes conseillers. Ces travaux présentent les aides qu'il est possible de proposer aux utilisateurs ainsi que les différentes techniques utilisées dans la mise en oeuvre des systèmes conseillers. Nous nous sommes fondés sur ces travaux pour définir le type d'aide que nous voulions apporter (en l'occurrence, le guidage).

Nous avons ensuite défini l'architecture que nous voulions mettre en oeuvre puis déterminé comment les tâches des utilisateurs pouvaient être modélisées. Nous avons proposé un processus d'élaboration d'un système conseiller épiphyte fondé sur l'utilisation de modèles de tâches. Nous avons choisi de laisser l'utilisateur sélectionner, parmi une liste de tâches définies par un analyste, celle qui lui paraissait correspondre à ses objectifs et pour laquelle il souhaitait être guidé.

Nous avons observé que la tâche de création des modèles était particulièrement difficile. Nous l'avons analysée, décomposée en plusieurs étapes puis nous avons proposé, pour chacune de ces étapes, différentes approches assistant la réalisation des tâches qui y sont associées.

Ces travaux ont permis la construction de deux prototypes de systèmes conseillers illustrant nos idées. Le premier prototype a été construit dans le cadre d'un contrat du LIUM avec le ministère français de l'Education Nationale et de la Recherche (projet "Educasource"). Le second est une simple mise à l'essai de notre approche (projet "Marco"). Ces prototypes ont été construits comme "preuves de concept". Ils ont permis d'étudier la faisabilité de notre approche et de l'architecture proposée : une implémentation effective a été réalisée, illustrant le fait qu'il est possible de construire un système conseiller fonctionnel à partir de notre approche. Cependant, nous n'avons pas effectué de tests d'utilisabilité et d'usage des prototypes, travail d'une autre nature qui reste une perspective pour de futurs travaux.

Nous avons choisi le langage Java pour développer ces prototypes de système conseiller, ce langage étant particulièrement adapté pour le développement d'applications Web.

L'approche méthodologique Platon 1 [TBB⁺04] fournit un cadre réflexif facilitant la conceptualisation des travaux de conception des EIAH : quatre dimensions d'analyse sont proposées : (1) la définition du projet de recherche, (2) le

cadre théorique de la recherche, (3) le résultat de la recherche et (4) le cycle de vie de la recherche. Ces quatre dimensions permettent une explicitation précise de la recherche effectuée. Cette grille d'analyse est prévue pour des travaux en EIAH. Elle nous a cependant permis de présenter les différents aspects de nos travaux. Ainsi, deux tableaux, issus de ce cadre d'analyse, sont présentés ci-après (résultat de la recherche, tableau 1.1, et cycle de vie de la recherche, tableau 1.2). Notons que les deux autres dimensions (définition du projet de recherche et cadre théorique de la recherche) sont proposées au chapitre 3 (tableaux 1.1 et 1.2) et que l'ensemble de cette analyse est présentée en annexe (annexe A).

1.4 Organisation du mémoire

Le premier chapitre de ce mémoire présente différents travaux sur les systèmes conseillers et permet de répondre aux questions suivantes : qu'est ce qu'un système conseiller ? Quelles caractéristiques peuvent être utilisées pour définir les systèmes conseillers ? Quels types d'aides les systèmes conseillers apportent-ils ? Sur quelles techniques se fonde le fonctionnement des systèmes conseillers ?

Le deuxième chapitre présente notre problématique. Nous définissons notre contexte de travail, présentons notre problématique en détail puis précisons le spectre de nos travaux. Nous indiquons ensuite les domaines scientifiques abordés (ceux auxquels nous avons contribué, ceux dans lesquels nous avons puisé des techniques et ceux dont nous nous sommes simplement inspirés). Nous présentons l'analyse de notre problématique en fonction des travaux sur les systèmes conseillers présentés dans le chapitre 1 et notamment par rapport au type d'aide apportée aux utilisateurs. Enfin, nous détaillons la méthodologie suivie ainsi que les terrains applicatifs choisis.

Le troisième chapitre détaille l'approche que nous avons proposée en définissant, d'une part, les principes généraux (nous décrivons l'architecture épiphyte générique, les modèles de tâches, les modèles d'utilisation) et, d'autre part, le processus d'élaboration d'un système conseiller fondé sur ces principes.

Le quatrième chapitre présente en détail l'architecture générique permettant l'élaboration de systèmes conseillers : les différents modules réseaux (en charge, par exemple, de la communication du système conseiller avec le serveur Web cible ou des différents types de connexion entre les utilisateurs et le système conseiller), le module de diagnostic (chargé de détecter, en fonction du modèle de tâches courant et de la navigation de l'utilisateur, le conseil à produire) ou encore le module d'élaboration de conseils.

Le cinquième chapitre détaille le processus de conception des modèles d'uti-

[C] Résultat de la recherche	
[C.1] Nature des résultats	<p>Les résultats que nous proposons sont :</p> <ul style="list-style-type: none"> - Une contribution à la définition de la notion de système conseiller (définition, éléments de classification, exemples, méthode de conception). - Des éléments de méthodologie de conception de systèmes conseillers. - Une architecture informatique générique, mise en oeuvre de l'approche proposée. - Des outils de support à la conception de systèmes conseillers. - Une preuve de concept par des prototypes opérationnels.
[C.2] Aspects généraux et génériques des résultats	<p>Plusieurs éléments de nos travaux sont réutilisables :</p> <ul style="list-style-type: none"> - Nous avons analysé les travaux sur les systèmes conseillers et présenté les différents types d'aide qu'ils permettent de proposer. La grille d'analyse proposée peut permettre de positionner d'autres travaux, que ce soit en fonction de l'aide proposée ou des techniques utilisées. - Nous avons proposé des éléments de méthodologie de conception de systèmes conseillers en précisant, notamment, les différentes étapes nécessaires à l'élaboration d'un système conseiller. Ces éléments peuvent être utilisés et adaptés, pour permettre la constructions de systèmes conseillers dans d'autres contextes.
[C.3] Types de validation des résultats	<p>Nous avons utilisé, comme méthode d'évaluation, des prototypes qui ont été construits comme "preuves de concept" : leur élaboration a validé la faisabilité de notre approche et de l'architecture proposée.</p>

TAB. 1.1 – *Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Résultat de la recherche*

[D] Cycle de vie de la recherche	
[D.1] Contexte de lancement de la recherche	Un contrat industriel a été signé entre le LIUM et le ministère de l'Education Nationale et de la Recherche. L'objectif de ce contrat était la proposition d'une approche et d'un prototype visant à étudier l'impact potentiel de l'association d'un système conseiller au site Web Educasource.
[D.2] Historique de la recherche	J'ai commencé à travailler en tant qu'ingénieur d'étude sur cette problématique. Ce travail s'est ensuite poursuivi, sous la direction de P. Tchounikine, dans le cadre d'une thèse.

TAB. 1.2 – *Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Cycle de vie de la recherche*

lisation et les outils d'aide que nous avons conçus et développés.

Le sixième chapitre présente des exemples de mises en oeuvre. Pour illustrer notre approche et tester sa faisabilité, nous nous sommes intéressés plus particulièrement à deux sites complexes : Educasource et Apache. Les différentes étapes nécessaires à la création des modèles de tâches et à l'instanciation du système conseiller sont décrites pour ces deux sites.

Ce mémoire se termine par un ensemble de conclusions et de perspectives.

Enfin, dans les annexes, nous présentons des informations évoquées dans ce document mais non détaillées telles que les pré-traitements effectués sur les logs des serveurs du site de la fondation Apache (site que nous avons utilisé comme exemple pour créer des modèles de tâches) ou encore le détail des calculs effectués sur ces mêmes logs.

Chapitre 2

Les systèmes conseillers et systèmes voisins

2.1 Définition Générale

2.1.1 Définition

Les systèmes conseillers sont apparus en 1992 avec le système tapestry [GNOT92] [RIS⁺94] [RV97]. Leur développement est fondé sur une constatation : sur le Web comme dans la vie courante, il est souvent nécessaire de faire des choix alors que nos connaissances, notre expérience ou notre niveau sont insuffisants pour que nous soyons à même d’appréhender correctement les différentes alternatives proposées. Dès lors, une aide, un conseil sont précieux [RV97].

Les systèmes conseillers sont définis comme “des systèmes qui proposent une aide active à l’utilisateur [...], aide fondée sur une analyse des actions et des productions de l’utilisateur” [PT02]. Ils peuvent être utilisés pour assister les utilisateurs de logiciels particuliers ou encore de sites Web. Dans la suite de ce chapitre, c’est aux systèmes conseillers proposant des conseils aux utilisateurs de ce type d’application informatique, les sites Web, que nous allons nous intéresser plus particulièrement.

Leur principe général est simple : l’utilisateur fournit (directement ou indirectement, consciemment ou inconsciemment) des informations au système qui, en retour, lui propose des conseils [RV97] [VPF02].

Un conseil est un ensemble d’informations. Il peut, par exemple, indiquer à l’utilisateur qu’il est possible d’effectuer telle ou telle action, lui apporter des données supplémentaires ou encore lui faire remarquer que sa dernière action est

peut être maladroite ou incohérente.

Un conseil n’oblige en rien l’utilisateur. Il n’intervient pas directement dans le travail de l’utilisateur [EZ03], *i.e.* il n’est qu’informatif [PT02].

Différents termes sont utilisés pour évoquer ces systèmes : on parle de systèmes conseillers, de systèmes de recommandations (“recommend systems”) ou encore de systèmes d’aide. Dans la suite de ce document, pour nos travaux, nous utiliserons le terme de système conseiller. Les termes de système de recommandations et de système d’aide seront utilisés pour faire référence à des travaux pour lesquels les auteurs ont utilisé ces terminologies. Précisons, par ailleurs, que les “systèmes voisins” indiqués dans le titre de ce chapitre font référence à certains travaux qui n’ont pas explicitement la dénomination de système conseiller (ou de système de recommandations) mais qui, de par l’aide qu’ils proposent ou la technique utilisée, en sont proches.

2.1.2 Caractéristiques

L’analyse des travaux sur les systèmes conseillers nous a permis de mettre en lumière plusieurs types de caractéristiques qui permettent de distinguer ces travaux : le champ d’application des systèmes conseillers, leur domaine d’application, les informations recueillies sur l’utilisateur, le processus de récupération de ces informations, leur persistance, les moments auxquels sont présentés les conseils, le type de l’aide proposée, la forme de présentation des conseils et enfin la technique sur laquelle se fonde le système conseiller.

- Le champ d’application. Le champ d’application est le type du système pour lequel un système conseiller est développé. Le spectre de ces champs d’application est large : l’aide peut être proposée à des utilisateurs de logiciels spécifiques (par exemple, médicaux, techniques etc.), de bornes multimédia, de sites Web, etc.
- Le domaine d’application. On trouve les systèmes conseillers dans des domaines variés. Citons, par exemple :
 - Domaines en relation avec le Web :
 - Proposition de conseils à propos de :
 - Livres [HCOC02] [MR00],
 - Musique [AAM01],
 - Articles scientifiques [MARS02],
 - Publications nautiques [GIK99],

- Photographies [LRS01],
 - Recherche dans les newsgroups [ACT01], [RIS⁺94],
 - Recherche de locations [BHY96],
 - Aide à l'utilisation de cours en ligne [BES97] [BC98] [HN01] [BAB⁺03],
 - Aide à la navigation [BS97] [JFM97] [Lie95] [JT98],
- Domaines autres que le Web :
 - Assistance médicale [FRFMS00] [EZ03],
 - Aide à la gestion des e-mails [MGT⁺87],
 - etc.
- Les informations concernant l'utilisateur. Pour fonctionner, les systèmes conseillers ont besoin de plus ou moins d'informations sur l'utilisateur. Ces différents besoins s'expliquent par les différentes techniques utilisées et par les différents types d'aide apportés. Ainsi, alors que certains systèmes conseillers se limiteront à analyser la navigation de l'utilisateur, d'autres vont construire puis maintenir un profil pour chaque utilisateur. La gestion des profils utilisateurs est souvent considérée comme le cœur des systèmes conseillers de ce type [MLR03] et le but de ces systèmes est d'acquérir un maximum d'informations sur chaque utilisateur pour ensuite lui proposer une aide la plus adaptée possible. Dans ce type de systèmes, les informations recueillies sont, par exemple :
 - Le temps passé à lire chaque page,
 - Les pages enregistrées dans le marque page du navigateur de l'utilisateur,
 - Les pages enregistrées sur le système de l'utilisateur,
 - Les pages lues,
 - Les pages imprimées,
 - Les articles achetés,
 - Les appréciations données aux pages,
 - Les centres d'intérêts,
 - Les domaines de compétences,
 - Etc.
- La technique de récupération des informations. Certains systèmes conseillers requièrent de nombreuses informations concernant l'utilisateur, d'autres moins. Les techniques utilisées pour récupérer ces informations peuvent permettre

de caractériser les systèmes. Différentes méthodes sont utilisées, par exemple :

- Interroger directement l'utilisateur. Il s'agit, par exemple, de demander à l'utilisateur lors de la création de son profil, son niveau de connaissances dans tel ou tel domaine, ses centres d'intérêt, ses goûts. Par la suite, au fil des lectures de l'utilisateur, il pourra lui être demandé ses appréciations sur tel ou tel article ou objet. L'interrogation peut se faire, par exemple, via un formulaire (au risque de déranger l'utilisateur qui pourrait juger cette pratique comme intrusive [MRS01] et refuser de renseigner le formulaire) ou, de façon moins directe, via des questions présentes sur certaines pages (on demande, par exemple, à l'utilisateur de noter la page qu'il vient de parcourir).
- Analyser le parcours de l'utilisateur. Moins intrusive, cette technique peut se faire à l'aide de différentes techniques (les types de systèmes présentés ci-dessous le sont en partant de celui qui permet de récupérer le plus de données vers celui qui en propose le moins) :
 - Systèmes dédiés [LCPM05] : un système dédié à l'observation de l'utilisateur est installé sur son ordinateur. Il permet d'enregistrer de façon très complète toutes les actions de l'utilisateur (par exemple, les mouvements de la souris, les temps de lecture d'une page, les données saisies au clavier, etc.).
 - Navigateurs dédiés [JH05] [HJ04] [HvD04] [CLWB01] ou modules associés à des navigateurs du marché [FBH00] : il s'agit de systèmes construits dans le but d'enregistrer les actions de l'utilisateur dans le cadre d'une navigation sur le Web. Les actions enregistrées peuvent être les suivantes :
 - Mémorisation d'une page dans l'historique de navigation,
 - Utilisation de la touche "retour",
 - Utilisation du bookmark,
 - Mouvements de la souris,
 - Clicks,
 - Défilement de la page,
 - Sauvegarde, impression de la page courante,
 - Saisie d'information dans les formulaires rencontrés,
 - Etc.
 - Systèmes de type proxy [RT04b] [JFM97] [MRS01] [MRS02] [MTCGdGP03] : il s'agit d'intercepter les requêtes HTTP du navigateur de l'utilisateur. Ce type de systèmes permet de créer une

image de la navigation de l'utilisateur observé. Cette image peut cependant être un peu floue à cause, par exemple, du cache du navigateur qui peut masquer certaines actions.

- La persistance des données. La persistance des données est une autre caractéristique des systèmes conseillers. Ainsi, les concepteurs de systèmes conseillers s'interrogent-ils sur la durée de vie et l'évolution des données relatives à l'utilisateur : les données peuvent, par exemple, être traitées de différentes façons [SKR99] :
 - Supprimées lorsque l'utilisateur quitte le site,
 - Conservées et réutilisées lorsque l'utilisateur reviendra sur le site,
 - Mises à jour en fonction des allées et venues de l'utilisateur,
 - Utilisées pour proposer des conseils à d'autres utilisateurs,
 - Présentées anonymement à d'autres utilisateurs,
 - Présentées non anonymement à d'autres utilisateurs,
 - Etc.

La persistance des données pose la question du respect de la vie privée des utilisateurs et notamment (1) lorsque sont utilisées des techniques de “collaborative filtering” où l'expérience de chaque utilisateur est utilisée pour proposer à d'autres des conseils (cf. ci-dessous) [RV97] et (2) lorsque les données récoltées sont suffisamment précises et fournies pour pouvoir intéresser des sociétés privées [SKR99].

- Les moments auxquels sont présentés les conseils. Les conseils peuvent être proposés à différents moments :
 - Tout au long de la navigation. A chaque nouvelle page ouverte, le système vérifie s'il dispose de pages présentant des informations complémentaires de cette nouvelle page et, si oui, les propose.
 - A des moments clés. Par exemple, lorsque le système détecte une inactivité de l'utilisateur [PB02] ou lorsque l'utilisateur utilise de nouveau le système après l'avoir abandonné pendant un certain temps
 - A la demande de l'utilisateur. Par exemple, un bouton d'aide peut être proposé à l'utilisateur [ZGH⁺05] : [BL98] suggère ainsi d'insérer un tel bouton dans les pages qui présentent un concept particulier ou une difficulté.
- L'aide proposée. L'aide proposée par les systèmes conseillers est extrêmement variée. Ainsi, elle peut être complexe, proposée en fonction du parcours de l'utilisateur, de son niveau, de son but, ou plus simple, proposée,

par exemple, uniquement en fonction de l'endroit où se trouve l'utilisateur (comme, par exemple, le trombone de Windows) [LCPM05].

- La technique de présentation des conseils. Pour présenter les conseils aux utilisateurs, plusieurs techniques peuvent être mises en oeuvre :
 - Afficher le conseil dans une page adjacente. Le système conseiller construit un conseil et le propose dans une page adjacente à celle du site Web.
 - Ajouter/modifier/supprimer des liens au sein d'une page. Le système conseiller modifie les liens de la page Web que l'utilisateur désire consulter.
 - Adapter le contenu d'une page. Le système conseiller modifie le contenu de la page Web que l'utilisateur désire consulter. Cette modification va au delà de la modification des liens qu'elle comprend : des paragraphes peuvent être ajoutés, d'autres supprimés, il s'agit de proposer un contenu qui soit fonction des caractéristiques de l'utilisateur.
- La technique utilisée. Différentes techniques sont utilisées pour faire fonctionner les systèmes conseillers. On peut citer, par exemple, les techniques dites de "collaborative filtering" qui s'appuient sur la comparaison des comportements des utilisateurs ou celles fondées sur la comparaison du contenu des documents parcourus par les utilisateurs ("content based comparison").

Dans la suite de ce chapitre, nous présentons plus avant différents travaux sur les systèmes conseillers en les considérant selon trois dimensions qui nous paraissent particulièrement pertinentes : l'objectif poursuivi par les concepteurs du système conseiller, la technique de présentation des informations/conseils et la technique de gestion des données manipulées. Ces différentes dimensions correspondent à des points de vue sur les travaux, et ne sont pas tout à fait indépendantes. Elles permettent cependant de mettre en évidence d'autres caractéristiques des travaux présentés dans la littérature.

2.2 Dimension 1 : types d'aide proposés

Les objectifs poursuivis par les concepteurs des systèmes conseillers sont, en général, liés à l'un ou l'autre ou un mélange des deux objectifs généraux suivants : (1) rendre accessible à l'utilisateur des informations qu'il pourrait avoir du mal à obtenir et (2) limiter la charge cognitive nécessaire à l'utilisation du site.

On sait, en effet, que la mémoire de travail d'un individu est limitée. Or, diminuer la charge cognitive nécessaire, par exemple, à la compréhension d'une page Web ou à la structure du site, permet à l'utilisateur de se consacrer pleinement à la recherche d'information, à la gestion de son but et limite les risques de désorientation [AT06].

Pour réaliser ces objectifs généraux, les systèmes conseillers peuvent proposer différents types d'aide comme, par exemple, apporter directement de l'information supplémentaire à l'utilisateur, simplifier sa navigation en jouant sur les liens ou sur le contenu des pages, ou encore le guider. Notons que guider l'utilisateur peut amener à considérer comme pertinent de lui apporter directement de l'information supplémentaire et/ou de simplifier sa navigation : comme nous l'avons déjà indiqué, ces types d'aide permettent d'offrir un point de vue particulier sur les travaux sur les systèmes conseillers mais ne sont pas indépendants.

2.2.1 Apporter directement de l'information supplémentaire à l'utilisateur

Principes. Il s'agit pour ces systèmes de proposer à l'utilisateur des informations supplémentaires, complémentaires de celles proposées dans la page qu'il consulte.

Exemples. L'utilisateur consulte une page (ce qui revient à sélectionner un produit ou une information); le système lui propose, par exemple, d'en parcourir d'autres présentant des informations en relation avec la page courante et susceptibles de l'intéresser. Différents systèmes fonctionnent ainsi :

- Amazon [Ama] (site de vente en ligne) recherche des similitudes entre les comportements des utilisateurs et se fonde sur ces comparaisons pour constituer un profil utilisateur et proposer des liens vers des objets en rapport avec l'objet choisi par l'utilisateur (par exemple, il sera proposé des conseils du type "Vous voulez acheter l'objet X. Sachez que les utilisateurs qui achètent cet objet, achètent aussi souvent l'objet Y").
- SEWeP [Evv03] se fonde sur une ontologie et sur les logs du site Web sur lequel il fonctionne pour proposer à l'utilisateur des pages du site qui pourraient l'intéresser.
- Libra [MR00] s'intéresse plus particulièrement à la recommandation de livres. Il se fonde sur une base de données et un profil utilisateur pour proposer une liste de livres susceptibles d'intéresser l'utilisateur.

2.2.2 Simplifier la navigation de l'utilisateur

Principes. Il s'agit pour ces systèmes de faciliter la navigation des utilisateurs. Dans ce but, les systèmes conseillers transforment la page Web initiale en l'adaptant à l'utilisateur. Ainsi, ces systèmes peuvent (1) insérer, supprimer ou encore annoter des liens des pages Web présentées à l'utilisateur. Proposer des liens vers des pages en relation avec la page courante permet à l'utilisateur d'accéder directement à ces pages sans passer par des menus intermédiaires [Bru96]. Ces techniques, parfois appelées "path-shortening" [KGX⁺03], permettent à l'utilisateur d'accéder à des pages qui n'auraient pas été accessibles directement depuis la page courante. En réduisant la longueur des chemins conduisant à l'information recherchée, elles permettent une simplification des navigations [ZGH⁺05].

Ces systèmes peuvent aussi (2) adapter, modifier, le contenu des pages présentées aux utilisateurs (et non simplement les liens). Ces modifications peuvent être fonction des buts, des connaissances, des centres d'intérêt ou des autres caractéristiques des utilisateurs [BL98] [KGX⁺03] [Vas96].

Exemples.

- Interbook [EB98] utilise des techniques d'annotation de liens, pour assister l'apprenant : certains liens sont annotés avec des étiquettes telles que "well learned", "learned", "ready to be learned" ou "not ready to be learned" en fonction du niveau de l'utilisateur et des informations proposées dans la page accessible via le lien.
- 2L670 et son évolution AHA! [BC98] [BAB⁺03] [BSS06] proposent des cours en ligne. Ces systèmes sont fondés sur une approche d'adaptation du contenu des pages. Ils prennent en considération les connaissances des utilisateurs et leur but. Ainsi, le système proposera un cours de grammaire anglaise en ayant pris soin de l'adapter au niveau de l'étudiant (maîtrise-t-il telle notion ? connaît-il telle autre ?). Il y a donc prise en compte des connaissances du sujet (le cours de grammaire proposé est adapté au niveau de l'étudiant) et de son but (un cours de grammaire pour un sujet qui veut apprendre la grammaire).

2.2.3 Guider l'utilisateur

Principes. Il s'agit pour ces systèmes de proposer un guidage et ainsi permettre à l'utilisateur de réaliser son objectif. Dans ce but, une aide est proposée à l'utilisateur tout au long de sa navigation. Cette aide doit être cohérente, *i.e.*, le système ne doit pas perdre de vue le but de l'utilisateur et proposer une succession de conseils

amenant l'utilisateur à réaliser ce but. Ce type d'aide se différencie ainsi des aides, plus ponctuelles, du type "Apporter directement de l'information supplémentaire à l'utilisateur" présentées au paragraphe 2.2.1.

Le guidage de l'utilisateur renvoie à la notion de tâche utilisateur : une tâche correspond à un enchaînement d'actions ou d'opérations permettant la réalisation d'un but dans un environnement donné (par abus de langage, une tâche est souvent dénotée par son but) [TPDB98].

Exemples. L'approche Musette [CPM03] [LCPM05] propose de guider les utilisateurs en fonction de leurs navigations et de leurs actions. Dans ce but, le système observe le comportement des utilisateurs et construit une trace primitive en se fondant sur deux modèles : un modèle d'utilisation qui définit "de quoi la trace sera constituée" et un modèle d'observation qui définit "comment celle-ci sera générée". La trace primitive, ainsi constituée, est ensuite analysée et des extasis (EXplained TAsk SIGNatures, *i.e.*, des "morceaux" de la trace) sont extraits. Ces extasis correspondent à des actions définies (par exemple, sauvegarder une page, enregistrer la page courante dans les signets, imprimer, etc.) auxquelles des aides sont associées. Pour chaque extasis détecté, le système est donc en mesure de proposer une aide.

Pixed [HFM04] est une application de l'approche Musette aux environnements d'apprentissage à distance. Les logs du serveur Web dédié à l'environnement d'apprentissage sont analysés en fonction de modèles. Cette analyse, associée à des mécanismes de raisonnement à partir de cas, permet à l'application de proposer aux utilisateurs qui le demandent un guidage via une aide fondée sur la proposition de chemins permettant la réalisation de leur objectif (l'objectif est un concept à apprendre et est choisi par chaque utilisateur dans une ontologie. Voir, entre autres, [CPM02] pour d'autres exemples de mise en oeuvre de l'approche Musette).

2.3 Dimension 2 : techniques de présentation des informations/conseils

Différentes techniques de présentation des conseils peuvent être proposées : le conseil peut être proposé dans une page adjacente à la page Web courante ou proposé en modifiant les liens ou le contenu de la page Web courante.

2.3.1 Afficher une page adjacente

Principes. Il s’agit de proposer à l’utilisateur, dans une page adjacente à la page Web courante, des informations supplémentaires, complémentaires de celles proposées dans la page qu’il consulte : l’utilisateur consulte une page ; le système lui en propose une autre présentant des informations en relation avec la page courante et susceptibles de l’intéresser.

Exemples.

- PageGather [PE98] propose, à partir de l’analyse des logs d’un site Web, de construire des pages d’index (“the index page synthesis”). Ces pages illustrent chacune un thème abordé dans le site et regroupent des pointeurs vers d’autres pages du site en rapport avec cette thématique. Ces pages, formant “a helpful tour guide”, sont ainsi présentées à l’utilisateur.
- SurfLen [FBH00] analyse les historiques de navigation des utilisateurs et propose à l’utilisateur, dans une page spécifique, des informations supplémentaires tout au long de sa navigation.

2.3.2 Ajouter/modifier/supprimer des liens au sein d’une page

Principes. Il s’agit de transformer les liens d’une page Web. Les opérations impliquant des liens vers des pages HTML peuvent être de différentes natures (ces mécanismes d’aide sont parfois appelés “adaptive navigation support (ANS)” [BL98] [Bru01] [Bru04]) :

- Insertion de liens : il s’agit d’insérer des liens dans les pages parcourues par l’utilisateur. Cette technique est appelée “link augmentation” [BEBH01]. Son but est de faciliter la navigation de l’utilisateur en réduisant la longueur du chemin qui va l’amener aux pages qu’il recherche.
- Tri de liens : il s’agit de trier (*i.e.* de ranger, de classer) les liens présentés dans les pages parcourues par l’utilisateur. Le but de cette technique est de faciliter la navigation de l’utilisateur [Bru96].
- Suppression de liens [KKP01] : il s’agit de supprimer des liens dans les pages parcourues par l’utilisateur. Cette technique est appelée “adaptive hiding” [BL98]. Son but est de limiter la charge cognitive nécessaire à la compréhension de la page en éliminant des liens inutiles pour l’utilisateur.
- Annotation de liens [Bru96] [JFM97] : il s’agit de fournir plus d’information à l’utilisateur sur les liens présents dans les pages en se fondant sur ses centres d’intérêt ou sur son niveau. Par exemple, on grisera un lien vers une

page jugée trop difficile pour l'utilisateur ou on attirera son attention sur un lien vers une page jugée intéressante [JH05].

Exemples. Interbook propose des ressources électroniques à des étudiants. Pour chaque étudiant un profil est créé, permettant au système d'annoter les liens proposés dans chaque page. Les annotations sont, par exemple : “ressource déjà connue” (tous les concepts présentés dans ce document sont connus de l'étudiant), “ressource accessible” (les concepts pré-requis sont connus de l'étudiant) ou encore “ressource non accessible” (des concepts pré-requis ne sont pas encore maîtrisés) [EB98].

2.3.3 Adapter le contenu d'une page

Principes. Il s'agit de réarranger la page (au delà des liens qu'elle comprend) avant de la proposer à l'utilisateur. Cette technique est appelée “adaptive presentation” ou “content-level adaptation” [BL98]. Son principe est le suivant : des éléments de la page initiale sont présentés à l'utilisateur, d'autres non. Elle se différencie des techniques présentées ci-dessus par le fait que c'est la page dans son ensemble (*i.e.*, du texte, des paragraphes) qui est modifiée et non plus simplement certains liens.

Exemples.

- [GI03] se fonde sur les propriétés sémantiques (définies par l'auteur) d'un document virtuel pour adapter la page qui sera présentée à l'utilisateur.
- AHA! [BAB⁺03] [BSS06] est un système qui propose et personnalise des cours en ligne à l'Université Technologique de Eindhoven (en anglais “an adaptive hypermedia learning environment”) et que nous avons déjà présenté page 27.

2.4 Dimension 3 : techniques de gestion des données manipulées

Les systèmes conseillers utilisent différentes techniques pour analyser les données à leur disposition. Le choix d'une technique par rapport à une autre est un critère majeur puisqu'il va orienter (1) le schéma fonctionnel du système conseiller

et (2) les conseils qu'il sera possible de présenter et donc le type de l'aide proposée. La technique utilisée peut d'ailleurs être un critère pour classifier les systèmes conseillers (cf. paragraphe 2.1.2).

Nous avons distingué trois catégories de techniques :

- Utilisation de la sémantique des informations contenues dans les pages : le but des systèmes fondés sur ce type de technique est de proposer des conseils en se fondant sur des informations relatives à la page et notamment à leur concept principal.
- Analyse du comportement des utilisateurs : le but des systèmes fondés sur ce type de technique est de proposer des conseils en se fondant, par exemple, sur une analyse des parcours des utilisateurs ou des pages qu'ils ont appréciées ou non.
- Méthodes mixtes : le but des systèmes fondés sur ce type de technique est de proposer des conseils en se fondant non seulement sur le contenu des pages parcourues mais aussi sur l'analyse de l'utilisateur (parcours, pages appréciées ou non).

2.4.1 Utilisation de la sémantique des informations contenues dans les pages

Certains systèmes conseillers fondent leur fonctionnement sur l'utilisation de descriptions sémantiques du contenu des pages. Ces descriptions sémantiques peuvent être obtenues à l'aide d'une analyse du contenu des pages ou être mises à disposition du système conseiller sous forme de concepts associés à la page et tirés d'une ontologie ou d'une base de données. Disposer d'informations relatives aux concepts présentés dans les pages parcourues par l'utilisateur permet, en comparant ces informations avec celles sur l'utilisateur (ses préférences, ses goûts, ses centres d'intérêt), de lui proposer des pages susceptibles de l'intéresser.

Nous détaillons ci-dessous plusieurs techniques utilisées par les systèmes conseillers pour prendre en compte le contenu des pages.

2.4.1.1 “Content based comparison”

Principes. Les principes des systèmes fondés sur cette approche sont :

- La recherche de pages sémantiquement proches. Il s'agit du développement d'une idée déjà ébauchée par les techniques d'“information filtering”

qui analysent les pages en y recherchant des mots clés. Ici, l'analyse est plus poussée et se fonde sur l'utilisation de techniques telles que K-Nearest Neighbors algorithm [MARS02]. Ces techniques construisent tout d'abord une représentation de la page à l'aide de mots extraits de la page et censés en être représentatifs. Elles comparent ensuite la représentation de la page courante (*i.e.*, sa liste de mots) avec d'autres représentations de pages déjà parcourues. Le but est de trouver des pages ayant une représentation proche, en d'autres termes, un concept proche. Ce mécanisme est appelé "item to item correlation" [Bur02a].

- La construction de profils utilisateurs ("user modelling techniques"). Pour tirer tout le bénéfice de l'analyse sémantique des pages lues, une fois extrait le concept clé (ou sa représentation), il est intéressant de conserver cette information. C'est pourquoi, les systèmes vont créer puis éventuellement faire évoluer des profils utilisateurs. Ainsi, sachant qu'un utilisateur est intéressé par des articles ou des objets d'un domaine particulier, le système conseiller sera à même de proposer des objets relatifs à ce domaine. Notons cependant que, d'une part, la construction de profil utilisateur n'est pas spécifique à cette approche (cf., par exemple, paragraphe 2.4.2.1, où est présentée une autre approche utilisant cette technique) et que, d'autre part, la création et l'utilisation de profils utilisateurs n'en font pas pour autant une méthode mixte telle que nous l'avons décrite au début de cette section : les méthodes mixtes, qui sont présentées au paragraphe 2.4.3, sont des méthodes qui proposent une utilisation de la sémantique des pages et une analyse du comportement des utilisateurs.

Exemples.

- Letizia [Lie95], un système conseiller assistant l'utilisateur tout au long de ses navigations sur le Web.
- COGITO [ADF⁺02], un système développé pour les sites de commerce en ligne.
- LIBRA [MR00], un système suggérant des livres à ses utilisateurs.
- SEWeP [EVV03] [ELPV04], un système fondé sur l'utilisation d'une ontologie et sur l'analyse des logs du site Web sur lequel il fonctionne, et recommandant des pages de ce site.

Avantages. L'avantage principal de cette approche réside dans le fait qu'il n'est pas nécessaire de demander des informations à l'utilisateur (approche non intrusive) : les systèmes utilisant cette approche sont capables d'analyser les pages, de

construire un profil pour chaque utilisateur et de maintenir ce profil sans intervention de l'utilisateur. Or, on sait que si l'utilisateur est prêt à fournir quelques informations au système, il ne faut pas que cette tâche requière trop de temps car le risque de voir l'utilisateur abandonner la tâche sera alors élevé [Dek05].

Inconvénients. Cette approche souffre cependant de plusieurs inconvénients :

- Les systèmes fondés sur cette approche analysent la page lue par l'utilisateur, extraient les concepts clés de la page puis proposent des pages déjà analysées et dont les concepts clés sont proches de ceux de la page courante. Le système se constitue donc une sorte de base de données dans laquelle il va piocher pour proposer des pages aux utilisateurs. A l'initialisation du système, cette base est vide et le système ne sera pas capable de proposer des pages aux utilisateurs. Cette période appelée "cold start period" [MARS02] n'est cependant pas rédhibitoire puisque l'utilisateur qui n'a pas fourni d'information au système, *i.e.*, n'a pas donné de son temps, ne sera pas frustré en ne recevant pas de conseil (ce qui n'est pas forcément le cas dans toutes les approches où il peut arriver qu'un utilisateur passe du temps pour renseigner, par exemple, son profil, sans pour autant en trouver de bénéfices).
- Lorsque les pages lues par les utilisateurs appartiennent à des domaines très variés, on observe un éparpillement des données et de fait, la construction d'une base de données opérationnelle pour le système conseiller s'en trouve ralentie. Dès lors, la période de démarrage (pendant laquelle le système ne peut proposer de conseil, *i.e.*, "the cold start period") se trouve rallongée.
- Ce type d'approche se fonde sur l'extraction automatique des concepts clés d'une page. Or, un tel exercice n'est pas toujours possible. Ainsi, en l'absence d'annotation, des pages contenant des sons, des images, des vidéos ou traitant de sujets généraux seront difficilement analysables [MMN01].
- Les systèmes fondés sur cette approche construisent un profil pour chaque utilisateur et vont proposer aux utilisateurs des pages dont le concept clé a été noté comme intéressant pour eux. De ce fait, lors du fonctionnement du système, on va observer une spécialisation des pages proposées (les pages appartiennent toutes à un même domaine et au fil des navigations on risque d'obtenir dans le profil, des domaines d'intérêt de plus en plus précis) sans qu'il soit possible d'ouvrir le champ des pages proposées à des domaines adjacents [BS97].
- L'objectif de ces systèmes est de proposer à l'utilisateur (1) des pages qui l'intéressent (des pages dont le concept clé appartient aux concepts qui semblent intéresser l'utilisateur) (2) des pages qu'il n'a pas encore lues (ce qui nécessite de maintenir dans le profil de chaque utilisateur la liste des

pages déjà parcourues) et (3) des pages réellement nouvelles, *i.e.*, des pages proposant une information nouvelle à l'utilisateur. Ce dernier point pose de nombreux problèmes, notamment, dans le cadre de la proposition de pages issues de sites d'actualités. Comment, en effet, s'assurer que l'utilisateur n'a pas déjà lu un article légèrement différent sur la forme mais présentant la même information ? Ce phénomène a été défini comme le "portfolio effect" et fait l'objet de recherches [Bur02a].

- La comparaison entre deux articles dépend du point de vue de l'utilisateur ("similar[ity] depends on what one's goal are." [Bur02a]) : on peut considérer que 2 articles abordent le même thème, que deux objets ayant une description semblable se ressemblent, mais cette affirmation reste subjective et dépend notamment de l'objectif de l'utilisateur. Ainsi, [Bur02a] fait remarquer que quelqu'un peut considérer qu'une voiture et un bateau sont similaires lorsque sa destination de vacances est proche de la mer, ce qui n'est pas le cas lorsque sa destination est au milieu des terres.

2.4.1.2 "Knowledge based recommendations"

Principes. Pour proposer des conseils, ce type de système s'appuie, comme les systèmes présentés ci-dessus, sur des informations relatives à la sémantique des pages. Cependant, ces informations ne sont plus extraites des pages mais puisées dans une ou plusieurs bases de connaissances. Ces bases de connaissances présentent des objets et leurs différentes caractéristiques et doivent pouvoir être interrogées [Bur99]. Ces bases peuvent être de différentes formes. Ce peut être :

- Des ontologies. Ainsi, [MRS01] utilise une ontologie (en l'occurrence dmoz [Dmo]) pour affiner les centres d'intérêts des utilisateurs. Ces ontologies peuvent être figées [CRP00] ou être mises à jour par le système conseiller [MARS02].
- Des bases de données. Par exemple, [BHY96] [Bur02a] présente Entree qui s'appuie sur une base de données pour proposer une assistance dans le choix d'un restaurant. Les restaurants sont enregistrés dans la base, ainsi que plusieurs de leurs caractéristiques (adresse, cuisine, prix, ambiance, etc.)

Plusieurs techniques sont proposées par les systèmes conseillers pour utiliser ces bases :

- Parcours de la base par les utilisateurs : ainsi [MRS01] permet aux utilisateurs d'affiner la déclaration de leurs centres d'intérêt en leur proposant de naviguer dans une ontologie.

- Parcours de la base par le système : ainsi [MRS02] utilise une ontologie pour (1) construire les profils (les profils sont construits à l’aide des termes proposés par l’ontologie) et (2) proposer aux utilisateurs des articles dont les domaines sont proches de ceux mémorisés dans leurs profils. Ce mécanisme permet d’ouvrir le champ des propositions faites aux utilisateurs et d’ainsi limiter l’effet, présenté plus haut, de spécialisation des profils et des conseils.
- Méthodes mixtes, *i.e.*, parcours de la base par les utilisateurs et le système. Ainsi, [GMV99] décrit, dans un premier temps, chaque objet de la base à l’aide d’un graphe conceptuel. Puis, cette description est transformée à l’aide d’une ontologie en une description sémantique de l’objet qui est enregistrée. Pour assister l’utilisateur dans ses recherches, les termes utilisés par l’utilisateur lors de la description de sa recherche sont transformés à l’aide de la même ontologie. La description sémantique de la recherche de l’utilisateur ainsi obtenue est alors comparée aux descriptions des objets enregistrés par le système, pour ensuite proposer ceux qui correspondent à la recherche de l’utilisateur.

Différentes méthodes sont utilisées pour extraire les données des bases. A côté des techniques classiques de recherche dans une base, on notera l’utilisation de techniques de raisonnement à partir de cas (RàPC, en anglais “case based reasoning”). Ces techniques permettent d’utiliser l’expérience enregistrée dans la base pour proposer à l’utilisateur différentes solutions [Bur02b].

Exemples.

- OntoSeek [GMV99] : un système proposant des articles en se fondant sur une analyse sémantique (1) des objets enregistrés dans la base et (2) des requêtes des utilisateurs.
- Karina [CRP00] : un système proposant dynamiquement à l’utilisateur un ensemble de pointeurs vers des ressources pédagogiques. Ces pointeurs sont organisés en un schéma de navigation précis, construit à l’aide d’une ontologie du domaine et dédié à un utilisateur donné (prise en compte de l’objectif de l’utilisateur).
- Foxtrot [MRS02] : un système proposant à des scientifiques des articles de recherche. Les navigations des scientifiques sont observées grâce à l’utilisation d’un proxy. L’observation de leurs lectures, associée à l’utilisation d’une ontologie, permet de proposer à l’utilisateur des documents dont la thématique n’est pas strictement liée aux centres d’intérêt qui lui ont été assignés.

On peut aussi citer les systèmes suivants qui se fondent sur des techniques de raisonnement à partir de cas :

- CoCoA [AAM01] : un système conseiller proposant à ses utilisateurs des morceaux de musique.
- Entree, Car navigator, video navigation, RentMe [BHY96] [Bur99] : ces systèmes ont été développés dans le cadre du projet FINDME. Ils utilisent des techniques de RàPC et proposent plusieurs types de conseils.

Enfin, citons les travaux sur les hypermédias adaptatifs (cf., par exemple, [BC98] [BAB⁺03] [ABHV04] [BB04] [BBS03] [BM02] [Bru96]) qui construisent et maintiennent à jour des profils utilisateurs et utilisent des bases de données pour enregistrer des informations concernant les pages à proposer aux utilisateurs. Ces informations, une fois mises en relation avec les données concernant l'utilisateur, vont permettre de lui proposer des pages adaptées.

Avantages. L'utilisation de telles bases permet :

1. d'augmenter la précision des recherches (les bases peuvent contenir des informations précises et, on l'a vu, des techniques comme le raisonnement à partir de cas permettent de raffiner les résultats).
2. de limiter le risque de n'avoir aucun résultat à proposer à l'utilisateur [Bur02b].
3. de supprimer la période pendant laquelle le système crée la base et n'est pas capable de proposer de conseil aux utilisateurs ("cold start period") [Bur99].

Inconvénients. Cependant, utiliser de telles bases, nécessite du travail d'ingénierie, puisqu'il va être nécessaire de construire une base pour un domaine ou un site particulier [Bur02b]. Par ailleurs, effectuer des recherches dans une base n'est pas toujours aisé : il est facile de décrire ce que l'on cherche quand il s'agit d'objets simples (un prix, un lieu, une date). Cependant, dès lors que l'on parle d'objets dont l'appréciation dépend de l'environnement ou de chacun, la recherche devient plus difficile.

2.4.1.3 Création de modèles cognitifs

Principes. Les techniques fondées sur la création et l'utilisation de modèles cognitifs s'appuient sur une analyse sémantique des pages et sur des hypothèses relatives aux mécanismes cognitifs des utilisateurs de sites Web. Il peut s'agir, par exemple, de prendre en compte les mécanismes cognitifs mis en oeuvre lors de la

compréhension d'un texte ou lors de l'orientation de l'utilisateur dans le site Web. Des modèles cognitifs peuvent être construits pour simuler le comportement d'un utilisateur ou d'un groupe d'utilisateurs en fonction de son but. Ces simulations vont permettre, par exemple, de tester l'utilisabilité d'un site ou de proposer des liens vers des pages.

Exemples.

- [KBP00] propose un modèle cognitif, CoLiDeS (pour *Comprehension-based Linked model of Deliberate Search*) simulant la navigation Web des utilisateurs. Ce modèle cognitif est fondé sur le principe que c'est, en grande partie, la compréhension des textes et des images d'une page Web qui guide la navigation de l'utilisateur. Lorsqu'ils naviguent, les utilisateurs choisissent, dans chaque page, le lien qui semble le plus proche de leur représentation du but. Plusieurs processus entrent alors en jeu : la navigation, l'observation privilégiée d'un objet, la compréhension de cet objet et sa sélection. Considérant un but donné, les auteurs vont simuler la navigation de l'utilisateur à l'aide du modèle. La technique de LSA ("latent semantic analysis") est utilisée par les auteurs pour mesurer la corrélation entre le but de l'utilisateur et les objets présents sur les pages Web (liens, images, textes, etc.). L'analyse des navigations observées permet de proposer aux concepteurs de site des outils sur lesquels ils peuvent s'appuyer pour (1) valider la structure du site et les pages proposées (des conseils, tels que la mise en valeur des liens ambigus qui nécessitent une description, un label plus précis ou plus développé ou la réduction de la profondeur du site, facteur qui semble influencer le nombre de navigations réussies dans un site, peuvent être proposés) et (2) tester la navigation d'un utilisateur ayant un but donné (en d'autres termes, vérifier l'utilisabilité du site pour un but et un public donné).
- L'approche proposée par [JvO05] est une extension de celle de [KBP00] présentée ci-dessus. Cette dernière présentait la compréhension de la page courante comme un élément majeur dans les choix de navigation de l'utilisateur. Le modèle cognitif proposé par [JvO05] ne se limite pas à cette dimension : les auteurs considèrent, en effet, que les capacités d'orientation et de positionnement dans le site Web sont aussi des facteurs importants à prendre en compte pour estimer la réussite ou non d'une navigation. Ils proposent donc CoLiDeS+, une évolution du modèle proposé par [KBP00]. Un prototype a été développé puis expérimenté [JH05]. L'objectif du prototype était de guider des personnes aveugles ou mal voyantes tout au long de leur navigation sur le Web. Des expérimentations du prototype ont montré que la suggestion de liens était bien perçue par les utilisateurs, qu'elle li-

mitait le sentiment de désorientation et que la navigation s'en trouvait plus structurée.

Avantages. Les systèmes issus de ces approches peuvent simuler le comportement d'utilisateurs dans le site, en fonction de leurs connaissances et de leurs buts. Ils se fondent sur des modèles cognitifs expliquant les mécanismes qui poussent l'utilisateur à choisir une page plutôt qu'une autre. Ces systèmes sont donc précieux pour identifier et remédier aux problèmes d'utilisabilité de site (il s'agit alors d'aider le concepteur de sites Web). Ils peuvent aussi être utilisés pour proposer des pages aux utilisateurs (il s'agit cette fois d'aider l'utilisateur de sites Web).

Inconvénients. Une première difficulté relative à cette approche est l'élaboration des modèles cognitifs. On sait que plusieurs mécanismes cognitifs entrent en jeu lorsque l'utilisateur parcourt un site et on peut s'interroger sur les mécanismes qui doivent être considérés par le modèle. Ainsi, les deux approches présentées ci-dessus, qui ont les mêmes fondements, proposent des modèles cognitifs différents : alors que [KBP00] ne prend en compte que la capacité de l'utilisateur à comprendre les textes et les images de la page qui lui sont présentés, [JvO05] prend en compte les capacités de positionnement de l'utilisateur. Un second obstacle réside dans la difficulté à déterminer le but de l'utilisateur puis de son utilisation. Les deux techniques présentées ci-dessus ont choisi de demander explicitement à l'utilisateur quel était son but lorsqu'il abordait le site. Or, on sait (1) que poser des questions n'est pas toujours bien accepté (de par le caractère intrusif de cette action) et (2) que le but change, évolue au fur et à mesure de la navigation et n'est pas toujours facile à définir, même par l'utilisateur lui-même [TR04].

2.4.2 Utilisation d'informations relatives au comportement des utilisateurs

A la différence des systèmes qui s'appuient sur une analyse sémantique des pages parcourues par l'utilisateur, les systèmes présentés ci-après s'orientent vers une analyse du comportement des utilisateurs et en particulier de leurs navigations sur le Web [BPAG05]. Notons bien la différence avec l'utilisation des modèles cognitifs : il ne s'agit plus de modéliser les mécanismes cognitifs qui sont en jeu lorsque l'utilisateur parcourt un site, mais d'analyser, d'extraire les actions de l'utilisateur et de fonder l'aide sur cette analyse. L'analyse est donc plus pragmatique : on ne va pas chercher à comprendre le pourquoi des actions de l'utilisateur, on va plutôt s'intéresser aux actions elles mêmes.

Pour enregistrer ces informations, ces systèmes construisent et maintiennent à jour des profils utilisateurs (on parle de techniques de “User modelling”). Ces profils regroupent des informations telles que le niveau de l'utilisateur, ses centres d'intérêt, ses buts, ses préférences, etc. Ces informations ont pu être obtenues en posant directement des questions aux utilisateurs [ARD05]. Cependant, ce type de questionnaire peut être mal perçu par les utilisateurs car (1) cette pratique peut être jugée comme intrusive [Dek05] et (2) prendre le temps de remplir un formulaire peut être considéré par l'utilisateur comme une perte de temps s'il n'en tire pas rapidement (voire immédiatement) un bénéfice [ARD05]. De fait, certains systèmes ont développé des techniques non intrusives à l'aide, par exemple, de proxies [CCS05] pour analyser le comportement des utilisateurs et créer des profils. On notera cependant que le respect de la vie privée des utilisateurs peut nécessiter (1) l'adaptation des informations enregistrées et (2) que les techniques utilisées par les systèmes pour collecter les informations soient adaptées aux lois en vigueur (on observe alors un autre problème : le Web étant international, les concepteurs de tels systèmes devront s'interroger sur les lois à prendre en compte) [KGX⁺03].

Nous présentons ci-dessous plusieurs techniques qui sont utilisées pour analyser le comportement des utilisateurs.

2.4.2.1 Techniques de “collaborative filtering”

Principes. Le but des approches utilisant ces techniques est de prédire les goûts de l'utilisateur pour tel ou tel objet qu'il n'a pas encore observé [GFMG05]. Ces approches sont fondées sur le fait (1) que des personnes qui ont déjà fait des choix similaires ont une probabilité élevée de faire de nouveau des choix identiques [RIS⁺94] ; il s'agit donc d'une corrélation personne à personne (“people to people correlation” [Bur02a]), qui s'oppose aux techniques citées plus haut qualifiées elles de corrélation item à item (“item to item correlation”) et (2) qu'il est plus efficace d'impliquer l'utilisateur [GNOT92] : en s'aidant d'informations récoltées implicitement sur l'utilisateur ou données explicitement par l'utilisateur, les conseils seront plus pertinents.

Le fonctionnement d'une majorité de ces systèmes est le suivant : au cours de sa navigation, il est demandé à l'utilisateur de noter ou de donner une appréciation sur la page qu'il est en train de lire. Ces informations sont enregistrées par le système. Ainsi, lorsqu'une page est appréciée par un utilisateur, le système recherche les autres utilisateurs qui ont apprécié cette page et extrait de sa base les pages qu'ils ont bien notées. Ces pages sont alors proposées à l'utilisateur.

Pour ce faire, le système construit et maintient à jour des profils des utilisateurs.

teurs. Typiquement, un profil est un vecteur d'objets auxquels sont associés des évaluations [Bur02a].

Algorithmes. Comme indiqué ci-dessus, ces approches comparent les profils utilisateurs [FBH00] puis proposent à un utilisateur des pages que d'autres utilisateurs, qui ont les mêmes goûts que lui, ont appréciées. Différents algorithmes sont utilisés pour comparer les utilisateurs [GFMG05] :

- “k-Nearest Neighbor method” ou méthode des k plus proches voisins [MMN01]. Cette méthode est aussi appelée “lazy learning approach” [GFMG05] puisqu'elle ne nécessite pas de phase d'apprentissage : chaque note donnée par l'utilisateur à une page est enregistrée et le système doit parcourir l'ensemble des données pour être à même de proposer un conseil. Les données de départ sont une matrice avec en colonne, les objets et en ligne, les utilisateurs. Chaque ligne, *i.e.*, chaque vecteur, représente donc les différentes notes données par l'utilisateur pour un objet. Ce sont ces vecteurs qui vont être comparés les uns aux autres.
On peut considérer trois étapes principales dans le fonctionnement de l'algorithme :
 1. Comparaison des vecteurs. Plusieurs techniques peuvent être utilisées pour comparer deux vecteurs. Citons “Pearson correlation” (utilisé, par exemple, par [RIS⁺94] ou [FBH00]) et “cosine similarity” (utilisé, par exemple, par [KBP00]).
 2. Sélection des utilisateurs dont les vecteurs sont les plus similaires à celui de l'utilisateur courant. Ces utilisateurs forment le voisinage de l'utilisateur courant.
 3. Prédiction de l'appréciation donnée par l'utilisateur courant en calculant la moyenne des notes données par les utilisateurs du voisinage.
- Différents outils fondés sur des techniques d'apprentissage ont été proposés :
 - Support Vector Machine (SVM). Cette méthode a été proposée par [CV95] en 1995. C'est une méthode de classification binaire. Elle serait plus performante de la techniques des “k-Nearest Neighbors” lorsqu'on observe un éparpillement des données [GFMG05].
 - Bayesian classifieur [PB97].
 - Réseaux de neurones.
 - “Rocchio's algorithm” [Roc71] adapté à la classification de texte [ILA95].
 - etc.

Exemples. Cette méthode a été utilisée fréquemment et on observe un grand nombre d'applications dans différents domaines :

- Recommandations de news [RIS⁺94].
- Recommandations de livres [Ama].
- Recommandations de musiques [Upe94].
- Recommandations de films [HSRF95].
- etc.

Avantages. Cette approche présente plusieurs avantages :

- Elle est simple et relativement facile à mettre en oeuvre.
- Elle est indépendante de la représentation des données manipulées.
- Elle fonctionne bien avec des objets complexes (à la différence des approches citées plus haut) : musique, films, etc.

Inconvénients. Cette approche présente plusieurs défauts :

- “Cold start problem”. On observe, comme dans l’approche fondée sur l’analyse du contenu des pages, une période de latence pendant laquelle le système n’est pas capable de proposer des pages aux utilisateurs. Cette période est due au fait que l’approche se fonde sur la comparaison d’utilisateurs. Or, quand peu d’utilisateurs sont enregistrés dans la base, il n’est pas possible (ou du moins rare) de trouver des affinités entre utilisateurs, et donc de proposer des conseils (“the cold start problem” [MARS02] [Bur99]). De plus, il apparaît que des difficultés peuvent subsister même avec une base non vide : un éparpillement des données (“data sparsity”) entraîne les mêmes problèmes, *i.e.*, l’impossibilité de proposer des conseils [FBH00] [Bur99]. Ces problèmes risquent d’entraîner une non-utilisation du système puisque l’on va demander aux premiers utilisateurs de noter des pages (donc de donner de leur temps, de s’impliquer) sans qu’ils en tirent de bénéfice immédiat.
- Intrusivité. De nombreux systèmes, fondés sur cette approche, interrogent l’utilisateur : il lui est demandé de noter les pages, de donner un avis, une appréciation. Cette démarche peut être considérée comme intrusive et gênante par des utilisateurs et donc les décourager d’utiliser le système [Dek05] [MRS01].
- Dysfonctionnements. Plusieurs dysfonctionnements sont observés :
 - Ce type de systèmes fonctionne mal avec des items trop généraux : l’exemple souvent proposé pour illustrer ce type de problème est celui

des bananes [Bur99] : si l'on tente d'aider l'utilisateur d'un site Web d'une grande surface, on lui conseillera inévitablement d'acheter des bananes puisque celles-ci sont fréquemment achetées et donc liées à de très nombreux produits (vous achetez un produit Y. Comme vous, M. A a acheté le produit Y. Il a aussi acheté des bananes. On vous propose des bananes...). A l'inverse, le système ne fonctionnera pas pour des objets qui s'achètent seuls (par exemple, une voiture) puisqu'il sera difficile pour le système d'associer l'objet à un autre objet, et donc de faire des propositions [Bur99].

- On observe une spécialisation du conseil : l'utilisation d'un profil utilisateur tend à ne proposer qu'un certain type de page à l'utilisateur, sans ouverture possible vers d'autres pages.
- On observe aussi une surexposition des articles populaires : ils sont beaucoup vus, beaucoup notés et donc beaucoup proposés [RAC⁺02]. Il y a donc mise en place d'un cercle vertueux. A l'inverse, des articles mal notés dès les premières appréciations, ne seront pas ou peu proposés.
- “portfolio effect”. Lorsque le système propose une page à un utilisateur, il s'assure que l'utilisateur ne l'a pas déjà lue. Cependant, le problème devient ardu lorsqu'il s'agit, par exemple, de proposer à l'utilisateur des articles de presse, des dépêches : il est difficile pour un système automatique de faire le tri entre différents articles présentant globalement la même information [Bur02a].
- Originalité. Il est reconnu [CGM⁺99] que ce type de système ne fonctionne pas avec des personnes trop originales, *i.e.*, des personnes qui font des choix différents de ceux effectués par la plupart des autres utilisateurs.

2.4.2.2 “Web usage mining”

Principes. Une approche pour proposer des conseils en se fondant sur une analyse des utilisateurs et de leurs comportements est d'utiliser des techniques de Web mining [EV03]. Les techniques de Web mining peuvent être divisées en 3 classes : analyse du contenu des pages (“content mining”), extraction d'informations en terme d'usage (“usage mining”) ou en terme de structure de pages ou de sites (“structure mining”) [SCDT00]. Il s'agit pour ces techniques d'analyser les traces laissées par les utilisateurs du Web sur les sites Web ou les proxys (logs) ou enregistrées par des navigateurs ad hoc.

Nous allons nous intéresser plus particulièrement au Web usage mining. Ces techniques permettent en effet de :

1. Extraire des comportements particuliers : essayer, grâce à l'analyse de traces, d'extraire des types caractéristiques d'utilisation, *i.e.*, des modèles d'utilisation ("pattern").
2. Comparer, grouper des objets ayant des caractéristiques similaires. On parle alors de "clustering".

En appliquant ces techniques aux logs d'un serveur Web, il est ainsi possible de disposer, sans intervention de l'utilisateur (ce qui n'est pas un avantage négligeable), d'informations sur les utilisateurs, informations qui pourront ensuite être utilisées pour leur proposer des conseils.

Algorithmes. Les algorithmes utilisés par les techniques d'"usage mining" fonctionnent généralement selon les étapes suivantes :

1. Pré-traitements ("preprocessing") [CMS99] [Mic02]. Il s'agit d'effectuer des traitements sur les données pour qu'elles puissent ensuite être analysées. Dans le cadre de l'analyse de logs de serveurs Web, il s'agit d'opérations telles que :
 - Anonymisation des données. Il s'agit alors, par exemple, de remplacer les IPs des utilisateurs par des numéros.
 - Suppressions des requêtes "exotiques" (attaques diverses, requêtes mal formulées, etc.).
 - Extraction de sessions. Il s'agit de repérer dans les logs le parcours d'un même utilisateur. On parle alors de session [BL00]. Plusieurs problèmes, spécifiques aux logs, se présentent alors :
 - L'utilisation de proxy masque l'utilisateur final : si deux utilisateurs utilisent le même proxy et parcourent simultanément un même site, les logs du site ne montreront qu'un seul utilisateur.
 - Des utilisateurs peuvent utiliser des outils masquant ou modifiant leur IP. On observe alors dans les logs plusieurs parcours pour un seul parcours effectif.
2. Recherche de modèles d'utilisation ("pattern discovery"), de "micro-comportement" [BPAG05]. Cette recherche peut être effectuée à l'aide de différentes techniques. Citons :
 - L'analyse statistique.

- Les règles d'association ("association rules" ; par exemple, montrer que telle page est consultée après telle autre) [LAR02].
 - Le regroupement d'objets ("clustering") : il s'agit de trouver et de regrouper des pages ou des utilisateurs ayant des caractéristiques communes.
3. Analyse des modèles trouvés ("pattern analysis"). Il s'agit d'analyser, de filtrer les modèles découverts lors de l'étape 2.

Exemples. Plusieurs systèmes conseillers utilisent des techniques de Web usage mining. Par exemple :

- [FCS01] et [PE98] qui se fondent sur ces techniques pour adapter les sites Web.
- [MCS00] et [MPT99] qui se fondent sur une analyse des usages d'un site Web pour proposer aux utilisateurs des listes de liens vers des pages.

Avantages. Cette approche présente le principal avantage d'être non intrusive : elle est totalement transparente pour l'utilisateur à qui rien n'est demandé et qui pourra bénéficier, sans investissement, de conseils.

Inconvénients. Les reproches associés à cette approche sont principalement dus aux logs et aux limitations inhérentes à leur utilisation. On sait, par exemple, que certaines informations ne sont pas présentes dans les logs : les phénomènes de cache (cache du navigateur, cache du proxy, etc.) privent les logs d'un certain nombre d'informations. Ainsi, quand l'utilisateur clique sur la touche de retour vers la page précédente, la page affichée est issue du cache du navigateur ; cette requête n'apparaîtra donc pas dans les logs du serveur Web. De même, les informations données par l'utilisateur ne sont pas toujours récupérées (par exemple, les informations associées à la commande POST ne sont pas enregistrées dans les logs). On sait aussi que certaines informations risquent d'être, sinon fausses, du moins altérées : l'IP de l'utilisateur, par exemple, est une information qui peut être incorrecte (cf. ci-dessus).

2.4.3 Utilisation de méthodes mixtes

Principes. Pour limiter les inconvénients des approches citées ci-dessus, des systèmes proposent d'en utiliser plusieurs simultanément. Il apparaît en effet que les techniques utilisées ne sont pas forcément concurrentes et, qu'ensemble, elles

peuvent permettre de construire des systèmes pertinents [HKR00]. Plusieurs approches ont été proposées pour coupler les différentes techniques :

- **Pondération (“Weighted”)**

Principes. Les systèmes issus de cette approche utilisent différentes techniques en parallèle, les pondèrent en fonction du contexte courant, et font ensuite des propositions aux utilisateurs en fonction de ces pondérations [GR04].

Exemples. Le système proposé par [CGM⁺99] (recherche d’articles d’actualité les plus appropriées dans un journal en ligne) s’adapte en fonction des données utilisées : si les données courantes sont très éparpillées c’est une technique fondée sur l’analyse du contenu sémantique des pages (“content based approach”) qui sera utilisée. En revanche, si les données sont riches (par exemple, si le système dispose de nombreuses appréciations sur les objets de sa base), ce sont des techniques de “collaborative filtering” qui seront choisies. Il est à noter que les auteurs réfutent la dénomination d’approche hybride : les deux approches ne fonctionnent pas ensemble (dans le sens d’un même traitement ordonné, avec les méthodes en séquence) mais en parallèle et sans qu’il n’y ait d’interaction entre elles.

[Paz99] propose une organisation différente : plusieurs techniques sont utilisées pour proposer des conseils. Pour chaque méthode, les conseils (*i.e.*, les liens proposés) sont ordonnés selon leur pertinence et pondérés. Un bilan des conseils est effectué et le système propose à l’utilisateur ceux de poids maximum.

Avantages. Le principal avantage d’une telle méthode est le fait que les techniques fonctionnent en parallèle : elles évoluent sans mettre en péril l’une ou l’autre, et la pondération des techniques permet de choisir la meilleure.

Inconvénients. Le fait d’utiliser les techniques en parallèle permet, lors de chaque conseil, de choisir le meilleur et donc globalement d’avoir une plus grande pertinence des réponses. Cependant, cela ne permet pas de corriger les défauts intrinsèques à chacune des approches (par exemple, on sait que les systèmes fondés sur l’approche de “collaborative filtering” connaissent une période, au démarrage du système, pendant laquelle ils ne peuvent proposer de conseil. Pendant cette période, une approche fondée sur la pondération ne pourra bénéficier de cette technique).

- **Séquencement**

Principes. Dans ce type d’approches, les différentes méthodes sont utilisées séquentiellement : des traitements sont effectués par une méthode puis complétés par une autre.

Exemples. Plusieurs systèmes [MMN01] [BS97] sont fondés sur l’approche suivante : (1) des techniques de “contend-based analysis” sont utilisées pour construire et enrichir les profils des utilisateurs et (2) des techniques de “collaborative filtering” sont employées pour proposer à l’utilisateur des conseils.

Avantages. Le but d’agréger plusieurs techniques est de limiter certains inconvénients propres à chacune. Ainsi, dans les exemples ci-dessus la succession de deux techniques permet de limiter les défauts des techniques de “collaborative filtering” : la technique de “contend-based analysis” permet d’enrichir les profils, et le fait d’avoir des profils présentant plus d’informations sur les utilisateurs permet de meilleures comparaisons entre utilisateurs, et donc de meilleurs conseils.

Inconvénients. Les défauts inhérents à chacune des approches ne sont pas gommés pour autant. Ainsi, dans certains domaines (photographie, vidéo, articles présentant des opinions, etc.) la méthode de “contend-based analysis” ne fonctionne pas bien et ne permettra pas d’enrichir qualitativement les profils des utilisateurs.

- **Autres**

La recherche dans le domaine des systèmes conseillers étant active et riche, une grande variété de techniques est proposée [Bur02a]. Les techniques les plus fréquemment utilisées ont été présentées ci-dessus. D’autres existent. On trouvera, par exemple :

- Des techniques qui proposent de passer d’une méthode à une autre selon le degré de confiance dans la méthode au moment où s’effectue la recherche (on parle de “switching” ; la difficulté est alors de déterminer quelle technique utiliser et à quel moment).
- Des techniques fondées sur la proposition aux utilisateurs de résultats générés en parallèle par différentes techniques (pas de notion d’ordre ou de préférence d’une méthode par rapport à une autre à un moment donné).

2.5 Synthèse

Les systèmes conseillers proposent une aide à l'utilisateur qui peut être de différentes natures : proposition à l'utilisateur d'informations supplémentaires, simplification de ses navigations ou encore guidage. Différentes techniques sont utilisées pour permettre la proposition de cette aide, que ce soit au niveau de sa présentation (dans une page adjacente, par le biais de modifications de liens de la page courante ou de son contenu) ou de la manipulation des données prises en compte pour la création des conseils (on trouve ainsi des techniques fondées sur une analyse de la sémantique des pages proposées, des techniques fondées sur une analyse du comportement des utilisateurs ou encore des techniques mixtes utilisant des informations relatives à la sémantique des pages et à l'utilisateur).

Il est important de noter que ces différentes dimensions correspondent à des points de vue qui sont plus ou moins pertinents, transversaux, redondants, selon les travaux. Ils permettent à notre sens de mettre en évidence des aspects importants de la recherche sur le domaine, mais ne doivent pas être perçus comme permettant de classer, au sens strict, les travaux. Par exemple, PageGather [PE98] propose à l'utilisateur des liens vers des pages du site courant, qui pourraient intéresser l'utilisateur. Ces liens apportent directement de l'information supplémentaire à l'utilisateur (en l'occurrence, ils informent l'utilisateur que cette page est susceptible de les intéresser) mais permettent aussi de simplifier sa navigation (en suivant ces liens, l'utilisateur peut accéder directement à une page lui proposant l'information qu'il recherche).

Nous présentons ci-dessous une grille d'analyse caractérisant différents travaux en fonction de ces trois dimensions.

Systèmes	Type d'aide			Tech. de présentation			Techniques de gestion des données					
	Apporter info supp.	Simplifier la navigation	Guider	Page adjacente	Modif. liens	Modif. contenu	Analyse sémantique			Analyse des utilisateurs		Mixtes
							Content Based Comp.	Knowl. Based Recom.	Modèles Cognitifs	Collab. filtering	Web Usage Mining	
AHA! [BBS03]		X				X		X				
Amazon [Ama]	X			X						X		
Automatic Pers. [MCS00]	X			X							X	
CoCoA [AAM01]	X			X				X				
COGITO [ADF ⁺ 02]	X			X			X					
CoLiDeS+ [JvO05]			X	X					X			
Entree (projet FINDME) [BHY96]		X				X		X				
FAIRWIS [BCGP02]	X	X		X						X		
Foxtrot [MRS02]	X					X						X
Karina [CRP00]		X				X		X				
Libra [MR00]	X			X			X					
Letizia [Lie95]	X			X			X					
MetaLens [SKR04]	X					X					X	
Musette, Pixed [HFM04]			X			X					X	
PageGather [PE98]	X	X		X							X	
QuIC [BEBH01]	X				X							X
SEWeP [EVV03]	X			X								X
Tapestry [GNOT92]	X			X								X
WebTool [MPT99]	X				X						X	

TAB. 2.1 – Systèmes conseillers, aide apportée et techniques utilisées

Chapitre 3

Problématique

3.1 Présentation

3.1.1 Contexte

Nous avons exposé dans l'introduction les difficultés que rencontrent les utilisateurs lorsqu'ils naviguent dans des sites Web. Ces difficultés sont telles qu'actuellement une navigation sur deux échoue.

Dès lors, qu'est-il possible de proposer aux utilisateurs d'un site Web dont l'analyse des usages montre qu'un nombre important d'entre eux est en difficulté ? En d'autres termes, quelles solutions, quelles approches vont-elles permettre de modifier, d'adapter le site Web pour qu'il devienne plus facilement utilisable ?

Plusieurs solutions peuvent être proposées :

- Redévelopper le site. L'usage du site en question peut être analysé, par exemple, à l'aide de l'analyse des logs du site Web (on va rechercher des comportements du type sorties trop rapides, sorties à des endroits où il ne devrait pas y avoir de sortie, utilisation détournée de fonctionnalités, etc.) ou via des interviews d'utilisateurs. Le site peut ensuite être redéveloppé en corrigeant son interface et sa structure pour répondre aux attentes des utilisateurs. Cependant, cette solution n'est pas nécessairement optimale : (1) modifier un site est long et coûteux et (2) cette technique ne donne, à moyen terme, aucune assurance aux concepteurs du site : les usages des utilisateurs peuvent changer et, de nouveau, des problèmes d'utilisation peuvent être observés.
- Utiliser un système conseiller. Les systèmes conseillers ont pour objectif de faciliter la navigation des utilisateurs sur internet ou dans des sites Web

particuliers. Les différentes techniques qu'ils utilisent ont été présentées au chapitre précédent. Ils permettent d'assister l'utilisateur en lui proposant des informations supplémentaires, en simplifiant sa navigation ou encore en le guidant.

3.1.2 Problématique détaillée

Nous avons choisi de nous intéresser aux systèmes conseillers et plus particulièrement à ceux dont l'architecture permet une séparation entre le système conseiller et les données du serveur Web. Nous avons observé au chapitre précédent que plusieurs types d'aide pouvaient être proposés à l'utilisateur et c'est à l'un d'entre eux, le guidage, que nous allons nous intéresser plus particulièrement. Ainsi, nous envisageons de guider l'utilisateur à travers des sites complexes en fondant l'aide proposée sur la tâche de l'utilisateur.

La question suivante résume la problématique de ces travaux :

Comment élaborer un système conseiller qui, prenant en compte la tâche de l'utilisateur, le guide à travers un site complexe ?

Cibler des sites complexes. Les difficultés que rencontrent les utilisateurs sont d'autant plus criantes que les sites sont complexes. Les sites complexes sont des sites qui présentent :

1. un nombre important de pages,
2. une structure large et profonde,
3. de nombreuses fonctionnalités et
4. de réelles difficultés d'utilisation liées aux trois points précédents et/ou à la complexité des tâches qui sous-tendent généralement l'utilisation de ce type de sites.

Ces sites visent, en général, à permettre à leurs utilisateurs d'accomplir des tâches complexes. C'est dans ce type de situation que l'utilisateur a le plus besoin d'être guidé et que l'ajout d'un système conseiller, prenant en compte la tâche de l'utilisateur, prend tout son sens. C'est donc à ce type de sites que nous allons nous intéresser plus particulièrement.

Pour illustrer nos propos dans ce paragraphe et dans la suite de ce document, nous nous sommes intéressés au site Apache [Apa]. Ce site présente les différents projets de la fondation Apache et notamment celui du serveur Web Apache, le serveur Web le plus utilisé au monde (source Netcraft [Net], décembre 2007). Le site Apache est un exemple de site complexe (données récupérées en mai 2006) :

- il présente près de 300 000 pages html,
- il est structuré en sous-projets (une petite trentaine), chacun étant structuré et présenté comme un petit site Web indépendant (d'où une grande diversité des interfaces),
- il présente de nombreuses fonctionnalités (moteur de recherche, forum, listes de diffusion, accès sécurisé, gestion des versions, etc.).

Utiliser des systèmes conseillers. Nous avons vu au paragraphe 3.1.1 que plusieurs approches pouvaient permettre d'améliorer la navigation des utilisateurs d'un site Web. C'est à la deuxième approche, proposant d'utiliser des systèmes conseillers, que nous avons choisi de nous intéresser. Parmi ces systèmes, nous nous intéressons plus particulièrement à ceux qui présentent une architecture qui leur permet d'être opérationnels sans qu'il soit nécessaire de modifier le site cible. Ces systèmes présentent plusieurs avantages : (1) il n'est pas nécessaire de modifier le site cible (les coûts de modification du site cible, évoqués plus haut, sont ainsi évités) et (2) une telle architecture amène à une séparation des conseils vis à vis des données du site cible, facilitant la maintenance du serveur Web et celle du système conseiller, augmentant la portabilité, la généricité, la flexibilité et la qualité du système conseiller.

Aider l'utilisateur. Les systèmes conseillers peuvent être utilisés de différentes manières pour aider l'utilisateur : on peut les employer pour proposer à l'utilisateur des informations complémentaires, simplifier sa navigation ou encore le guider (cf. paragraphe 2.2). Nous nous intéressons à cette dernière forme d'assistance, le guidage. On peut remarquer que le guidage de l'utilisateur, *i.e.*, l'aider à naviguer dans le site, peut amener à considérer, comme des sous-objectifs, le fait de lui proposer des informations supplémentaires et/ou de simplifier sa navigation.

Prendre en compte la tâche de l'utilisateur. L'aide proposée à l'utilisateur, en l'occurrence le guidage, peut être fonction des préférences, du parcours ou encore de la tâche de l'utilisateur. C'est sur ce dernier point que nous avons décidé de bâtir notre approche : le fonctionnement du système conseiller et l'aide proposée

seront fondés sur un ensemble de tâches, une tâche correspondant à un enchaînement d’actions ou d’opérations permettant la réalisation d’un but dans un environnement donné. Dans notre approche, c’est un expert¹ qui définit les tâches prises en compte par le système conseiller et élabore une aide associée à ces tâches.

1. Dans la suite de ce document, nous appellerons “expert” la ou les personnes (ce ne sont pas forcément à chaque fois les mêmes personnes) qui interviennent lors d’étapes où une expertise particulière est nécessaire. Par exemple, un expert peut identifier les tâches qui nécessitent une aide, un autre élaborer les conseils à présenter aux utilisateurs, un troisième définir la ou les conditions de présentation des conseils, et un dernier gérer sa représentation informatique.

Reprenons l'exemple du site Apache. Ce site est notamment visité par des d'utilisateurs qui ont pour objectif d'installer le serveur Web Apache.

Une façon assez naturelle (et confirmée par l'analyse des logs) de réaliser un tel objectif est de décomposer la tâche "installer le serveur Web Apache" en sous-tâches relatives à une recherche d'informations :

- Rechercher des informations sur les possibilités du serveur Apache
- Rechercher des informations sur ses limites
- Rechercher des informations sur les différentes étapes qui permettent son installation
- etc.

Cette décomposition en sous-tâches peut se révéler plus ou moins aisée selon le niveau de complexité de la tâche et/ou les compétences de l'utilisateur, et ce d'autant que la structure du site ne la facilite pas.

La prise en compte de la tâche "installer le serveur Web Apache" peut s'effectuer en construisant une modélisation des différentes actions nécessaires à la réalisation de cette tâche, ce que nous appellerons dans la suite de ce document un modèle de tâches. La figure 3.1 présente un exemple de modèle d'utilisation. Ce type de modèles représente différents chemins dans le site Web (*i.e.*, différentes pages Web parcourues successivement) et les conseils, associés aux chemins, qui servent à aider l'utilisateur. Il permet, par ailleurs, au système conseiller de suivre le parcours de l'utilisateur dans le site Web.

Ainsi, le modèle présenté figure 3.1 propose un accès à des pages permettant de découvrir des informations sur les possibilités du serveur Apache (sous-tâche "Rechercher des informations sur les possibilités du serveur Apache"), à des pages proposant des informations sur ses limites (sous-tâche "Rechercher des informations sur ses limites") et à des pages présentant des informations sur les différentes étapes qui permettent son installation (sous-tâche "Rechercher des informations sur les différentes étapes qui permettent son installation"). Il modélise de cette façon les différentes actions nécessaires à la réalisation de la tâche "installer le serveur Web Apache".

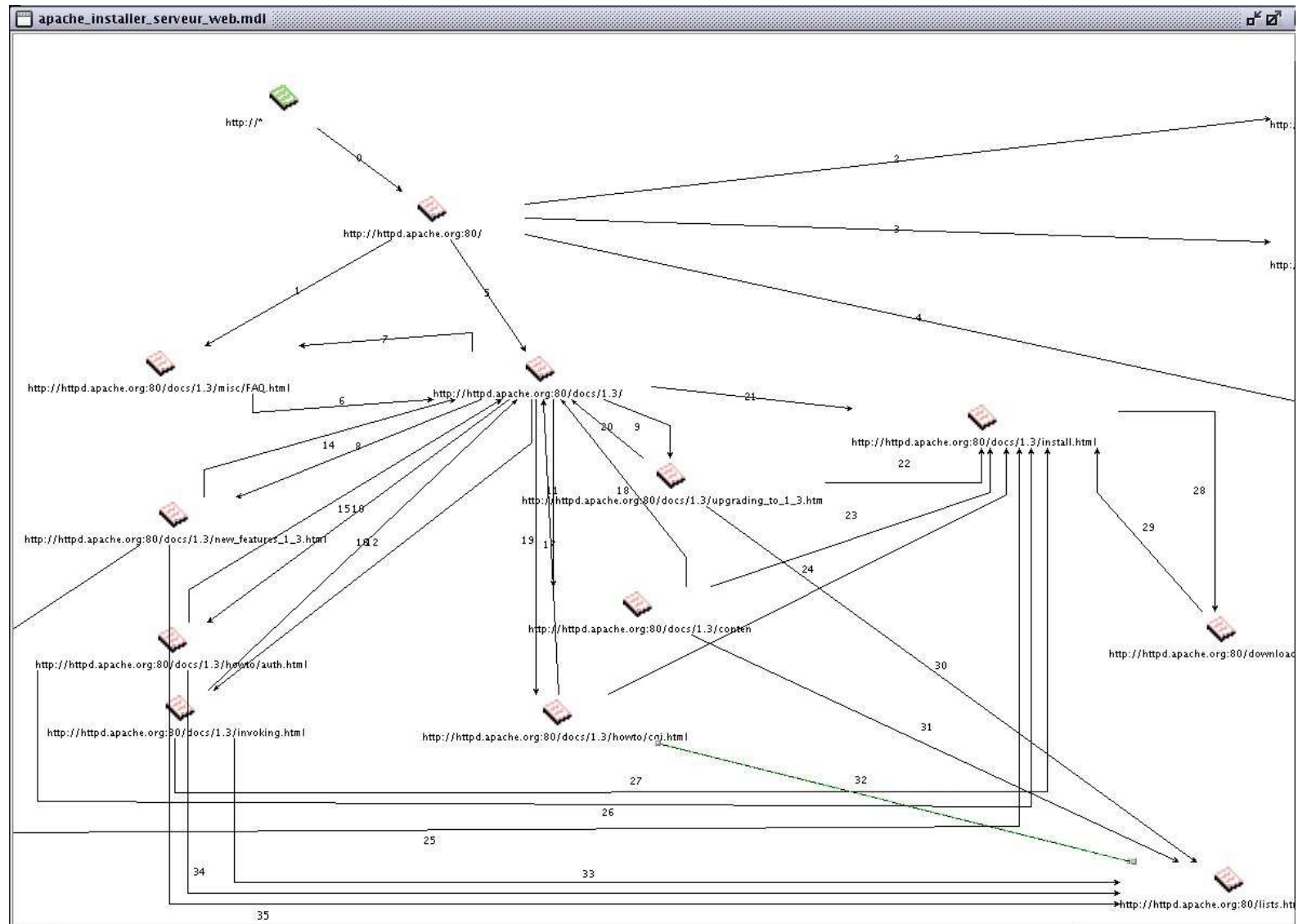


FIG. 3.1 – Exemple de modèle d'utilisation

3.1.3 Spectre des travaux

Proposer des conseils et juger de leur qualité (*i.e.*, répondre à la question “ces conseils vont-ils vraiment aider l'utilisateur?”) n'est pas chose facile [PT99]. Une telle étude impose en effet une analyse des actions des utilisateurs de sites Web pour déterminer, par exemple, les effets de la disposition des informations dans les pages, les effets de la structure des pages, les effets des différents types d'interactions avec l'utilisateur et propositions (pour plus de détails, voir, par exemple, [TPDB98]).

Dans cette thèse, nous ne cherchons pas à proposer un système conseiller “clés en main” qui pourrait être installé tel quel sur un système et offrir immédiatement des conseils. Nous cherchons à proposer une approche et une architecture permettant de construire un système conseiller flexible sur lequel il sera possible de s'appuyer pour élaborer et modifier des conseils. Ces créations et modifications de conseils pourront se fonder, par exemple, sur des travaux en ergonomie. Notre travail n'aborde pas ces derniers aspects.

Notons de plus que de nombreux travaux ont étudié les utilisateurs de sites Web, leurs difficultés et les conseils qui pouvaient leur être bénéfiques (voir, par exemple, [BES97] [BL98] [Che05] [JH05] [BSS06]). Nous nous sommes appuyés sur ces travaux pour composer des exemples de conseils lors du développement des prototypes. Cependant, nous n'avons pas abordé, en tant que telle, l'utilisation effective des prototypes de systèmes conseillers construits.

3.2 Domaines scientifiques abordés

Au cours de cette thèse nous avons abordé, à différents niveaux, plusieurs domaines scientifiques.

3.2.1 Domaines scientifiques auxquels nous avons contribué

- Les systèmes conseillers (cf. 2.1.1). C'est le thème central de la problématique.
- Les hypermédias adaptatifs (cf. 2.4.3). La problématique des travaux sur les hypermédias adaptatifs est la suivante : il s'agit de s'interroger sur l'élaboration d'approches permettant de proposer des hypermédias capables de s'adapter à leurs utilisateurs. L'adaptation peut être fonction de caractéristiques telles que le niveau des utilisateurs, leurs connaissances, leur expérience, etc. (voir, par exemple, les travaux de Brusilovsky [Bru96] [BM02])

[BBS03] [BB04]). Notre problématique est donc assez proche, puisque, si l'on observe le fonctionnement d'un système conseiller du point de vue de ses utilisateurs (*i.e.*, on observe les pages du site Web consultées et les conseils qui leur sont attachés), on note une modification, une adaptation de l'environnement de l'utilisateur (adaptation qui est, en l'occurrence, fonction de la tâche de l'utilisateur). Nos travaux sont donc pleinement dans les thématiques relatives aux hypermédias adaptatifs. Nous avons d'ailleurs publié un article dans la revue "Hypermedia and Multimedia" lors d'un numéro spécial consacré aux hypermédias adaptatifs ("adaptive hypermedia in the Age of the Adaptive Web") [RT04b].

3.2.2 Domaines scientifiques en relation avec nos travaux

- L'Ingénierie des Connaissances et l'Intelligence Artificielle. Notre problématique amène à construire des modèles des tâches qui représentent les tâches que peuvent vouloir réaliser les utilisateurs d'un site Web lorsqu'ils y accèdent. Ce type de mécanisme fait partie des thématiques abordées dans ces domaines. Nous avons ainsi publié un article aux journées francophones d'Ingénierie des Connaissances en 2004 [RT04a].
- Les environnements informatiques pour l'apprentissage humain (EIAH). "Un Environnement Informatique pour l'Apprentissage Humain (EIAH) est un environnement conçu dans le but de favoriser l'apprentissage humain" [TBB⁺04]. Ce domaine de recherche est particulièrement pluridisciplinaire puisqu'il touche des domaines tels que la pédagogie, la psychologie cognitive, les sciences de l'éducation et l'informatique [Tch02]. Notre problématique impose un suivi, un guidage de l'utilisateur lors de son parcours d'un site Web, guidage qui est fonction de modèles de tâches prédéfinis. Or, cette situation est proche de certaines problématiques étudiées par les EIAH.

3.2.3 Domaines scientifiques dans lesquels nous avons puisé des techniques et/ou dont nous nous sommes inspirés

Notre travail n'apporte pas de contribution directe aux domaines scientifiques cités ci-dessous. Nous avons utilisé ces domaines pour y puiser des techniques et/ou des idées.

- La fouille de données ("Data Mining") et plus particulièrement la recherche d'usages particuliers dans les logs des serveurs Web ("Web usage mining" - cf. 2.4.2.2). Il s'agit pour ces travaux d'analyser des données pour tenter

d'en extraire des informations qui pourront ensuite être utilisées. Notre problématique nous amène à prendre en compte la tâche de l'utilisateur. Pour déterminer quels peuvent être le ou les objectifs des utilisateurs qui accèdent à un site Web, une technique est d'analyser les logs du site dans le but d'en extraire les usages récurrents.

Ainsi, un des prototypes que nous avons construits utilise certaines de ces techniques pour mettre en évidence des usages récurrents. Il s'agit cependant plus d'une recherche d'usages particuliers, assistée par des outils informatiques, que d'une découverte automatique d'usages prototypiques du site. La découverte automatique d'usages prototypiques d'un site Web n'est pour l'instant pour nous qu'une perspective.

- Les techniques de construction de profils utilisateurs ("User Modelling techniques" - cf., par exemple, 2.4.2.1). Il s'agit pour ces techniques de créer puis d'utiliser des profils utilisateurs. Notre problématique n'aborde pas, *a priori*, ce domaine mais :
 1. Ces techniques sont largement utilisées par les systèmes conseillers (cf. chapitre 2) et la compréhension du fonctionnement de ces systèmes imposait d'étudier ces questions.
 2. Guider l'utilisateur impose de prendre en compte son parcours. Dès lors, des profils, même très sommaires (présentant juste l'historique de navigation), devront être créés.
- Les Sciences Cognitives. Pour guider l'utilisateur dans un site complexe, il est nécessaire de répondre à des questions telles que :
 - Quels conseils proposer ?
 - Quelle explication donner ?
 - Quelle fonctionnalité faut-il proposer à un instant donné de la navigation d'un utilisateur ayant un objectif donné ?

Il s'agit dans cette thèse de proposer une approche et une architecture permettant l'élaboration d'un système conseiller sur lequel un spécialiste en sciences cognitives pourra s'appuyer pour construire puis proposer des conseils. Pour illustrer notre travail, nous avons construit des prototypes de systèmes conseillers ainsi que des exemples de conseils. Dans ce but, nous nous sommes intéressés à des travaux relatifs aux sciences cognitives, à l'ergonomie des systèmes informatiques ou encore aux interfaces Homme-Machine. Cependant, ce travail de thèse n'aborde pas en tant que tel cet aspect de la construction des conseils.

3.3 Problématique et état de l’art

Au chapitre 2, nous avons présenté différents types de systèmes conseillers et nous les avons structurés, entre autres, en fonction du type de l’aide apportée, distinguant ainsi trois classes :

- Apporter directement de l’information supplémentaire à l’utilisateur,
- Simplifier la navigation de l’utilisateur,
- Guider l’utilisateur.

Nous allons, dans ce paragraphe, passer en revue ces trois types d’aide et étudier les relations entre ces approches et notre problématique.

3.3.1 Apporter directement de l’information supplémentaire à l’utilisateur

Il s’agit pour les systèmes apportant directement de l’information supplémentaire à l’utilisateur de lui proposer un supplément d’information. Ce supplément d’information peut être de différentes natures. Par exemple, des liens vers des pages susceptibles d’intéresser l’utilisateur peuvent être proposés. Ces systèmes utilisent différents mécanismes pour repérer les domaines qui semblent intéresser l’utilisateur (ces mécanismes sont, par exemple, un questionnaire, une analyse du parcours de l’utilisateur, une analyse des pages qu’il consulte, etc.). Les centres d’intérêt de l’utilisateur ainsi détectés, sont ensuite utilisés pour permettre la proposition d’informations les plus pertinentes possibles.

Revenons à l'exemple cité plus haut (cf. paragraphe 3.1.2), c'est-à-dire, à un utilisateur accédant au site Web Apache et dont le but général est d'"installer un serveur Web Apache".

Comme nous l'avons vu au paragraphe 2.4.1.2, des ontologies peuvent être utilisées pour chercher à déterminer les centres d'intérêt de l'utilisateur et permettre à un système conseiller de proposer un supplément d'information. Dans notre exemple, l'ontologie est utilisée pour associer aux pages parcourues par l'utilisateur, des termes tels que :

- possibilités
- limites
- modalités de téléchargement
- étapes d'installation
- etc.

Ce sont ces mots clés qui seront utilisés pour rechercher des pages susceptibles d'intéresser l'utilisateur. Il y a donc transformation d'un objectif large ("installer un serveur Web Apache") en des séries de termes reflétant chacune un sous-objectif.

Cet exemple illustre une différence entre la problématique qui sous-tend cette approche et la nôtre : dans cette approche, on observe une traduction de la tâche de l'utilisateur en données intelligibles par le système. Ainsi, dans l'exemple ci-dessus, le but de l'utilisateur est traduit en un ensemble de mots clés. On peut donc considérer qu'il y a une certaine prise en compte du but de l'utilisateur mais que cette prise en compte n'est que partielle puisqu'elle n'est effective qu'à travers une traduction en mots clés. Dans notre problématique, la notion de tâche est considérée différemment puisque considérée telle quelle, c'est-à-dire sans transformation (ainsi, dans l'exemple présenté page 53, un modèle de tâches a été construit pour assister l'utilisateur dans la réalisation de la tâche "installer un serveur Web Apache". Cependant, même si pour construire ce modèle, l'expert a décomposé la tâche générale en sous-tâches, il n'y a pas eu traduction de la notion de tâche en un autre formalisme).

D'autres approches proposant ce type d'aide se fondent sur la comparaison de comportements utilisateurs en utilisant, par exemple, des techniques de "collaborative filtering" (cf. paragraphe 2.4.2.1).

Si on reprend l'exemple d'un utilisateur accédant au site Web Apache et dont le but général est d'"installer un serveur Web Apache", les systèmes fondés sur ces techniques vont, dans un premier temps, observer la navigation des utilisateurs. Soit un utilisateur A ayant consulté :

- la page "features.html" présentant les caractéristiques du serveur Web Apache,
- la page "download.html" présentant les modalités de téléchargement du serveur Web Apache,
- la page "install.html" présentant les étapes à suivre pour installer un serveur Web Apache.

Si un utilisateur B consulte les pages "features.html" et "download.html", le système va lui proposer un lien vers la page "install.html".

Ce second exemple illustre, une nouvelle fois, le fait que les approches proposant ce type d'aide ne prennent en compte qu'indirectement la notion de tâche : l'utilisateur A, dans le but de réaliser l'objectif qui est le sien, va parcourir un certain nombre de pages. L'utilisateur B, en parcourant sensiblement les mêmes pages que A se verra proposer des liens vers des pages parcourues par A et qu'il n'a pas encore parcourues. Les systèmes issus de cette approche considèrent donc que ces deux utilisateurs ont des objectifs similaires. C'est cette approximation qui différencie la problématique qui sous-tend ces systèmes et la nôtre, puisque dans le cadre de notre problématique, la notion de tâche est prise en compte explicitement.

3.3.2 Simplifier la navigation de l'utilisateur

Il s'agit pour les systèmes dont le but est de faciliter la navigation des utilisateurs de sites Web, de modifier les pages Web proposées aux utilisateurs (par exemple, comme cela été présenté au paragraphe 2.3.2, en supprimant des liens). Ces approches prennent en compte l'objectif de l'utilisateur et ses caractéristiques (centres d'intérêt, connaissances, etc.).

On observe donc des similitudes avec notre problématique : comme dans le cadre de notre problématique, l'objectif de l'utilisateur, lorsqu'il accède au site Web, va être pris en compte. Par exemple, des systèmes comme AHA! prennent en compte le but de l'utilisateur en proposant des pages qui correspondent aux notions que l'étudiant souhaite maîtriser.

On observe cependant une différence majeure : l'aide proposée n'est pas de type guidage : certes, des pages sont adaptées, des listes de liens sont réorganisées mais il s'agit plus de proposer à l'utilisateur une organisation séquentielle des

pages qu'il doit parcourir, que des éléments de guidage proprement dit.

Enfin, on observe que les travaux qui apportent une telle aide sont principalement des approches proposant à un utilisateur des cours en ligne. Les tâches prises en compte dans ce type de systèmes sont donc semblables (on trouve, par exemple, des tâches comme “j'apprends l'anglais : le présent simple et progressif” ou “j'apprends l'anglais : les modaux”). Or, notre problématique se différencie de celle sous-jacente à ces systèmes puisque nous ne considérons, *a priori*, aucune limitation dans le type de tâches.

3.3.3 Guider l'utilisateur

L'objectif des systèmes proposant de guider l'utilisateur est de lui permettre de réaliser son objectif. Les systèmes tels que Pixed [HM02] [HFM04] ou Ardeco [CPM02] issus de l'approche Musette [CPM03] [LCPM05] prennent en compte le parcours et/ou la tâche de l'utilisateur. L'objectif de Pixed [HFM04] est d'aider l'utilisateur à réaliser son but, en l'occurrence acquérir, en ligne, des connaissances. L'aide proposée est de haut niveau : il est proposé à l'utilisateur des chemins vers la réalisation de son objectif. Des séquences d'apprentissages sont enregistrées et des pages annotées. Ces éléments seront ensuite utilisés pour proposer une aide. C'est l'utilisateur qui choisit lui même son but en sélectionnant, dans une ontologie, le concept qu'il veut étudier.

Les travaux sur Musette et ses différentes mises en oeuvre présentent une problématique proche de la nôtre puisque :

1. Il va s'agir de prendre en compte le but de l'utilisateur et permettre à l'utilisateur de réaliser son objectif. Comme dans notre problématique, cette approche cherche à formaliser le comportement de l'utilisateur plutôt que l'utilisateur lui même.
2. L'accomplissement de l'objectif de l'utilisateur va être assisté par le système via la proposition de conseils permettant un guidage de l'utilisateur.

On observe cependant plusieurs différences :

1. Dans notre approche l'aide proposée est élaborée par un expert. C'est lui qui définit les conseils à proposer et les facteurs déclenchants. Dans les travaux issus de l'approche Musette, les conseils sont élaborés par les experts lors de l'initialisation du système mais aussi par les utilisateurs. Ces conseils sont construits par le biais d'annotations et présentés aux utilisateurs lorsque la navigation annotée correspond à la navigation de l'utilisateur.

2. Nous proposons d'associer à un site Web complexe, un système conseiller évitant de modifier celui-ci en profondeur. Dans les travaux issus de l'approche Musette, la problématique est différente puisque système conseiller et site Web ne font qu'un : il y a un système qui propose à la fois les pages et l'aide associée.

3.4 Déroulement de la recherche

Dans cette thèse, nous avons :

1. Conçu une approche qui prenne en compte les différentes caractéristiques de notre problématique et qui permette ainsi :
 - La création d'un système conseiller ayant une architecture épiphyte et fonctionnant dans des sites Web complexes.
 - La prise en compte du but de l'utilisateur et sa modélisation.
 - La proposition à l'utilisateur d'une aide de type guidage qui soit fonction du but de l'utilisateur.
2. Proposé différentes étapes nécessaires à la construction d'un tel système conseiller.
3. Proposé une architecture fonctionnelle.
4. Elaboré un premier prototype qui a permis :
 - De montrer la faisabilité de notre approche.
 - De détecter les défauts et les insuffisances de notre approche et de notre architecture.
5. Elaboré un second prototype qui a permis :
 - De modifier notre approche pour en corriger certains défauts.
 - De modifier notre architecture.
 - De mettre en lumière de nouvelles insuffisances et ainsi envisager des évolutions à ce travail.

3.5 Terrains applicatifs

Ces travaux se sont appuyés sur deux terrains applicatifs : le site Web Educa-source et le site Web de la fondation Apache.

3.5.1 Educasource

Educasource [Edu] est un site du ministère de l'Education Nationale et de la Recherche. Il propose à un public d'enseignants un ensemble de descripteurs de ressources, ressources qui peuvent être utilisées dans l'élaboration des cours. Après une année d'utilisation, les usages du site ont été analysés par un cabinet de conseil. Des dysfonctionnements ont été mis en évidence. Le LIUM a signé un contrat avec le ministère de l'Education Nationale et de la Recherche. L'objectif du contrat était de proposer une approche et un prototype permettant de corriger les dysfonctionnements.

Une approche a été proposée et un prototype construit. Cependant, pendant ces travaux, les objectifs des concepteurs du site Educasource ont radicalement changé, ne nous permettant pas d'expérimenter notre approche.

3.5.2 Apache

Le site de la fondation Apache [Apa] présente les différents projets de la fondation Apache et notamment celui du serveur Web Apache (serveur Web le plus utilisé dans le monde). Ce site est un site extrêmement complexe : il présente de nombreuses pages, une architecture complexe, différentes fonctionnalités (cf. paragraphe 3.1.2). C'est donc un terrain applicatif idéal pour mettre en oeuvre notre système conseiller. Nous avons signé en 2005 une convention avec la fondation Apache nous donnant accès à 16 mois de logs de leurs serveurs. L'analyse de ces logs a permis de disposer de données pour étudier notre approche.

Un prototype a été élaboré ainsi que plusieurs modèles de tâches. Ce prototype et ces modèles ont permis de montrer la faisabilité de notre approche.

3.6 Synthèse

Dans ce chapitre, nous avons présenté la problématique de ce travail de thèse. Il s'agit d'étudier l'élaboration d'un certain type de systèmes conseillers :

1. Ces systèmes peuvent fonctionner sans qu'il soit nécessaire de modifier le site cible.
2. Ces systèmes proposent une aide de type guidage.
3. Ces systèmes prennent en compte le but de l'utilisateur.
4. Ces systèmes sont dédiés à des sites Web complexes.

Les questions de recherche liées à cette problématique sont multiples : quelle architecture composer ? Quelles données utiliser ? Comment construire des modèles de tâches ? Comment y associer des conseils ? Comment proposer ces conseils à l'utilisateur ? Quelle méthodologie adopter pour l'élaboration du système conseiller ? C'est à ces questions que nous avons travaillé.

En terme de système, notre objectif a été de développer un ou plusieurs prototypes de système conseiller et de montrer la faisabilité de notre approche.

Comme nous l'avons indiqué dans l'introduction, Platon 1 [TBB⁺04] offre un cadre de réflexion sur les travaux en EIAH à travers différentes dimensions. Deux dimensions ont été présentées dans l'introduction (résultat de la recherche, tableau 1.1, et cycle de vie de la recherche, tableau 1.2). Nous présentons, ci-dessous, deux autres dimensions relatives à la définition du projet de recherche (tableau 3.1) et au cadre théorique de la recherche (tableau 3.2).

[A] Définition du projet de recherche	
[A.1] Objectifs de la recherche	Contribution à l'élaboration des systèmes conseillers fondés sur les principes suivants : <ul style="list-style-type: none"> - Prise en compte de la tâche de l'utilisateur. - Aide de type guidage. - Installation du système conseiller sans modification du site cible. - Site cible complexe.
[A.2] Contraintes sur la construction du système informatique	(1) Élaborer une architecture fonctionnelle. (2) Disposer d'un prototype utilisable.
[A.3] Finalités du système informatique	Prototype de système conseiller permettant (1) de montrer la faisabilité de notre approche et (2) de l'illustrer.

TAB. 3.1 – *Notre problématique vue sous l'angle de l'approche méthodologique Platon-1 - Définition du projet de recherche*

[B] Cadre théorique de la recherche	
[B.6] Théorie de la conception	Nous avons adopté un processus de conception itératif : après avoir défini la problématique, nous avons proposé une approche puis conçu une première architecture sur laquelle nous nous sommes fondés pour construire un prototype illustrant l'approche et sa faisabilité. La construction puis l'analyse de ce prototype nous ont conduit à faire évoluer notre approche et proposer une deuxième architecture, puis un deuxième prototype fondé sur la nouvelle architecture.

TAB. 3.2 – *Notre problématique vue sous l'angle de l'approche méthodologique Platon-1 - Cadre théorique de la recherche*

Chapitre 4

Approche générale proposée : une architecture épiphyte prenant en compte la tâche de l'utilisateur

Dans ce chapitre, nous présentons les principes de l'approche que nous proposons en restant volontairement au niveau des principes généraux. Cette approche peut, en effet, être mise en oeuvre de différentes façons. Aux chapitres 5 et 6, nous présentons l'opérationnalisation particulière de cette approche que nous avons réalisée.

4.1 Principes généraux

Nous proposons une approche qui permet l'élaboration d'un système conseiller capable de guider l'utilisateur en prenant en considération sa tâche.

Nous avons fondé cette approche sur plusieurs principes :

1. L'architecture du système conseiller élaboré est épiphyte,
2. Des modèles de tâches, explicitant les usages prototypiques du site étudié, sont identifiés,
3. Des conseils sont construits et associés aux modèles de tâches (l'association de conseils à un modèle de tâches crée un nouveau type de modèle que nous appellerons modèle d'utilisation),
4. Les modèles d'utilisation créés sont utilisés par le système conseiller pour analyser la navigation de l'utilisateur et permettre la proposition, en temps

réel, des conseils, conseils présentés dans une fenêtre qui s'ouvre en superposition par rapport à celle du site.

Dans ce paragraphe, nous allons, dans un premier temps, proposer les définitions des notions qui nous seront utiles dans la suite de ce travail, puis nous détaillerons les principes présentés ci-dessus.

4.1.1 Définitions

Graphe du site. Le graphe du site est l'ensemble des pages et des liens proposés par le site Web. Chaque noeud du graphe correspond à une page Web. Chaque arc du graphe correspond à un lien hypertexte, c'est-à-dire à une action de l'utilisateur lui permettant de passer d'une page à une autre. Le graphe d'un site Web, notamment dans le cas d'un site complexe, peut être extrêmement large et peut nécessiter l'utilisation d'outils informatiques pour créer, manipuler, afficher, les noeuds et les arcs. La figure 4.1 présente, en second plan, un exemple de graphe de site.

Modèle de tâches. Un modèle de tâches est un sous ensemble du graphe du site. Un modèle de tâches correspond à une modélisation des différentes actions de navigation au sein du site nécessaires à la réalisation d'une tâche. Précisons, qu'un modèle de tâches peut, le cas échéant, modéliser différentes façons de réaliser la tâche (*i.e.*, modéliser différents cheminements).

Exemple de modèle de tâches : pour reprendre l'exemple du site Apache, les noeuds grisés et les arcs qui les relie de la figure 4.1 modélisent un parcours dans le site Web Apache permettant la réalisation de la tâche "installation du serveur Web Apache".

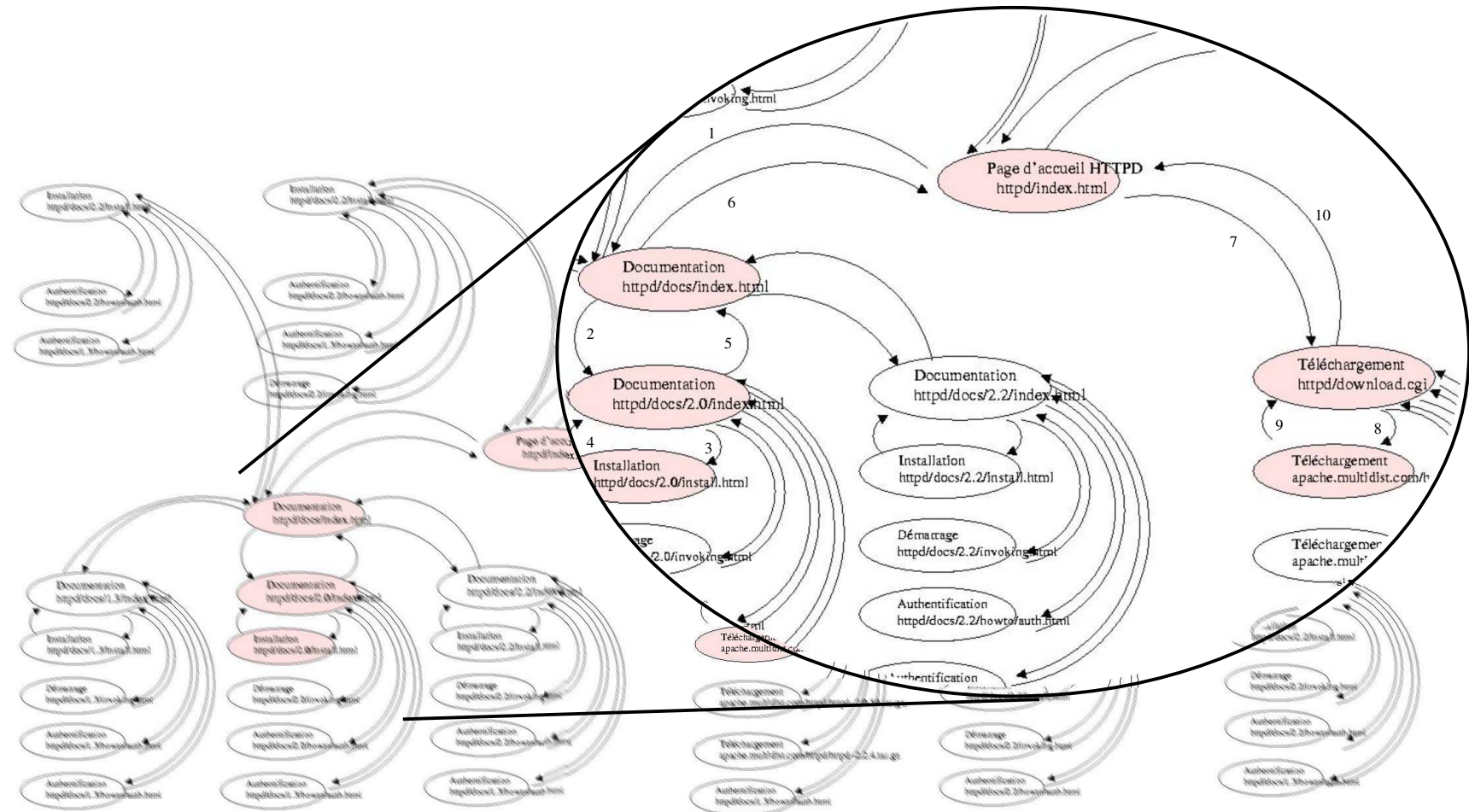


FIG. 4.1 – Exemple de modèle de tâches (l'ordre des transitions est indiqué par les chiffres). En second plan, un exemple de graphe de site.

Conseil. Un conseil est un ensemble d'informations présentées à l'utilisateur. Il peut comprendre du texte, des liens vers d'autres pages ou vers des fonctionnalités proposées par le site Web ou propres au système conseiller.

Les figures 4.2 et 4.3 présentent des exemples de conseil.

Les conseils sont déclenchés lorsque l'utilisateur clique sur un lien. C'est pourquoi, dans nos modèles, nous avons choisi de lier conseils et arcs (qui, rappelons le, représentent chacun une action, c'est-à-dire un clic de l'utilisateur sur un lien HTTP qui permet de passer d'une page d'un site Web à une autre). A chaque arc, il est possible d'associer un conseil. Lorsque l'utilisateur clique sur un lien, le système conseiller présente le conseil attaché à l'arc modélisant le lien (si un conseil a été attaché). Il est à noter que le conseil peut comporter différents types d'information et d'aide et que certains éléments peuvent être mis à jour dynamiquement. Enfin, précisons que des mécanismes plus complexes de déclenchement des conseils peuvent être mis en oeuvre (par exemple, la présentation d'un conseil peut être conditionnée à un historique particulier : le conseil n'est présenté que si l'utilisateur a parcouru les pages A, B et C).

Attacher des conseils aux arcs d'un modèle de tâches amène à construire un nouveau type de modèle que nous appellerons modèle d'utilisation.

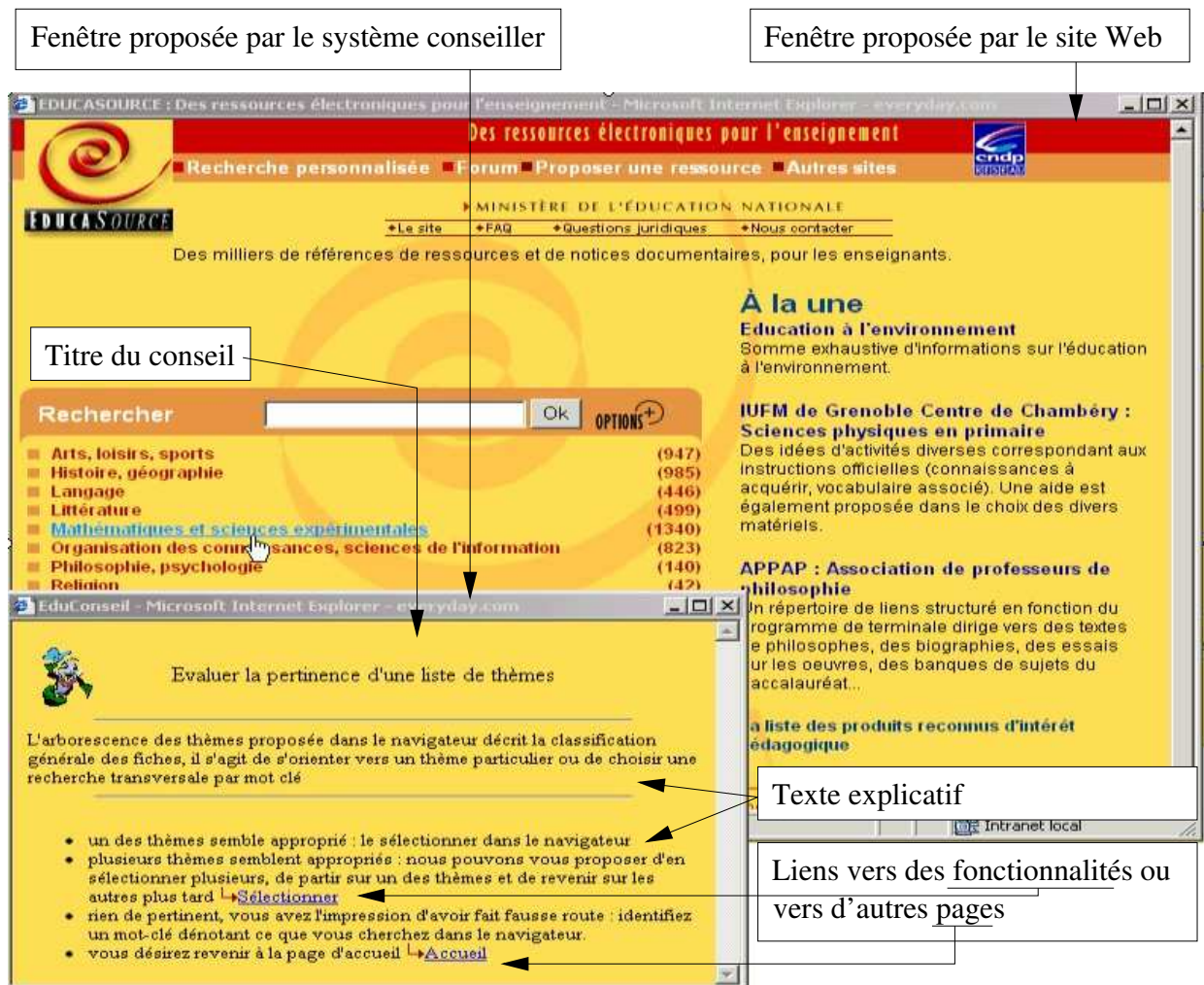


FIG. 4.2 – Exemple de conseil



FIG. 4.3 – Exemple de conseil

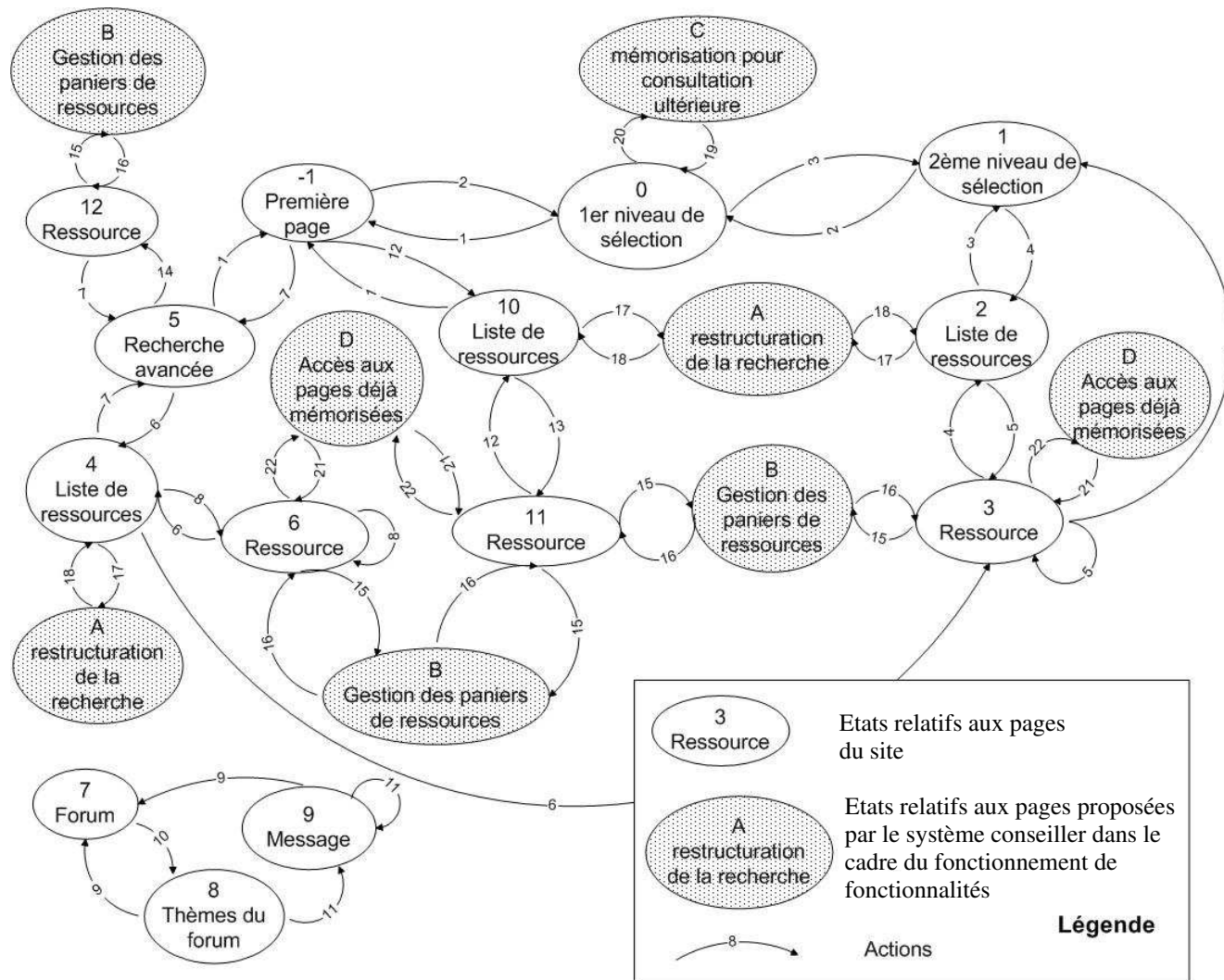


FIG. 4.4 – Exemple de modèle d'utilisation.

Modèle d'utilisation. Un modèle d'utilisation est un modèle de tâches ayant subi quelques transformations :

- Des conseils ont été ajoutés aux arcs. Comme nous l'avons indiqué précédemment, les conseils sont liés aux arcs du modèle. A chaque arc, il est possible d'attacher au maximum un conseil. Lorsque l'utilisateur clique sur un lien, le conseil, attaché à l'arc modélisant le lien, est présenté.
- Certains noeuds ont pu être regroupés. Certains conseils présentés aux utilisateurs sont identiques lorsque l'utilisateur accède à une page A ou à une page B. Pour profiter de cet état de fait et ainsi limiter la taille des modèles d'utilisation, chaque noeud du modèle d'utilisation peut être associé à plusieurs URLs. La figure 4.5 présente ces transformations.

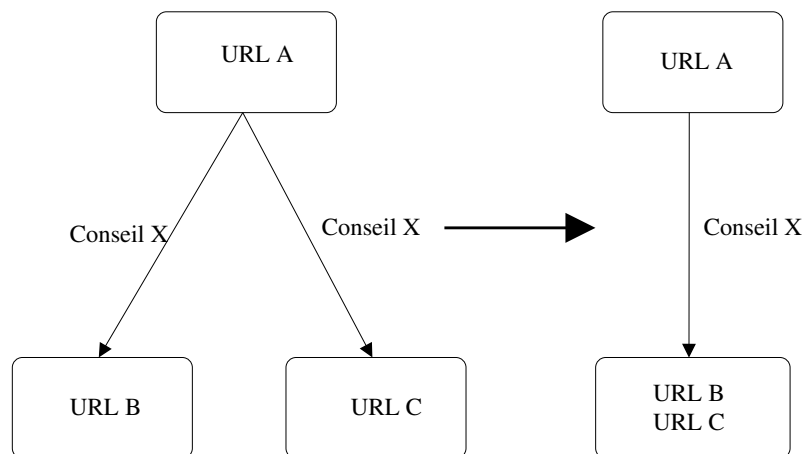


FIG. 4.5 – *Fusion de noeuds du modèle d'utilisation.*

Les modèles d'utilisation, comme les modèles de tâches, sont des graphes. La figure 4.4 présente un exemple de modèle d'utilisation. Une fois le système conseiller opérationnel, trois types de pages sont proposés aux utilisateurs au cours de leurs navigations :

- Des pages proposées par le site Web (par exemple, la page proposée en second plan figure 4.3). Ces pages sont modélisées par les noeuds blancs de la figure 4.4.
- Des pages de conseils proposées par le système conseiller et associées à des pages proposées par le site Web (par exemple, la page proposée en premier plan figure 4.3). Ces pages, toujours associées à des pages proposées par le site Web, ne sont pas représentées dans les modèles d'utilisation.

- De nouvelles pages créées ou transformées par le système conseiller (c'est-à-dire non associées à des pages proposées par le site Web). Ces pages permettent la mise en oeuvre de certaines fonctionnalités. La figure 4.6 présente un exemple de ce type de pages :
 - En second plan, nous présentons la page proposée par le site Web qui a servi de base à la construction de la page proposée par le système conseiller. Cette page n'est pas présentée à l'utilisateur : elle sert simplement à la construction de la page proposée par le système conseiller.
 - En premier plan, la page qui est proposée à l'utilisateur et qui correspond à la transformation de la page initiale pour y ajouter une fonctionnalité (en l'occurrence, sélectionner plusieurs thématiques jugées intéressantes pour y revenir plus tard).



FIG. 4.6 – Exemple de conseil nécessitant l'ouverture d'une page spécifique.

Graphe du système conseiller. Le graphe du système conseiller est l'union de plusieurs graphes : le graphe du site et les graphes des différents modèles d'utilisation. Il modélise l'ensemble des parcours qu'il est possible d'effectuer dans le site ainsi que les pages propres au système conseiller et proposées dans le cadre de la mise en oeuvre de fonctionnalités proposées par le système conseiller.

Trace d'utilisation. Une trace d'utilisation est un sous-ensemble du graphe du système conseiller correspondant aux pages et aux liens parcourus effectivement par un utilisateur au cours d'une session.

4.1.2 Une architecture épiphyte

Différentes architectures peuvent être mises en place lors de l'élaboration d'un système conseiller. Certains systèmes proposent des conseils aux utilisateurs parcourant le Web quel que soit le site visité. Ils sont, bien entendu, indépendants du site visité (n'étant pas dédiés à un site particulier) et sont placés soit du côté client (à côté du navigateur ou dans un navigateur *ad hoc*), soit sur un serveur dédié (cf., par exemple, le paragraphe 2.1.2 présentant les systèmes conseillers et notamment WebMemex [MTCGdGP03]). Comme nous l'avons expliqué dans le chapitre précédent, ce n'est pas à ce type de systèmes que nous nous intéressons, mais plutôt aux systèmes qui prodiguent des conseils aux utilisateurs d'un site particulier. Les systèmes mettant en oeuvre cette approche ont différentes architectures :

1. Une architecture intégrée. Le système conseiller peut être complètement intégré dans le site pour lequel il doit proposer des conseils. Citons, par exemple, CoCoA [AAM01], les systèmes FindMe [BHY96] [Bur02a] ou encore AHA! [BBS03] [BC98], systèmes que nous avons déjà présentés au chapitre 2. Dans ce type de système, le système conseiller et le système permettant de proposer des pages (par exemple, un serveur Web) ne font qu'un : c'est un système unique, créé pour analyser le comportement des utilisateurs, utiliser des données qui lui sont propres puis proposer des pages.
2. Une architecture couplée. Le système conseiller est semi-indépendant : il fonctionne "à côté" du système proposant les pages. Cependant, son mode de fonctionnement nécessite l'importation de données, données que possède le système proposant les pages. On peut citer à titre d'exemple, des systèmes comme celui proposé par [MCS00] : dans ce système, la proposition de recommandations est fondée sur des techniques d'analyses des logs du serveur Web et l'analyse de ces données impose de coupler les deux

systèmes (la vitesse des réseaux informatiques peut autoriser une certaine distance physique entre les systèmes mais, dans ce type d'architecture, les deux systèmes, même distants physiquement, restent proches conceptuellement).

3. Une architecture épiphyte. C'est ce type d'architecture qui nous intéresse plus particulièrement. Il a été utilisé dans des travaux tels que ceux sur Epi-Talk [GPPG95]. Il s'agit de greffer le système conseiller sur le serveur Web qui propose des pages pour lesquelles des conseils doivent être proposés (nous l'appellerons par la suite "le serveur cible"). Le terme "greffe" n'implique pas de notion de proximité physique entre le système conseiller et le serveur Web cible. Au contraire : il s'agit de faire fonctionner les deux systèmes en conservant une certaine distance entre eux. Cette distance entre les deux systèmes est de deux ordres : physique et conceptuelle. Physique car l'architecture du système épiphyte que nous proposons doit permettre un fonctionnement à distance du système conseiller. Conceptuelle car le système conseiller est un système à part entière et doit pouvoir être installé et fonctionner sans qu'il soit nécessaire de modifier le serveur Web cible. Il y a bien deux systèmes distincts, deux conceptions différentes, ayant chacune leurs objectifs. La figure 4.7 illustre cette architecture et en présente un schéma simplifié.

Une architecture épiphyte présente plusieurs avantages :

- **Généricité.** La séparation effective entre le système conseiller et le serveur Web cible permet de créer un système conseiller composé d'éléments génériques qu'il est possible de faire fonctionner quel que soit le site cible, et de limiter la présence d'éléments spécifiques dédiés à un site particulier :
 - **Éléments génériques.** Les éléments génériques sont des modules qui ne dépendent pas du système cible. Il s'agit, par exemple, des composants qui touchent à la gestion des modèles : les modèles changent en fonction des sites Web cibles mais, du point de vue du système conseiller, leur gestion est identique.
 - **Éléments spécifiques.** Les éléments spécifiques doivent être plus ou moins modifiés, adaptés au site Web cible avant de pouvoir fonctionner. Ce sont, par exemple, des fonctionnalités particulières absentes du site cible et offertes par le système conseiller mais qui restent dédiées à ce site. Faire fonctionner le système conseiller avec un nouveau site cible nécessite donc d'adapter certaines fonctionnalités ou d'en redévelopper de nouvelles.

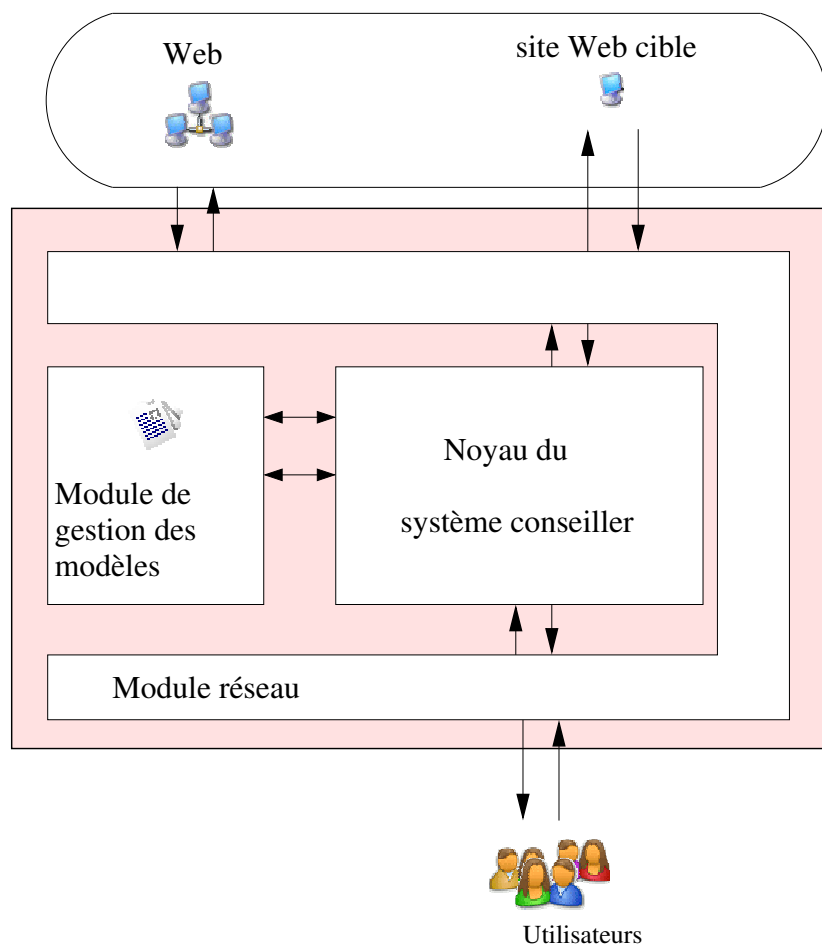


FIG. 4.7 – Schéma simplifié de l'architecture proposée.

- **Utilisabilité.** L'architecture épiphyte permet d'ajouter un système conseiller à un site Web cible sans qu'il soit nécessaire de le modifier. A l'inverse, cette architecture permet d'arrêter le système conseiller, si besoin, sans conséquence aucune pour le serveur Web cible.

Par ailleurs, l'architecture épiphyte permet également de voir les sites Web sous un angle différent : on peut imaginer créer un site Web minimaliste (*i.e.*, présentant une base minimale d'information et dont l'utilisabilité convient à tous les types d'utilisateurs) auquel on associe un système conseiller. Les modèles d'utilisation, qui contiennent l'information relative au parcours pour lequel l'utilisateur est accompagné, et les conseils peuvent alors être perçus comme des extensions logiques du site : chaque modèle d'utilisation propose (1) un parcours dans le site en fonction d'un objectif déterminé, (2) des conseils spécifiques au parcours et à l'utilisateur et (3) des fonctionnalités appropriées. La création d'un nouveau modèle d'utilisation peut être alors perçue comme la création d'un nouveau site dédié à un objectif donné. La figure 4.8 illustre cette vision du couple système conseiller - site cible.

- **Portabilité.** La dissociation système conseiller / serveur cible amène à une portabilité du système conseiller : le système conseiller peut fonctionner sur un système hôte différent du système hébergeant le serveur Web site cible. Le système hôte choisi fournit uniquement du temps CPU et un accès réseau.
- **Évolutivité.** La dissociation du système conseiller et du serveur Web cible amène à construire deux systèmes distincts. Cette séparation permet :
 - De faire évoluer indépendamment et facilement, l'un ou l'autre des systèmes : les deux systèmes sont nettement différenciés, chacun ayant ses données propres et un objectif propre. Les évolutions du serveur Web cible imposent des modifications des couches "modèles" et "conseils" du système conseiller. Les évolutions du système conseiller et/ou les modifications des données qu'il manipule, ne nécessitent pas de modification du site Web. Il est ainsi possible de faire évoluer le système conseiller au fur et à mesure de l'évolution des usages constatés (par exemple, en ouvrant le champ d'action du système conseiller en guidant l'utilisateur vers d'autres sites, en lui proposant de nouvelles fonctionnalités, etc.) sans toucher au site Web lui même.
 - D'offrir un spectre d'action potentiellement large : de par son architecture épiphyte, la portée du système conseiller n'est pas limitée à un site unique. Plusieurs sites peuvent être considérés de concert, le système conseiller permettant ainsi la création d'une sorte de méta-site virtuel.

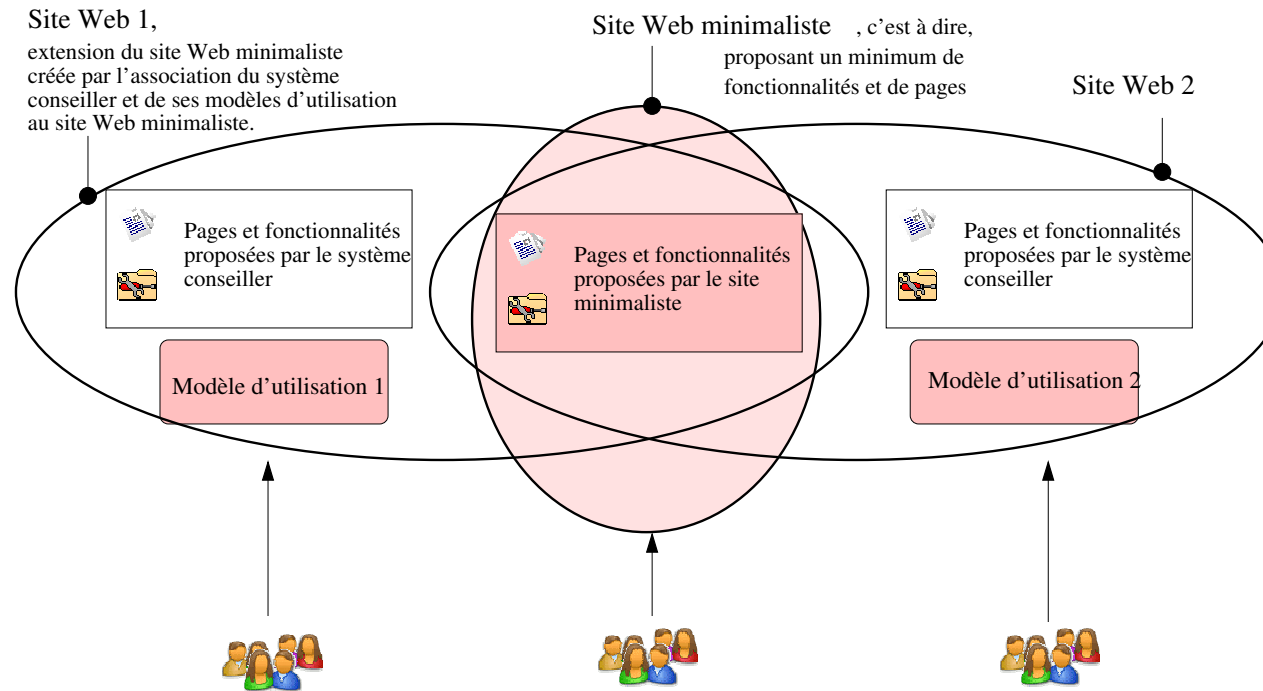


FIG. 4.8 – Extension d'un site Web minimaliste par ajout d'un système conseiller

4.1.3 Des modèles de tâches

4.1.3.1 Modéliser la tâche

Nous cherchons à prendre en compte la tâche de l'utilisateur (rappelons (cf. page 51), qu'une tâche correspond à un enchaînement d'actions ou d'opérations permettant la réalisation d'un but dans un environnement donné). Modéliser des tâches va permettre de considérer le but de l'utilisateur mais aussi les différentes actions à entreprendre pour réaliser cette tâche (les actions qu'un utilisateur peut réaliser sont, par exemple, choisir un chemin particulier dans le site, utiliser une fonctionnalité, etc.). Nous allons utiliser les graphes pour représenter les modèles de tâches.

Les modèles de tâches représentent différentes utilisations du site selon différents objectifs qui peuvent correspondre à ceux des utilisateurs ou ceux des concepteurs de sites :

- Du point de vue des utilisateurs. Les modèles de tâches représentent des utilisations ou des types d'utilisation du site considéré (cf. paragraphe 4.1.1). Nous nous intéressons plus particulièrement à des sites complexes qui permettent la réalisation de tâches elles aussi complexes et qui sont les plus à même de poser des problèmes aux utilisateurs. Les modèles de tâches sont utilisés pour modéliser ce type de tâches et permettre au système conseiller de proposer une aide à l'utilisateur.
- Du point de vue des concepteurs de site. Des usages imprévus du site peuvent apparaître. Ces usages peuvent satisfaire les concepteurs du site. Dans le cas contraire, ils peuvent être modélisés pour permettre aux concepteurs du site, via un guidage approprié, d'accompagner l'utilisateur vers la réalisation d'usages plus conformes à leurs attentes.

4.1.3.2 Identifier les tâches considérées par le système conseiller

De par la variété des tâches que peuvent réaliser les utilisateurs d'un site complexe, il ne s'agit pas de tenter de modéliser tous les usages possibles d'un site Web mais d'analyser les usages effectifs du site (par le biais de mécanismes tels qu'une analyse des logs, des interviews des utilisateurs, etc.) et d'en extraire les plus représentatifs. Ce sont ces usages du site, dits prototypiques, qui seront modélisés.

L'analyse des usages du site Apache et de ses logs nous a permis, par exemple,

d'identifier différents modèles de tâches :

- “Découvrir les outils proposés par la fondation Apache pour publier sur le Web”,
- “Installer un serveur Web Apache”,
- “Devenir membre de la fondation Apache”,
- “Participer au développement des projets XML de la fondation Apache”,
- “Lutter contre le spam sur un serveur de messagerie”,
- etc.

4.1.4 Des modèles d'utilisation

Les modèles d'utilisation sont créés en associant des conseils aux modèles de tâches (cf. paragraphe 4.1.1). Plus précisément, il s'agit de permettre le positionnement sur chaque arc de conseils. Rappelons que cette association est faite sur les arcs plutôt que sur les noeuds puisque les arcs représentent les actions de l'utilisateur : dans un hypermédia, les arcs d'un modèle de tâches représentent les liens hypertextes qui permettent de passer d'une page à une autre, ils modélisent donc les choix qui s'offrent aux utilisateurs (l'utilisateur peut passer de telle page à telle autre). Or, nous voulons être à même de proposer des conseils en fonction des actions de l'utilisateur, il est donc naturel d'associer arcs et conseils.

4.1.4.1 Création des conseils

Un conseil est une information proposée à l'utilisateur par le biais du système conseiller [PT99]. Nous avons choisi d'afficher le conseil dans une fenêtre adjacente à celle du site cible : il s'agit de proposer à l'utilisateur un environnement le plus clair possible lui permettant d'identifier rapidement ce qui provient du site cible et ce qui est de l'ordre du conseil. Dans ce but, nous proposons que deux pages soient proposées à l'utilisateur : d'un côté, une page contenant les informations envoyées par le site cible et de l'autre, celle produite par le système conseiller. Le système conseiller ne force en rien l'utilisateur : ce dernier peut ne pas tenir compte du conseil et fermer la fenêtre du système conseiller.

Les conseils proposés peuvent avoir plusieurs formes. Par exemple :

- Texte. Des recommandations, des explications sont proposées aux utilisateurs (expliquer une page, souligner l'existence d'une fonctionnalité, en exposer le fonctionnement, etc.). Des études sur l'utilisation des hypermédias (voir, par exemple, [TR04] ou [AT06]) ont montré que ce type de conseil

permet de limiter la charge cognitive que représente la lecture du site et ainsi d'aider l'utilisateur à se situer dans le site, à comprendre plus aisément les pages qui lui sont présentées et à ne pas perdre de vue son but.

- Liens. Des liens vers des pages du site Web considéré ou vers d'autres sites Web sont proposés aux utilisateurs. Toujours selon la littérature (par exemple, [JH05]), ce type de conseil peut permettre à l'utilisateur de naviguer plus facilement et/ou plus rapidement dans le site. Le but est de proposer à l'utilisateur une navigation moins redondante, permettant ainsi une accessión plus rapide aux données recherchées.
- Fonctionnalités. Des fonctionnalités sont proposées aux utilisateurs. Il a été montré ([AT06], [HWOM06]) que ce type de conseil aide l'utilisateur en élargissant l'éventail des possibilités qui lui sont offertes. Il peut s'agir de nouvelles fonctionnalités embarquées avec le système conseiller ou de fonctionnalités déjà proposées par le site mais sur d'autres pages.

Ces différents types de conseils ne sont présentés qu'à titre d'exemple. Ils peuvent également être d'une tout autre forme selon les connaissances, les besoins, les idées de l'expert qui construit le modèle d'utilisation et élabore les conseils.

La figure 4.9 présente un exemple de modèle d'utilisation créé pour le site Apache pour les utilisateurs désirant installer le serveur Web HTTPD. Ce graphe n'est pas présenté dans son ensemble car nous n'avons pu proposer le graphe complet tout en conservant une certaine lisibilité. Ce graphe illustre les difficultés de construction de tels graphes : lorsqu'on travaille sur des sites complexes, le nombre de pages et le nombre de liens entre ces pages deviennent rapidement importants, rendant difficile la représentation des modèles, mais également leur gestion.

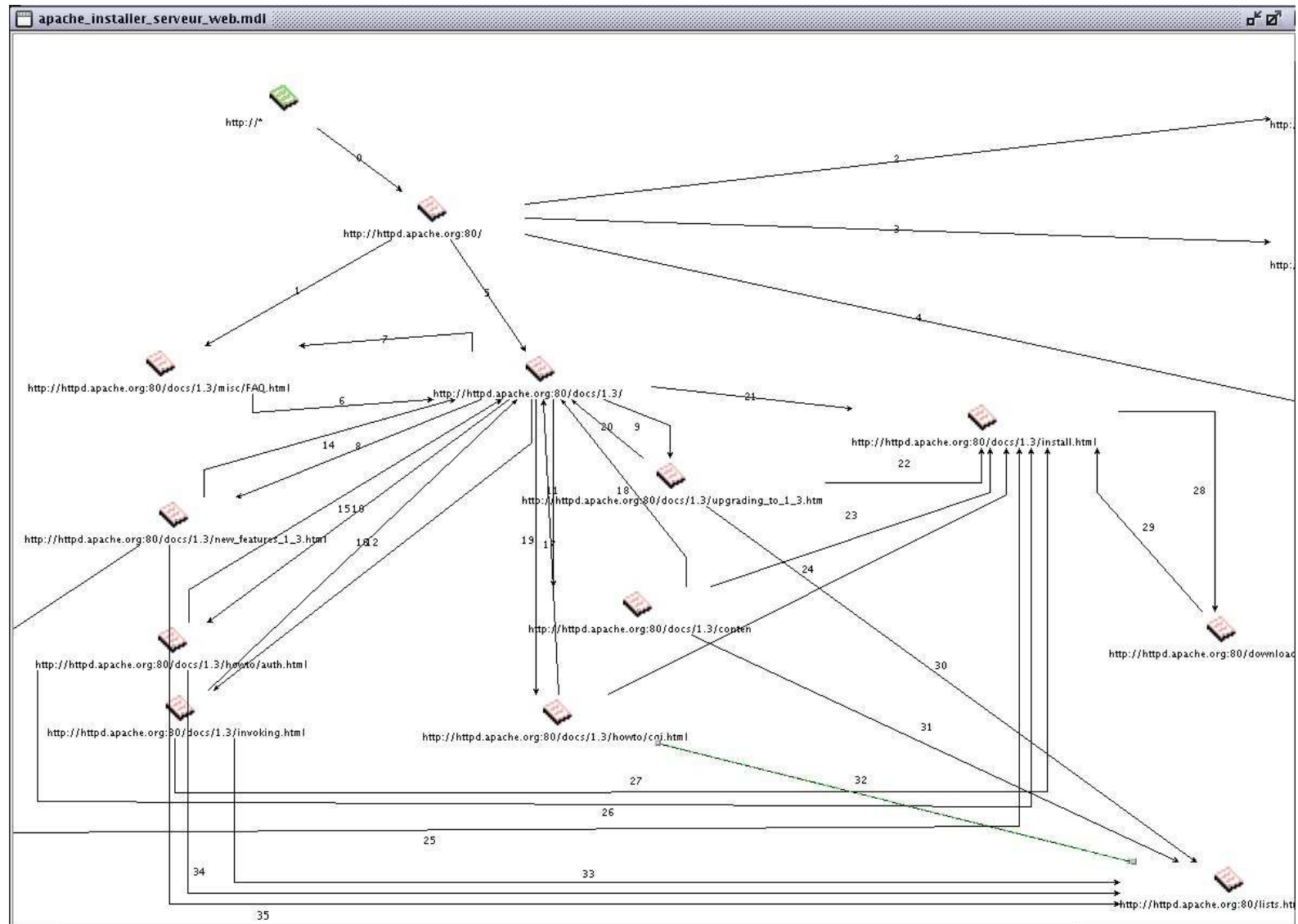


FIG. 4.9 – Exemple de modèle d'utilisation

4.1.4.2 Utilisation des modèles

Choix du modèle utilisé. Au paragraphe 4.1.3.2, plusieurs exemples de tâches qu'un site complexe permet de réaliser ont été présentés. Ces tâches peuvent être représentées sous la forme de modèles d'utilisation proposés aux utilisateurs par le système conseiller. Le fait de proposer plusieurs modèles pose le problème de la sélection du modèle de référence. Deux approches sont possibles :

1. Déterminer automatiquement le modèle le plus pertinent par un diagnostic de la navigation de l'utilisateur.
 - Avantages : l'utilisateur n'a pas à intervenir, à préciser la tâche qu'il compte accomplir sur le site. On sait que nombre d'utilisateurs, d'une part, répugnent à répondre à des questionnaires et, d'autre part, ont du mal à exprimer leurs besoins [Che05].
 - Inconvénients : ce type d'approche est peu précis, l'utilisateur ayant parfois une navigation chaotique [Vas96]. Or, l'impact d'une erreur peut être lourd : une erreur dans l'appréciation du but de l'utilisateur amène le système conseiller à proposer des conseils qui ne correspondent pas aux besoins de l'utilisateur et donc à être abandonné.
2. Laisser l'utilisateur choisir le modèle le plus approprié.
 - Avantages : En présentant clairement les modèles et les tâches qu'ils permettent de réaliser, le risque de choisir un modèle qui ne corresponde pas aux besoins de l'utilisateur est très limité puisque c'est l'utilisateur lui-même qui choisit le modèle. Or, cet avantage est primordial puisqu'il permet (presque) de s'assurer de l'adéquation entre l'objectif de l'utilisateur et le modèle choisi.
 - Inconvénients :
 - L'utilisateur peut aborder le site sans une idée précise de son but. Cette approche l'oblige alors (s'il veut utiliser le système conseiller, ce qui n'est en rien obligatoire) à réfléchir à ses objectifs, ce qui n'est pas forcément aisé, notamment pour les utilisateurs novices. Cet inconvénient peut cependant être atténué, par exemple, en créant un modèle correspondant à une navigation libre dans le site et en proposant une assistance générique de type visualisation de l'historique de la navigation, sauvegarde automatique des pages et offrant de multiples options pour un retour en cas de désorientation ou d'accès à des pages qui ne correspondent plus à l'objectif.

- La tâche de l'utilisateur peut changer au cours de sa navigation dans le site. Nous verrons par la suite qu'on peut pallier cet inconvénient, par exemple, en détectant des comportements qui ne correspondent plus aux parcours proposés par le modèle choisi.
- Interroger l'utilisateur peut être mal perçu. Considérée comme intrusive, l'obligation de répondre à des questions peut rebuter l'utilisateur.

Dans les travaux menés au cours de cette thèse, nous avons choisi cette dernière approche : nous proposerons à l'utilisateur, en début de session, une liste présentant les modèles d'utilisation disponibles et leurs différentes caractéristiques (description de la tâche que le modèle permet de réaliser, utilisateurs ciblés, etc.). Notons que ce choix peut être remis en cause. Il n'a pas d'implication forte sur notre approche générale (c'est-à-dire sur l'architecture générale ou sur les principes de base de l'approche) et d'autres choix pourraient être mis en oeuvre.

Mode d'utilisation des modèles. Le fonctionnement du système conseiller est fondé sur les modèles d'utilisation. Le parcours de l'utilisateur est analysé en fonction du modèle que l'utilisateur a sélectionné : l'utilisateur est sur une page, il choisit d'en visionner une autre et pour cela clique sur un hyperlien ; le système conseiller va traduire cette action en termes d'arcs et de noeuds du modèle, puis va proposer à l'utilisateur les éventuels conseils associés à l'arc correspondant à l'hyperlien choisi par l'utilisateur. En d'autres termes, le système conseiller suit, à travers le modèle sélectionné, la navigation de l'utilisateur dans le site et propose, le cas échéant, des conseils.

4.2 Processus d'élaboration d'un système conseiller

L'élaboration d'un système conseiller fondé sur notre approche peut se décomposer en plusieurs étapes :

0. Création/adaptation/amélioration d'un système conseiller générique

- - - - -

1. Identification et création des modèles de tâches
2. Création des conseils
3. Positionnement des conseils sur les modèles
4. Instanciation d'un système conseiller générique

5. Analyse de l'utilisation et de l'efficacité du système conseiller

Ces étapes sont plus ou moins séquentielles. L'étape 0 (création d'un système conseiller générique) n'est à effectuer qu'une fois. Les étapes 1, 2 et 3 peuvent être effectuées de façon plus ou moins parallèle. Enfin, l'étape 4, comme l'étape 5, impose que les étapes précédentes soient terminées. Plusieurs itérations du processus peuvent être nécessaires.

Nous allons, ci-dessous, préciser chacune de ces étapes.

4.2.1 Création/adaptation/amélioration d'un système conseiller générique

Les deux éléments principaux qui composent notre approche sont, d'une part, les modèles d'utilisation qui permettent la prise en compte de la tâche de l'utilisateur et, d'autre part, un système conseiller générique qui va permettre la proposition des conseils en se fondant sur ces modèles d'utilisation. Il s'agit donc, dans un premier temps, de disposer d'une architecture qui puisse (1) lire des modèles d'utilisation, (2) analyser le comportement des utilisateurs en temps réel et en fonction de ces modèles et (3) leur proposer les conseils inclus dans les modèles. Ce système peut être qualifié de générique puisqu'il n'est pas dédié à un site particulier. Il peut ensuite être utilisé pour créer différents systèmes conseillers particuliers : c'est l'association des modèles (qui sont eux créés pour un site particulier) au système conseiller générique qui permet son instantiation. Un système conseiller générique est donc une architecture informatique présentant les fonctionnalités nécessaires à un système conseiller (architecture réseau, mécanismes de comparaison des actions de l'utilisateur par rapport aux modèles, etc.) mais qui ne comprend pas de modèle de tâches ni de conseil particulier (il est ainsi semblable à une "coquille" et de même nature qu'un "générateur" de système expert).

Lors des itérations du processus, ce système conseiller générique peut être amélioré en fonction des remarques des concepteurs de modèles, des concepteurs de sites cibles ou des utilisateurs du système.

Dans cette thèse, nous proposons une architecture informatique opérationnelle de ce type (appelée Marco pour Model bAsed ReCOmmender system), *i.e.*, un système conseiller générique directement utilisable.

4.2.2 Identification et création des modèles de tâches

Etre à même d'identifier et de créer des modèles de tâches (étape B, détaillée ci-dessous) impose d'appréhender la structure du site (étape A.1) et d'examiner les usages effectifs du site (étape A.2).

Etape A - Analyser le site cible.

1. Analyse de la structure du site. L'analyse de la structure du site passe par la réponse à des questions du type :
 - Comment le site est-il organisé ? (architecture hiérarchique, structuration par domaine, par catégorie, etc.)
 - Quelles sont les pages clés ? Les pages clés d'un site sont des pages charnières, *i.e.*, ce sont des pages sur lesquelles l'utilisateur doit faire des choix. De par leur importance, ces pages permettent à l'utilisateur de s'orienter dans le site.
 - Quels types de pages sont présents dans le site (pages présentant un sommaire, une liste, une information seule, etc.) ?
 - Quelles fonctionnalités sont proposées ?
 - etc.

Ces questions vont permettre de mettre en lumière l'organisation des pages du site, les fonctionnalités proposées, ainsi que de créer le graphe du site.

2. Analyse des usages du site. Il s'agit d'analyser comment le site est utilisé. Cette analyse peut se faire, par exemple, à l'aide d'une analyse des logs du serveur Web ou à l'aide d'interviews d'utilisateurs. Analyser les usages d'un site Web est un problème en soi (voir les travaux en fouille de données, par exemple) que nous n'avons pas abordé en tant que tel. Nous avons cependant été obligé de tenir ce rôle d'analyste pour déterminer différents usages des sites Web étudiés et ainsi créer des modèles d'utilisation nécessaires au fonctionnement de nos prototypes et à l'illustration de notre approche. Ces différents modèles d'utilisation sont présentés à titre d'exemple dans ce mémoire.

Etape B - Identifier les tâches. Créer les modèles de tâches. L'analyse de la structure du site et de ses usages vise à dégager les différentes utilisations du site Web, les difficultés rencontrées par les utilisateurs, les usages inattendus du site ou de certaines fonctionnalités. Il s'agit ensuite d'identifier les tâches qui vont être assistées, puis de créer les modèles de tâches, *i.e.*, de déterminer, pour chaque

modèle de tâches, les noeuds et les liens qui le composent. Plusieurs modes de créations du graphe représentant le modèle de tâches sont envisageables :

- Création manuelle. L'expert crée à la main les différents noeuds et les arcs du graphe. Cependant, pour des sites complexes, créer manuellement les graphes modélisant ces sites peut être long, fastidieux et source d'erreurs.
- Création assistée. L'expert est assisté lors de la création du graphe. On peut imaginer des aides telles que :
 - Retranscrire dans un graphe une navigation de l'expert : l'expert navigue sur le Web et les pages qu'il parcourt sont automatiquement insérées dans le graphe.
 - Proposer la création automatique des arcs : l'expert saisit les noeuds ; le système recherche dans le site cible les liens entre les pages et crée les arcs.
 - Proposer la complétion automatique du graphe : un noeud est inséré dans le graphe ; le système propose à l'expert d'insérer automatiquement les n niveaux suivants.
 - etc.
- Création automatique. Le système crée automatiquement le graphe représentant le site cible en se fondant sur le site lui même, sur les logs du site ou sur d'autres données.

Ces différentes approches de création des modèles de tâches ne sont que des exemples. Nous présenterons dans les chapitres suivants les outils que nous avons élaborés.

4.2.3 Création des conseils

Il s'agit pour l'expert de définir les conseils qu'il désire présenter à l'utilisateur et les conditions de leur présentation. Les conseils peuvent être de différentes natures (cf. paragraphe 4.1.4.1). Selon l'environnement dans lequel travaille l'expert, cette étape peut être plus ou moins assistée : on peut laisser l'expert créer un conseil sous la forme d'une page Web et lui proposer d'associer l'URL de cette page à un arc du graphe. Un environnement plus complet peut aussi être proposé et permettre à l'utilisateur d'utiliser des exemples de conseils qu'il lui suffit ensuite d'adapter au site cible. Nous présenterons dans les chapitres suivants les outils que nous avons élaborés.

4.2.4 Positionnement des conseils sur les modèles

Il s'agit d'associer des conseils aux arcs. Nous avons précisé au paragraphe 4.1.1 qu'un conseil au plus pouvait être attaché à un arc (un arc représente une action, on ne présente qu'un conseil par action, la notion de conseil n'étant pas limitative : un conseil peut être composé de plusieurs textes, liens et fonctionnalités). L'expert, en fonction de l'analyse du site effectuée précédemment (cf. paragraphe 4.2.2), va positionner les conseils. Là encore, selon l'environnement dans lequel travaille l'expert, cette tâche peut être plus ou moins manuelle : par exemple, un graphe peut être représenté sous la forme d'une matrice et manipulé dans un environnement très rudimentaire. A l'inverse, un environnement graphique peut être conçu pour faciliter ces manipulations.

4.2.5 Instanciation du système conseiller générique

Les étapes 1 à 3 ont permis de concevoir des modèles d'utilisation. L'instanciation du système conseiller générique (étape 4) s'effectue par la prise en compte des modèles d'utilisation par le système conseiller. Une fois que sont installés les modèles d'utilisation, le système conseiller est à même de suivre la navigation des utilisateurs en fonction des modèles et de leur proposer des conseils.

4.2.6 Analyse de l'utilisation et de l'efficacité du système conseiller et évolutions

Après quelques temps d'utilisation, il est possible d'analyser les usages du couple système conseiller - site cible. Le site cible répond-il aux attentes des utilisateurs ? Les usages constatés correspondent-ils mieux aux attentes des concepteurs ? Les modèles sont-ils utilisés ? Observe-t-on des sorties intempestives des modèles, *i.e.*, les modèles correspondent-ils effectivement à des usages du site ? etc. Toutes ces questions vont permettre d'améliorer le système par une nouvelle itération du processus : modification du système conseiller générique (si des dysfonctionnements propres au système conseiller sont observés ; lorsque le développement du système conseiller générique est stabilisé, cette étape n'a plus lieu d'être), création de nouveaux modèles de tâches, adaptation/modification des anciens, création de nouveaux conseils, adaptation/modification des anciens, etc.

4.3 Synthèse

Pour que l'aide proposée par le système conseiller prenne en compte le but de l'utilisateur, nous proposons de fonder le fonctionnement global du système conseiller sur la notion de tâche de l'utilisateur. Pour réaliser cet objectif, nous proposons une approche fondée sur plusieurs principes :

1. Une architecture épiphyte. Une telle architecture permet une séparation franche entre le système conseiller et le serveur Web cible. Cette séparation permet de proposer une approche générique et favorise la portabilité, la simplicité et l'évolutivité du système conseiller.
2. Des modèles de tâches. Les modèles de tâches permettent de modéliser différents usages du site Web cible. Leur création est une tâche en soi. Elle nécessite une analyse pointue du site et de ses usages.
3. Des modèles d'utilisation. Ces modèles sont créés en associant des conseils aux modèles de tâches. Comme pour les modèles de tâches, créer des conseils n'est pas trivial, nécessite des compétences particulières et est une tâche à part entière. Une fois créés, les modèles d'utilisation sont utilisés par le système conseiller pour suivre la navigation de l'utilisateur et le guider en lui proposant des conseils.

L'élaboration d'un système conseiller fondé sur ces principes repose sur plusieurs étapes qu'il est possible de réitérer :

0. Création/adaptation/modification d'un système conseiller générique (si une telle architecture n'est pas disponible). Cette étape n'est réitérée qu'en cas de problème propre au fonctionnement du système conseiller.

1. Identification et création des modèles de tâches.
2. Création des conseils.
3. Positionnement des conseils sur les modèles de tâches et création des modèles d'utilisation.
4. Instanciation du système conseiller générique : installation des modèles d'utilisation, adaptation réseau.
5. Analyse du fonctionnement du couple système conseiller - site cible : analyse des usages courants et si besoin nouvelle itération à partir de l'étape 0 ou 1.

Nous présentons dans les chapitres suivants l'architecture et les outils informatiques que nous avons développés pour tester la faisabilité de cette approche en proposant une opérationnalisation particulière.

Chapitre 5

Présentation détaillée de l'architecture proposée : le système conseiller générique Marco

Nous avons proposé, chapitre 4, une approche permettant l'élaboration d'un système conseiller épiphyte. Nous présentons, dans ce chapitre, une architecture et des outils permettant une mise en oeuvre de cette approche.

5.1 Introduction

Pour proposer une architecture capable de supporter un système conseiller épiphyte, plusieurs perspectives sont envisageables [SCDT00] :

- Des approches développées côté serveur.

Principe. Il s'agit de proposer des systèmes qui vont fonctionner sur la même machine que le serveur Web cible (ou, au moins, sur une machine physiquement proche). Ces approches reposent, par exemple, sur :

- Une analyse des informations qui circulent sur le réseau.
- Une analyse des logs des serveurs Web.

Intérêt. Ces techniques sont intéressantes puisqu'elles permettent de disposer de beaucoup d'informations, même si, dans les logs d'un serveur Web quelques informations sont masquées : par exemple, les paramètres de la

commande POST ou l'accès à des pages stockées dans les caches et délivrées à l'utilisateur sans intervention du serveur Web.

Inconvénients par rapport à notre problématique. Notre approche spécifie un système conseiller épiphyte, c'est-à-dire un système conseiller qui puisse être greffé sur un site Web sans qu'il soit nécessaire de le modifier. L'objectif est de séparer effectivement le système conseiller du serveur Web. Or, utiliser les logs du serveur Web ou analyser le réseau pour fournir, en temps réel, des conseils impose une proximité géographique des systèmes ce qui va à l'encontre de l'objectif de séparation du système conseiller et du site Web cible (on peut observer que l'émergence des réseaux à très haut débit peut permettre d'envisager une analyse à distance et en temps réel des logs d'un serveur Web. Cependant, proposer une telle architecture nous paraît aléatoire et risqué, notre objectif étant de proposer une architecture qui permette l'élaboration d'un système conseiller réactif et fiable).

- Des approches développées côté client.

Principe. Il s'agit de proposer des systèmes qui vont fonctionner sur l'ordinateur de l'utilisateur. Ces approches peuvent reposer sur :

- Des applets Java : les applets Java sont des programmes qui vont être chargés en même temps qu'une page Web et qui vont s'exécuter sur l'ordinateur de l'utilisateur.
- Des navigateurs ad hoc : il est possible de développer des navigateurs Web spécifiques capables de proposer des pages Web et les conseils d'un système conseiller.

Intérêt. Ces techniques permettent d'analyser le comportement de l'utilisateur et de proposer des conseils de façon efficace : un programme va fonctionner chez le client, observer ses actions (les observations peuvent être très fines : utilisation des ascenseurs, mouvements de souris, etc.) et communiquer avec le système conseiller.

Inconvénients par rapport à notre problématique. Utiliser de tels outils va imposer une participation de l'utilisateur et des changements dans ses habitudes : dans le meilleur des cas, il doit accepter l'exécution d'une applet, dans le pire, il doit changer de navigateur et opter pour un navigateur ad hoc développé pour fonctionner avec le système conseiller.

De plus, il existe une grande hétérogénéité des ordinateurs et des systèmes d'exploitations. Faire fonctionner un programme coté client impose de développer de nombreuses versions pour pouvoir s'adapter aux systèmes des

utilisateurs, et donc de prendre le risque de voir un utilisateur être incapable de faire fonctionner le logiciel sur son poste.

Enfin, travailler côté client complique l'évolutivité des développements du système conseiller : chaque nouvelle version du système conseiller doit être mise à jour sur chaque poste client (si cette mise à jour peut être automatisée, des dysfonctionnements sont toujours à craindre).

- Des approches fondées sur l'utilisation de proxies.

Principe. Il s'agit de proposer des systèmes dont l'architecture va être de type proxy, *.i.e.*, des systèmes qui sont placés entre le serveur Web et le client et qui offrent donc une solution intermédiaire par rapport aux approches présentées ci-dessus.

Intérêt. Ces systèmes sont simples à utiliser : si le système est un proxy véritable, l'utilisateur devra simplement modifier la configuration de son navigateur : il définira le proxy du système conseiller comme étant le proxy qu'il veut utiliser. Dès lors toutes ses communications passeront par ce proxy.

Si c'est l'architecture du système conseiller qui ressemble à celle d'un proxy, l'utilisateur devra uniquement fournir l'URL du système conseiller à la place de celle du serveur Web cible. Pour l'utilisateur, il n'y a donc rien à installer et peu de modifications de son environnement à effectuer.

Inconvénients par rapport à notre problématique. Les informations captées par ces systèmes sont plus limitées que celles captées par un système installé côté client : par exemple, des informations telles que l'utilisation des ascenseurs, les mouvements de la souris, les passages d'un onglet à un autre (en fait, les informations relatives à l'interface utilisateur) ne sont pas recueillies par ces systèmes.

Par rapport à notre problématique, l'approche par proxy présente plusieurs avantages :

1. Elle permet de dissocier effectivement le site Web du système conseiller, propriété que nous recherchons puisque notre problématique vise à séparer les données du site cible de celles du système conseiller.
2. Elle permet d'envisager de proposer des conseils à des utilisateurs sans que ces derniers aient de nombreuses manipulations à effectuer (il n'y a notamment pas d'installation à effectuer) et sans qu'il soit nécessaire d'utiliser un système particulier ou une version de système particulière.

C'est pour ces raisons que nous avons choisi une architecture de type proxy pour élaborer le système conseiller générique Marco.

La figure 5.1 présente l'architecture générale du système conseiller générique Marco que nous proposons. Ce schéma expose les différents modules qui composent le système conseiller, leurs relations et les voies de communications qui existent entre le système conseiller, les experts, les utilisateurs et le site cible. Ces différents éléments sont présentés ci-dessous.

5.2 Présentation générale

Nous proposons dans ce paragraphe une présentation générale du système conseiller générique Marco et de ses différents modules. Les détails de cette architecture sont présentés au paragraphe suivant.

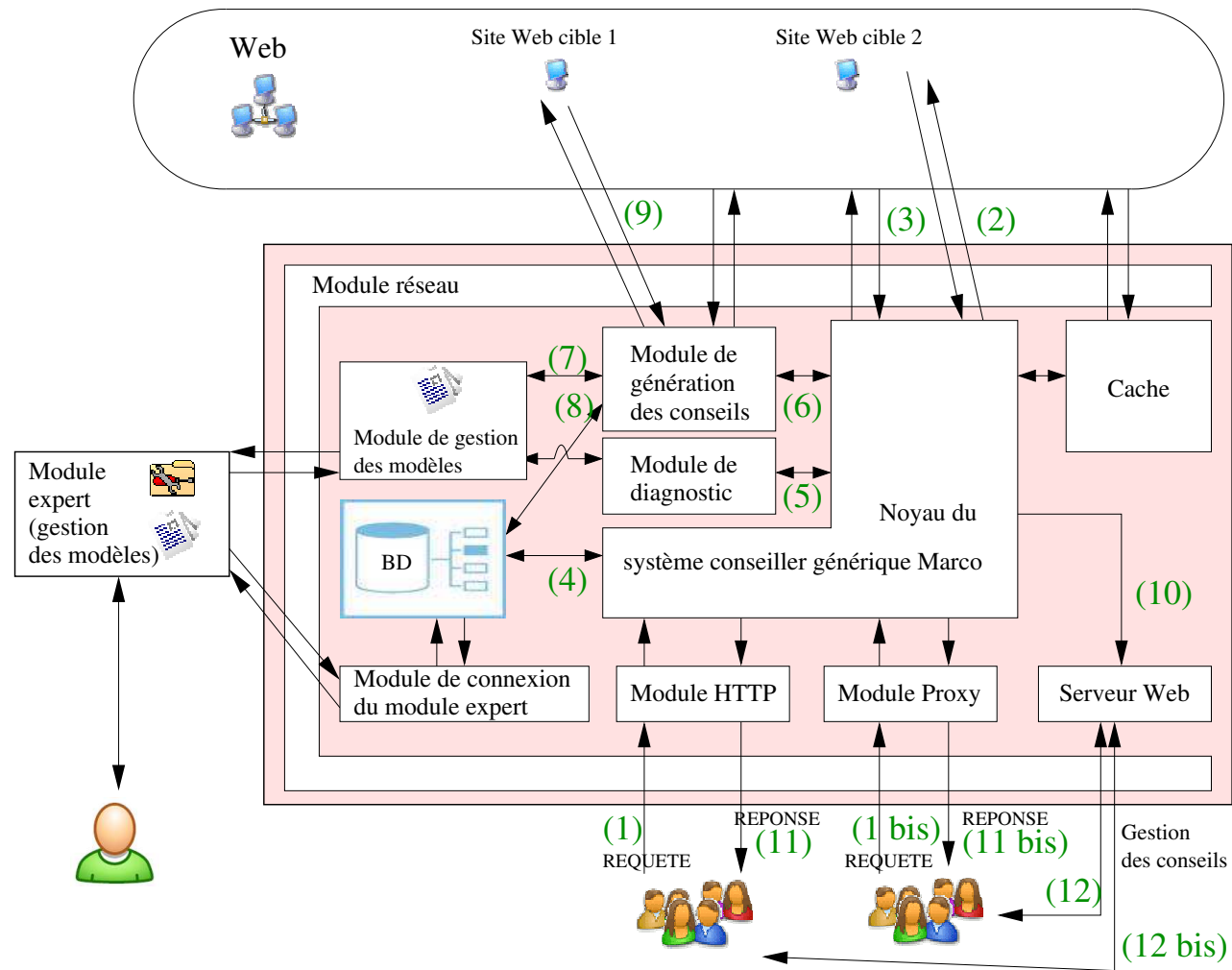


FIG. 5.1 – Schéma général de fonctionnement du système conseiller générique Marco

5.2.1 Modules et notions

Pour fonctionner, le noyau sollicite différents modules :

- Le module de diagnostic :
 - Fonction : ce module analyse la navigation de l'utilisateur en fonction du modèle d'utilisation que l'utilisateur a choisi.
 - Principe de fonctionnement : ce module, en fonction du modèle choisi par l'utilisateur et de la requête HTTP que l'utilisateur vient d'effectuer (*i.e.*, du lien que l'utilisateur vient d'activer), indique au noyau s'il existe un arc du modèle correspondant à la requête.
- Le module de génération de conseils :
 - Fonction : nous avons choisi (cf. paragraphe 4.1.4.1) de proposer les conseils dans une fenêtre adjacente à celle proposée par le site Web. Pour pouvoir être interprété par le navigateur Web de l'utilisateur et lui être présenté, le conseil doit être envoyé au format HTML par le système conseiller. Ce module a pour fonction de construire la page HTML correspondant à un conseil.
 - Principe de fonctionnement : lorsque l'utilisateur clique sur un lien, qu'un arc du modèle correspond à ce lien et qu'un conseil a été associé à cet arc, ce conseil est transformé en une page HTML qui est renvoyée au noyau.
- Les modules de communication :
 - Le module réseau :
 - Fonction : ce module est en charge des communications réseaux du système conseiller.
 - Principe de fonctionnement : ce module est en attente de requêtes HTTP de l'utilisateur, qu'il transmet aux modules assurant une gestion d'un niveau plus élevé des requêtes (module HTTP ou module proxy, par exemple)
 - Le module HTTP :
 - Fonction : ce module permet aux utilisateurs de se connecter au système conseiller pour bénéficier de conseils.
 - Principe de fonctionnement : pour initialiser le processus, l'utilisateur saisit dans son navigateur, non pas l'URL du site cible, mais celle du système conseiller. Le système conseiller renvoie à l'utilisateur une page du site cible et fait donc office d'intermédiaire entre le site cible et l'utilisateur. Pour continuer de te-

nir cette position intermédiaire, le système conseiller, avant d'envoyer une page à l'utilisateur, modifie chaque URL qu'il fait pointer vers lui. Ces modifications à la volée permettent au système conseiller d'intercepter chacune des requêtes de l'utilisateur. Elles sont totalement transparentes pour l'utilisateur.

- Le module Proxy :
 - Fonction : ce module permet aux utilisateurs de se connecter au système conseiller pour bénéficier de conseils.
 - Principe de fonctionnement : l'utilisation du module HTTP, décrite ci-dessus, impose à l'utilisateur de se connecter au système conseiller en saisissant, dans son navigateur, l'URL du système conseiller. L'utilisation du module Proxy évite ceci : l'utilisateur modifie les paramètres de son navigateur et définit le système conseiller comme étant son proxy. Dès lors, chacune de ses requêtes HTTP sera réceptionnée par le système conseiller via le module proxy.
- Le serveur Web :
 - Fonction : comme nous l'avons expliqué précédemment, les conseils sont transformés au format HTML pour pouvoir être interprétés par le navigateur des utilisateurs. Ces pages HTML sont enregistrées dans les structures de données du serveur Web. Ainsi, dans le but d'équilibrer la charge de travail entre les différents modules, c'est le serveur Web, et non le noyau du système conseiller, qui répond aux requêtes des navigateurs des utilisateurs.
 - Principe de fonctionnement : lorsque le système conseiller crée des conseils, ces derniers sont enregistrés dans un emplacement accessible par le serveur Web. C'est ce serveur Web qui est utilisé (c'est-à-dire qui répond aux requêtes des navigateurs) lorsque le conseil est affiché.
- Le module d'identification des experts :
 - Fonction : ce module permet la connexion des experts sur les outils mis à leur disposition pour leur permettre d'élaborer des modèles d'utilisation.
 - Principe de fonctionnement : lorsqu'il démarre un outil, l'expert doit s'identifier. Ce module reçoit l'identifiant et le mot de passe de l'expert et autorise son accès à l'outil.

Nous explicitons ci-dessous plusieurs notions qui vont nous être utiles pour expliquer en détail, dans la suite de ce chapitre, le fonctionnement des différents

modules.

- Identification des utilisateurs : le système conseiller analyse le parcours de l'utilisateur en fonction des modèles d'utilisation et propose des conseils. Il s'agit de déterminer précisément qui est qui. Quand une requête arrive au système conseiller, ce dernier doit être capable de déterminer quel est cet utilisateur : est-ce un nouvel utilisateur ? est-ce un utilisateur ayant déjà choisi un modèle et parcourant le site cible ? etc.

Cependant, l'utilisateur doit pouvoir accéder avec facilité au site cible que celui-ci soit ou non couplé avec le système conseiller. Il n'est donc pas envisageable de demander à l'utilisateur de s'authentifier lorsqu'il veut accéder au site cible et bénéficier des conseils générés par le système conseiller.

Pour résoudre ce problème, un numéro de session (sessionid) est attribué à chaque utilisateur. Il permet au système conseiller, notamment lorsque le module HTTP est utilisé, d'identifier précisément l'utilisateur. En effet, l'utilisation du module HTTP impose la réécriture des URLs des pages qui transitent via le système conseiller (cf. ci-dessus "Le module HTTP"). Au cours de cette réécriture il est facile d'ajouter, à chaque URL, un paramètre spécifiant le numéro de session de l'utilisateur.

Lorsque le numéro de session n'est pas utilisable (par exemple, lorsque le module proxy est utilisé), la seule solution envisageable est de se fonder sur le numéro IP (Internet Protocol) pour identifier l'utilisateur. A chaque ordinateur connecté à internet est attribué un numéro IP. Ce numéro peut être public ; dans ce cas, c'est ce numéro qui identifie l'ordinateur. Il peut aussi être privé ; dans ce cas, des mécanismes de type routeur NAT (Network Address Translation) sont utilisés pour transformer le numéro IP privé en numéro IP public. Ces mécanismes permettent le partage d'un même numéro IP entre différents ordinateurs. Identifier l'utilisateur en fonction de son numéro IP entraîne donc une approximation : plusieurs utilisateurs d'un même réseau privé seront vus par le système conseiller comme un seul.

- URLs absolues. Les URLs absolues précisent à la fois le nom du site et la page à télécharger. Une URL absolue est, par exemple :
"http://httpd.apache.org/download.html".
- URLs relatives. A la différence des URLs absolues, les URLs relatives ne précisent pas le nom du site. Une URL relative est, par exemple : "download.html". L'adresse est alors relative à la page sur laquelle se trouve le lien contenant l'URL.

Exemple : l'utilisateur visualise une page dont l'URL absolue est "http://test.org/docs/document1.html". Dans cette page, on trouve un lien dont l'URL, relative, est "document2.html". Si l'utilisateur clique sur ce

lien, le navigateur de l'utilisateur va construire une URL absolue à partir de l'URL relative :

- (1) le chemin absolu est "http://test.org/docs/",
- (2) l'URL relative demandée est "document2.html" et
- (3) l'URL absolue correspondante est "http://test.org/docs/document2.html".

5.2.2 Données manipulées

Différents types de données sont manipulés par le système conseiller :

- Données nécessaires au fonctionnement du système conseiller. Dans le cadre du fonctionnement du système conseiller, certaines données sont initialisées à chaque session et enregistrées, le temps de la session, dans la base de données du système conseiller :
 - Un numéro de session : il permet d'identifier l'utilisateur utilisant le système conseiller via le module HTTP
 - Le numéro IP : il permet d'identifier l'utilisateur utilisant le système conseiller via le module Proxy
 - Le modèle choisi : le système conseiller analyse le parcours de l'utilisateur en fonction du modèle que l'utilisateur a choisi. Enregistrer ce choix est nécessaire pour que le système conseiller sache à quel modèle se référer lorsqu'une nouvelle requête arrive.
- Données relatives aux modèles. Le système conseiller est fondé sur l'usage de modèles d'utilisation. Les modèles créés et à disposition des utilisateurs sont stockés dans une base de modèles.
- Données nécessaires au fonctionnement de certaines fonctionnalités. Des informations sont enregistrées dans la base de données du système conseiller et sont, soit à disposition de fonctionnalités qui peuvent être proposées aux utilisateurs, soit intégrées à la base de données dans le cadre du développement d'une fonctionnalité particulière. A titre d'exemple, on peut citer :
 - L'historique de navigation de l'utilisateur : cet historique concerne les pages du site cible et les noeuds du modèle choisi. Ces informations sont à disposition de fonctionnalités dont le développement nécessiterait ce type d'information.
 - Les mots clés utilisés lors des recherches dans le site cible : lorsque l'utilisateur fait une recherche dans le site cible, le formulaire utilisé

impose, en général, la saisie de mots clés décrivant l'objet de la recherche. Ce sont ces mots clés qui sont mémorisés. Ces informations sont à disposition de fonctionnalités dont le développement nécessiterait ce type d'information (par exemple, aider l'utilisateur dans le choix de ses mots clés).

- Les informations enregistrées peuvent être plus générales. Par exemple, une fonctionnalité que nous avons développée propose à l'utilisateur de visualiser son parcours dans le site. Cette visualisation est effectuée en présentant une liste de liens et de vignettes, miniatures des pages parcourues. Ces images sont enregistrées sur le site Web du système conseiller et la base de données fait le lien entre l'URL de la page et l'adresse de la miniature. C'est ce type de fonctionnalité supplémentaire qu'il peut être intéressant de développer au fur et à mesure de l'utilisation du système conseiller générique. Ces fonctionnalités sont ensuite proposées à l'expert lors de l'instanciation du système conseiller générique.

5.2.3 Exemple de fonctionnement du système conseiller

L'architecture que nous présentons dans ce chapitre ne prend sens que dans le contexte d'une instanciation particulière. Afin de mieux expliquer le fonctionnement de ce système conseiller générique, nous présentons ici le fonctionnement d'un système conseiller particulier construit à partir de cette architecture.

Le système conseiller, quel que soit le mode de connexion utilisé (via le module Proxy ou HTTP) fonctionne comme un proxy : il va intercepter toutes les communications entre l'utilisateur et le Web. Les requêtes à destination du site cible sont analysées par le système conseiller puis envoyées au serveur désigné par la requête. De même, la réponse du site cible, avant d'être transmise à l'utilisateur, est interceptée par le système conseiller et analysée. En fonction de cette analyse, un conseil est éventuellement produit et le résultat de la requête (*i.e.*, la page demandée par le client) plus le conseil (s'il y a) sont renvoyés au client.

Ainsi, une séquence d'échanges de communication entre l'utilisateur, le système conseiller et le ou les serveurs Web cibles s'établit de la sorte :

1. L'utilisateur saisit, dans son navigateur, l'URL du système conseiller (on prend ici le cas où l'utilisateur a modifié la configuration de son navigateur et défini le système conseiller comme proxy).
2. Les différents modèles d'utilisation disponibles lui sont présentés.
3. L'utilisateur choisit le modèle qui lui paraît le plus approprié. Ce faisant,

il est dirigé vers la page d'entrée du modèle (dans le modèle, un noeud est défini comme le noeud d'entrée ; c'est vers la première URL rattachée à ce noeud qu'est dirigé l'utilisateur).

4. Le système conseiller reçoit une requête en provenance du navigateur de l'utilisateur (cf. (1) et (1 bis) de la figure 5.1). La requête va être analysée en fonction de son type :

- La requête pointe sur un fichier binaire (par exemple, une image ou un document ayant un format particulier).
- La requête pointe sur une page HTML non présente dans le modèle.
- La requête pointe sur une page HTML faisant partie de celles impliquées dans le modèle d'utilisation choisi par l'utilisateur.

Elle est ensuite transmise :

- Au serveur cible (2) pour les requêtes qui concernent une page du modèle.
- Au serveur Web concerné pour les requêtes qui concernent des pages qui n'appartiennent pas au modèle courant (rappelons que nous considérons, dans cet exemple, que l'utilisateur a modifié son navigateur et configuré le système conseiller comme étant son proxy ; dès lors, toutes ses requêtes transitent par le système conseiller).

5. Le système conseiller reçoit une réponse à sa requête (3) :

- Si la requête concernait un *fichier binaire*, la réponse est directement renvoyée vers le client via le module de connexion concerné (11) ou (11 bis) (dans cet exemple, il s'agit du module proxy donc (11 bis)).
- Si la requête concernait une *page HTML qui ne correspond pas à un noeud du modèle d'utilisation*, la réponse est directement renvoyée vers le client via le module de connexion concerné (11 bis).
- Si la requête concernait une *page HTML qui correspond à un noeud du modèle d'utilisation*, plusieurs modules du système vont être successivement activés :
 - (a) Le noyau : il met à jour l'historique de l'utilisateur (4).
 - (b) Le noyau : il transmet la requête au module de diagnostic (5).
 - (c) Le module de diagnostic : il détermine quelle est l'action courante de l'utilisateur dans le modèle, *i.e.*, sur quel arc du modèle d'utilisation se trouve l'utilisateur.
 - (d) Le noyau : il transmet l'action courante au module de génération de conseils (6).
 - (e) Le module de génération de conseils : il va générer le conseil. Il s'appuie pour cela sur :
 - La page reçue du site Web cible qui peut contenir des infor-

mations utiles à la création d'un conseil (par exemple, proposer à l'utilisateur de sélectionner plusieurs liens jugés intéressants pour y revenir ultérieurement, impose d'analyser la page).

- Le modèle courant (7) qui contient les informations concernant le conseil à générer.
 - La base de données (8) qui contient des informations sur l'utilisateur.
 - Le Web (9) qui peut être utilisé pour récupérer des informations complémentaires, utiles à la construction de certains conseils (le système conseiller génère donc automatiquement des requêtes supplémentaires).
- (f) Le noyau : il va enregistrer le conseil dans les structures de données du serveur Web du système conseiller (10).
- (g) Le noyau : il va modifier très légèrement la page reçue du site Web cible : une fonction javascript va être ajoutée dans la page pour permettre l'ouverture de la fenêtre de présentation du conseil en même temps que la page (notons que lorsque le module Proxy est utilisé, cette modification, invisible à l'utilisateur, est la seule qui est effectuée sur la page reçue du site cible).
- (h) Les modules de connexion (module HTTP ou module Proxy) : ils vont transmettre la page reçue légèrement modifiée au navigateur de l'utilisateur (11) et (11 bis). Celui-ci, en ouvrant cette page, ira automatiquement récupérer le conseil sur le site Web du système conseiller (12) et (12 bis).

5.3 Détails de l'architecture technique

Cette section détaille les dimensions techniques des différents modules de l'architecture.

5.3.1 Noyau du système de conseiller

Le noyau du système conseiller est l'élément central du système. C'est à partir de ce module que va être organisé le fonctionnement du système conseiller. Les autres modules ont chacun un rôle précis et le noyau va les utiliser pour réaliser l'objectif qui est le sien : d'une part, faire transiter les requêtes HTTP et les

pages HTML entre les utilisateurs et les serveurs Web cibles (utilisation du module Cache et du module Réseau) et d'autre part, analyser la navigation de chaque utilisateur en fonction du modèle choisi et proposer, si nécessaire, des conseils (utilisation du module Diagnostic, du module de Gestion des modèles et du module de Création de conseils). L'algorithme 1 présente ce fonctionnement.

5.3.2 Module réseau

La couche réseau du système conseiller est conçue comme celle d'un proxy : le système est l'intermédiaire entre l'utilisateur (plus précisément son navigateur) et le site cible. Son fonctionnement peut se résumer ainsi :

1. L'utilisateur clique sur un lien d'une page Web. Son navigateur Web envoie une requête au serveur Web concerné.
2. Le module réseau du système conseiller, interceptant la communication, reçoit la requête. Il fait suivre la requête au noyau qui va l'analyser. Si cette requête doit être relayée vers un serveur Web cible (*i.e.*, la réponse à cette requête n'est pas dans le cache du système conseiller), le module réseau est sollicité par le noyau pour envoyer la requête au serveur Web cible.
3. Le serveur Web cible reçoit la requête et en réponse, envoie une page HTML.
4. Le module réseau du système conseiller réceptionne la page HTML qu'il fait suivre au noyau. Le noyau (1) active différents modules qui vont élaborer un conseil et (2) modifie la page HTML à renvoyer à l'utilisateur pour que le navigateur Web de ce dernier télécharge automatiquement le conseil lors de l'ouverture de la page HTML.
5. Le module réseau fait suivre la page HTML.
6. Le navigateur Web de l'utilisateur reçoit la page HTML, la présente à l'utilisateur et envoie au serveur Web du système conseiller une requête de téléchargement du conseil.
7. Le module réseau du système conseiller reçoit la requête et la transmet au serveur Web du système conseiller.
8. Le serveur Web du système conseiller reçoit la requête et en réponse, envoie une page HTML correspondant au conseil.
9. Le module réseau fait suivre la page HTML.
10. Le navigateur Web de l'utilisateur reçoit la page HTML correspondant au conseil et la présente à l'utilisateur.

Comme nous l'avons déjà dit dans l'introduction, nous avons choisi le langage java pour développer notre architecture. Ce langage est particulièrement adapté

Algorithme 1 Algorithme du noyau du système conseiller

tantque vrai faire

si nouvelle requête **alors**

// Gestion des flux entrants (requêtes HTTP)

// et des flux sortants (pages HTML)

// Utilisés : module Cache et module Réseau

si requete \in cache **alors**

 page_courante \leftarrow recuperer_page_dans_cache(requete)

sinon

 page_courante \leftarrow recuperer_page_dans_site_cible(requete)

 enregistrer_page_dans_cache (page_courante)

finsi

// Gestion des conseils

// Utilisés : module Diagnostic, module de Gestion des modèles et

// module de Création de conseils

si format_HTML(page_courante) **alors**

 type_conseil \leftarrow analyser_navigation_utilisateur(utilisateur,requete)

si type_conseil **alors**

 conseil \leftarrow generer_conseil(type_conseil,page_courante)

 URL_conseil \leftarrow enregistrer_conseil_sur_site_Web_SC(conseil)

 page_courante \leftarrow modifier_page_courante_pour_ouverture_conseil
 (URL_conseil, page_courante)

finsi

finsi

 envoyer_page_a_l_utilisateur(page_courante)

finsi

fin tantque

pour le développement d'applications Web. Ainsi, coder une architecture en proxy est relativement simple, comme l'illustre le code présenté listing 5.1 ¹.

Listing 5.1: architecture en proxy codée en java

```
public static void main(String[] args)
{
    /*
    ...
    partie de code supprime dans un soucis de clarte.
    Les codes complets sont fournis en annexe.
    ...
    */
    while (true)
    {
        try
        {
            ss = new ServerSocket(MARCO_PORT);
            System.out.println("-----");
            System.out.println(MARCO_HOST_SIMPLE
                               + " a l'affut de connexions sur le port "
                               + ss.getLocalPort());
            System.out.println("-----");

            // attente de connexions
            while (true)
            {
                Connexion j = new Connexion(ss.accept());
                j.start();
            }
        }
        catch (IOException e)
        {
            System.err.println(
                "MARCO - socket principal HS - Port "
                + MARCO_PORT + " - erreur : " + e);
            System.exit(0);
        }
    }
}

// le systeme est a l'affut de connexions
// chaque requete recue est geree par la methode gererRequete
```

1. Rappelons que notre système conseiller s'appelle Marco. Dans les codes java présentés les variables contenant le nom Marco font donc référence au système conseiller (par exemple, le `BufferedReader` (gestionnaire de flux) `Client_to_marco` fait référence à une connexion et un flux entre le navigateur du client et le système conseiller Marco).

```

private void gererRequete(RequeteWebTipV3 requete ,
                          PrintStream marco_to_client ,
                          BufferedReader Client_to_marco)
{
    /*
    ...
    simplifications . Les codes complets sont fournis en annexe .
    ...
    */

    // lecture de la fin de la requete envoyee par le client
    entete = new Entete_client(requete.getMode() + " "
                               + requete.getUrld() + " "
                               + requete.getFin_requete() + "\n");
    entete.lire(Client_to_marco , client_to_marco_socket);

    try
    {
        // initialisation de la connexion entre Marco et
        // le serveur Web cible
        marco_to_cible_socket =
            new Socket(requete.getTargetwebsite().getNom() ,
                      requete.getTargetwebsite().getPort());
        BufferedInputStream Cible_to_marco =
            new BufferedInputStream(marco_to_cible_socket.getInputStream());
        PrintStream Marco_to_cible =
            new PrintStream(marco_to_cible_socket.getOutputStream());

        // on envoie l'entete au serveur
        envoyerEnteteAuServeur(requete , marco_to_cible , entete);

        // si la requete est une requete texte
        if (requete.isTexte())
        {
            // La methode ecouteCibleEtPrepareConseil fait
            // les actions suivantes :
            // - recuperation de la reponse du serveur
            // - interaction avec le module de diagnostic du systeme
            //   conseiller pour determiner le conseil a proposer
            // - insertion (si necessaire) du conseil
            // - envoi vers le client
            entete_serveur =
                Ecoute_cibleV2.ecouteCibleEtPrepareConseil(
                    marco_to_client , Cible_to_marco ,
                    marco_to_cible_socket , requete);
        }
        else // la requete n'est pas une requete texte
            // (ce peut etre , par exemple , une demande d'image)
    }
}

```

```

    {
        // La methode ecouteCibleEtFaitSuivre fait
        // les actions suivantes :
        // - recuperation de la reponse du serveur
        // - envoi vers le client
        entete_serveur = Ecoute_cibleV2.ecouteCibleEtFaitSuivre(
            Marco_to_client , Cible_to_marco ,
            marco_to_cible_socket , requete);
    }

    // des que le serveur a fini d'envoyer ces donnees
    // on clot la connexion
    client_to_marco_socket.close();
    System.out.println("socket vers le client ferme");

    marco_to_cible_socket.close();
    System.out.println("socket vers le serveur cible ferme");

}
catch (IOException e)
{
    System.err.println("connexion : pb IOException : "
        + e.getMessage());
}
}

```

Deux modules viennent compléter ce module et permettent la communication réseau entre les utilisateurs et le système conseiller. Il s'agit des modules Proxy et HTTP.

5.3.3 Module HTTP

Principe de fonctionnement du module HTTP. Le module HTTP est une des deux approches offertes par notre architecture pour que les utilisateurs et le système conseiller puissent communiquer. Le principe de fonctionnement de ce module est le suivant : le système conseiller modifie, à la volée, les URLs des pages demandées par l'utilisateur. Les modifications effectuées portent principalement sur l'adresse du site cible de la requête : le système conseiller la modifie pour être systématiquement le destinataire des requêtes HTTP de l'utilisateur.

Ainsi, si on reprend l'exemple du site Apache, une URL dans une page du site HTTPD peut être :

"http://httpd.apache.org/download.html".

Le système conseiller analyse la page et modifie, à la volée, les URLs. L'URL ci-dessus est modifiée et devient :

"http://marco.univ-lemans.fr:8001/download.html
?sessionId=000075412&tw=http://httpd.apache.org".

Les modifications sont :

1. Un changement du site cible : la requête HTTP ne sera pas destinée au site httpd.apache.org mais au système conseiller (marco.univ-lemans.fr:8001).
2. Un ajout de paramètres :
 - sessionId : le numéro de session identifiant l'utilisateur que nous avons évoqué ci-dessus.
 - tw : ce paramètre précise le site vers lequel le système conseiller doit faire suivre la requête (tw = Target Web Site).

Illustration du fonctionnement du module HTTP. Sur le site du système conseiller, l'utilisateur choisit un modèle qui correspond à ses besoins (par exemple, "installer le serveur Web HTTPD de la fondation Apache"). Les actions suivantes vont alors s'enchaîner :

1. Pour sélectionner un modèle, l'utilisateur clique sur le lien qui correspond au modèle choisi. En cliquant sur ce lien, il exécute une requête HTTP, par exemple : "get /download.html?sessionId=000075412&tw=http://httpd.apache.org" (cette requête correspond à l'URL présentée ci-dessus).
2. Le système conseiller reçoit la requête et l'analyse :
 - Elle provient de l'utilisateur "000075412" (information donnée par le paramètre sessionId).
 - Elle est à destination du site "httpd.apache.org" (information donnée par le paramètre tw).
 - La page désirée est "/download.html".
3. Le système conseiller envoie la requête au serveur Web httpd.apache.org.
4. Le système conseiller reçoit la réponse. A ce stade, l'objectif du système conseiller est d'éviter que des liens mettent en contact direct le navigateur de l'utilisateur et le site cible. Nous avons montré plus haut comment les

URLs étaient modifiées. Notons cependant que :

- (a) Ces transformations, qui s'effectuent à la volée, sont plus légères pour les URLs relatives que pour les URLs absolues : les URLs relatives posent moins de problèmes puisque, sans indication particulière, le navigateur les renvoie vers le site à l'origine de la page (et donc, dans notre cas, vers le système conseiller). Ainsi, pour les URLs relatives, seuls des paramètres (par exemple, sessionid et tws évoqués plus haut) sont ajoutés.
- (b) Dans un souci de rapidité, seuls les liens vers des pages HTML (ou assimilées) vont être modifiés : les URLs absolues utilisées, par exemple, pour des images sont laissées telles quelles et le navigateur fait la requête directement auprès du site concerné ce qui, pour ce type de fichiers, ne pose pas de problème.

La figure 5.2 présente un exemple de modification d'une page HTML.

Soulignons que l'utilisateur n'intervient à aucun moment dans ces traitements et que ces derniers sont complètement transparents pour lui. Sa seule tâche est d'initialiser correctement le processus : l'accès aux pages désirées ne se fait pas via le site cible (*i.e.*, dans notre exemple, il ne saisit pas dans son navigateur l'URL "http://httpd.apache.org") mais via le système conseiller (*i.e.*, dans notre exemple, il saisit l'URL du système conseiller "http://marco.univ-lemans.fr:8001").



FIG. 5.2 – Exemples de modifications à la volée effectuées par le système conseiller

Identification des utilisateurs et module HTTP. Les modifications à la volée, décrites ci-dessus, permettent d'insérer, dans toutes les URLs des pages que parcourt l'utilisateur, un numéro de session le caractérisant. Ce numéro identifie de façon unique l'utilisateur et permet ainsi de suivre sa progression dans le modèle.

Avantages du module HTTP. Transformer à la volée les URLs présente l'avantage d'offrir la possibilité d'ajouter un paramètre aux URLs : en insérant un numéro de session, le système conseiller identifie de façon précise l'utilisateur. Une autre solution aurait pu être d'utiliser l'adresse IP de l'utilisateur. Cependant, une telle approche souffre d'un manque de précision lorsque l'utilisateur est, par exemple, derrière un routeur NAT (Network Address Translation) et qu'il partage ainsi son IP avec d'autres.

Inconvénients du module HTTP. Les transformations sont délicates : les pages HTML ne sont pas toujours syntaxiquement correctes, les navigateurs ayant une certaine marge de tolérance.

Exemple :

Le code, syntaxiquement correct, d'une page HTML très simple est présenté ci-dessous :

Listing 5.2: Code d'une page HTML

```
<html>
<body>
<a href=''lien.html''>voir page lien.html</a>
</body>
</html>
```

Ce code peut être altéré sans que l'affichage n'en soit affecté. Ainsi, les codes ci-dessous permettent d'afficher la même page que le code présenté plus haut. Pourtant, dans le premier exemple, les guillemets fermant du lien HTTP sont absents et dans le deuxième exemple, c'est toute la fin du code qui a été supprimée.

Listing 5.3: Premier exemple de code altéré d'une page HTML

```
<html>
<body>
<a href=''lien.html>voir page lien.html</a>
</body>
</html>
```

Listing 5.4: Deuxième exemple de code altéré d'une page HTML

```
<html>
<body>
<a href='' lien.html ''>voir page lien.html
```

Or, c'est justement sur ces éléments (guillemets, balises) que le système conseiller se fonde pour extraire les URLs et déterminer si elles doivent être ou non modifiées. Des mécanismes de vérification, d'adaptation ont donc été développés pour permettre au système conseiller de détecter puis modifier correctement les URLs même lorsque le code est altéré.

5.3.4 Module Proxy

Principe de fonctionnement du module Proxy. Le module Proxy est l'autre approche proposée par notre architecture pour que les utilisateurs et le système conseiller puissent communiquer. Comme nous l'avons déjà indiqué paragraphe 5.2.1, pour utiliser ce module, l'utilisateur doit modifier la configuration de son navigateur et définir le système conseiller comme proxy. Une fois cette modification effectuée, l'utilisateur n'a plus rien à faire : toutes ses requêtes Web vont être dirigées vers le système conseiller.

Illustration du fonctionnement du module Proxy. Le fonctionnement du module Proxy est en partie identique à celui du module HTTP : le système conseiller reçoit des requêtes des utilisateurs, les fait suivre au site Web cible puis réceptionne la réponse du site cible qu'il renvoie au client.

Cependant, le fonctionnement du module Proxy est plus simple puisque, à la différence du module HTTP, aucune modification à la volée n'est nécessaire. En fait presque aucune : quelques lignes de code javascript, permettant l'ouverture des conseils au moment de la lecture de la page reçue du serveur Web cible, sont insérées ; ces modifications sont cependant tout à fait mineures par rapport aux modifications à la volée effectuées par le module HTTP. Des exemples de ces modifications sont présentés figure 5.3.

Il est à noter que le fait de déclarer le système conseiller en tant que proxy, entraîne la réception, par le système conseiller, de l'ensemble des requêtes de l'utilisateur. Le module Proxy effectue donc un tri dans les requêtes :

- Celles à destination du (ou des) site(s) cible(s), ainsi que les pages envoyées en retour par les serveurs Web subissent les manipulations nécessaires au fonctionnement du système conseiller (analyse de la requête, modifications de la page)

- Les autres sont envoyées directement au site Web concerné et sa réponse est transmise au client sans qu'aucun traitement particulier ne soit effectué.

Identification des utilisateurs et module Proxy. L'identification des utilisateurs du module Proxy s'effectue à l'aide de l'adresse IP. Pour chaque requête, l'adresse IP de l'utilisateur est extraite et la base de donnée du système conseiller permet de faire le lien utilisateur - adresse IP.

Page envoyée par le site cible et modifiée par le système conseiller

Déclaration de la fonction (javascript) permettant l'ouverture du conseil

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <script language="JavaScript"> var monwebconseil = null;var monmemo = null;
                                function opentip(URL,Name,features) {monwebconseil=window.open(URL,Name,features);}
  </script>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="author" content="Documentation Group">
  <meta name="email" content="docs@httpd.apache.org">
  <title>Welcome! - The Apache HTTP Server Project</title>
</head>
<body onload="javascript:monwebconseil.close()"
      onunload="opentip('http://marco:80/webconseil/tips/11198232333_01162330137.html',
                        'Conseil',
                        '|width=580,height=390,top=400,left=600,scrollbars,resizable=yes,location=no,menubar=no,status=yes');"
      bgcolor="#ffffff" text="#000000" link="#525d76">
  <p><a href="/"></a></p>
  <table border="0" width="100%" cellspacing="4">
    <tbody>
```

Téléchargement du conseil et ouverture dans une fenêtre adjacente
lors de l'ouverture de la page envoyée par le site cible

Fermeture de la fenêtre du conseil lors de la fermeture de la page envoyée par le serveur Web cible

FIG. 5.3 – Modifications d'une page HTML permettant l'ouverture d'une fenêtre conseil

Avantages du module Proxy. L'indéniable avantage de cette approche est sa simplicité. Mettre en place une telle architecture réseau est extrêmement simple puisque le système conseiller fait simplement office de relais, modifiant uniquement les entêtes des requêtes et faisant suivre les données.

Inconvénients du module Proxy. L'inconvénient majeur de cette approche est l'approximation due à l'usage de l'adresse IP : s'appuyer sur l'adresse IP de l'utilisateur revient à considérer comme une seule personne tous les utilisateurs de systèmes placés derrière un routeur NAT. Pour éviter cette approximation, une solution serait d'ajouter un numéro de session à chaque lien, mais on retrouve alors les inconvénients du module HTTP. En conséquence, nous avons choisi de conserver ce module et l'approximation inhérente à son fonctionnement.

Notons de plus que ce module peut être difficile à utiliser par certains utilisateurs : certains postes informatiques doivent impérativement définir un proxy d'entreprise pour naviguer sur le Web. Dans ces conditions, seul le module HTTP est utilisable.

5.3.5 Serveur Web et cache

Deux mécanismes ont été mis en place pour permettre une plus grande réactivité du système conseiller : un cache et un serveur Web.

Cache. Le cache permet de limiter le trafic réseau en stockant des fichiers (pages HTML, images, etc.) demandés par les utilisateurs. Pour chaque requête, le noyau du système conseiller vérifie auprès du cache si la page est contenue dans le cache puis, si c'est le cas, vérifie auprès du site cible si la page est à jour. Lorsque ces conditions sont remplies, c'est le fichier correspondant à cette page qui est transmis à l'utilisateur. Le système conseiller évite ainsi l'envoi d'une requête au serveur cible et l'attente de la réponse. Lorsque les conditions ne sont pas remplies (le fichier correspondant à la page est absent du cache ou trop ancien), le noyau du système conseiller efface l'ancienne version du fichier (s'il en existait une dans le cache) puis enregistre, dans un fichier, la page récupérée sur le serveur cible dans le cache.

Serveur Web. Un serveur Web est associé au système conseiller. Il permet d'alléger son fonctionnement en répondant à certaines requêtes. Ainsi, dans le cadre de la construction de conseils, le système conseiller opère comme suit :

1. Le module générateur de conseil construit le conseil (comme nous l'avons déjà indiqué, c'est une page HTML qui est produite).
2. Le noyau réceptionne ce conseil.
3. Il enregistre la page HTML correspondant au conseil dans un espace accessible par le serveur Web.

4. Il modifie la page à destination du client : lors de l'ouverture de cette page sur le navigateur de l'utilisateur, un code javascript sera activé, permettant le téléchargement de la page du conseil auprès du serveur Web du système conseiller et sa présentation à l'utilisateur (rappel : la figure 5.3 présente ces modifications).

5.3.6 Module de diagnostic

Le module de diagnostic assure le suivi de la navigation de l'utilisateur : pour un utilisateur donné, pour le modèle qu'il a choisi et pour une requête donnée, il renvoie au noyau l'arc du modèle sur lequel "se situe" l'utilisateur (rappel : au paragraphe 4.1.1, nous avons indiqué et explicité notre choix d'attacher les conseils aux arcs des modèles d'utilisation ; connaître l'arc sur lequel "se situe" l'utilisateur permet ensuite au système conseiller de déterminer le conseil à proposer (s'il y a)).

Pour déterminer l'arc sur lequel se trouve l'utilisateur, le module va répertorier les noeuds qui pourraient être à l'origine de l'arc recherché et ceux qui pourraient être à son arrivée. Plusieurs types de noeuds sont présents dans un modèle d'utilisation :

- Des noeuds classiques, caractérisés par un indice et des URLs.
- Des noeuds génériques (que nous qualifierons de "wildnodes" dans la suite de ce mémoire) : les URLs des noeuds génériques sont en partie ou complètement génériques (selon la place de l'"" dans l'URL) et représentent plusieurs URLs.

Exemple de fonctionnement des noeuds génériques :

Si l'URL d'un wildnode d'un modèle d'utilisation est 'http://www.apache.org/*', ce noeud modélisera toutes les pages du site Web apache (*i.e.*, toutes les pages dont l'URL débute par 'http://www.apache.org/' ; par exemple, 'http://www.apache.org/download.html').

Ce type de noeuds est utile puisqu'il offre la possibilité à l'expert de modéliser des phrases telles que : "d'où qu'il vienne, je veux présenter à l'utilisateur accédant à la page X, le conseil Y". Il rend cependant plus complexe la recherche de l'arc courant.

L'algorithme 2 présente le fonctionnement du module de diagnostic. Trois étapes sont nécessaires dans la recherche de l'arc correspondant à la requête de l'utilisateur :

1. Recherche des noeuds précédents (*i.e.*, les noeuds qui pourraient être le

noeud de départ de l'arc courant) (lignes 5 à 11 de l'algorithme 2) :

- Le noeud précédent a été enregistré avec les informations sur l'utilisateur dans la base de données. Il est récupéré (ligne 6).
- Les wildnodes sont parcourus et analysés : la fonction `est_une_forme_simplifiee` (ligne 8) effectue la comparaisons entre URL du wildnode et l'URL de la requête. Si l'URL du wildnode est une forme générique de celle de la requête (ce qui est le cas dans cet exemple), la fonction renvoie VRAI. On recherche, ainsi, dans ces lignes les wildnodes qui pourraient être considérés comme noeuds de départ de l'arc courant.

A l'issue de cette première étape, le système dispose de l'ensemble des noeuds du modèle susceptibles d'être le noeud de départ de l'arc recherché.

2. Recherche des noeuds courants (lignes 13 à 19 de l'algorithme 2) : comme dans l'étape précédente, il s'agit de passer en revue les noeuds du modèle d'utilisation (wildnodes compris) et de répertorier ceux qui pourraient être le noeud d'arrivée de l'arc courant.
3. Recherche de l'arc courant (lignes 21 à 33 de l'algorithme 2) : on recherche s'il existe un arc entre les noeuds de départ et d'arrivée répertoriés. Notons que l'on passe d'abord en revue les noeuds classiques (*i.e.*, ceux qui ne sont pas des wildnodes). Ce principe permet l'utilisation conjointe des URLs génériques et des URLs spécifiant précisément une page.

5.3.7 Module d'élaboration des conseils

Comme nous l'avons indiqué précédemment, (1) un conseil est un ensemble d'informations proposant à l'utilisateur des explications, des liens vers des pages susceptibles de l'intéresser ou des fonctionnalités absentes de la page et jugées utiles par l'expert et (2) c'est sous la forme d'une page HTML qu'il est présenté à l'utilisateur.

Au niveau informatique, "l'objet" conseil peut avoir différentes formes, c'est-à-dire être modélisé de différentes façons. Cependant, quel que soit la modélisation choisie, la phase d'élaboration des conseils peut se décliner comme suit :

1. Récupération de l'objet conseil.
2. Génération du code HTML correspondant au conseil.
3. Envoi au noyau du code HTML correspondant au conseil.

Nous verrons dans le chapitre suivant comment l'objet conseil peut être modélisé et quels outils peuvent être mis à disposition de l'expert pour l'aider à construire des conseils.

Algorithme 2 Algorithme du module de diagnostic

```
1: // initialisations
2: Noeuds[] noeuds_precedents_possibles[], noeuds_courants_possibles[]
3: Arc arc
4: //
5: // recherche des noeuds précédents potentiels
6: noeuds_precedents_possibles[] ← dernier_noeud_parcouru(utilisateur)
7: pour  $i = 1$  to  $i = nb\_wildnodes$  faire
8:   si wildnode = TOUS_LES_NOEUDS OR est_une_forme_simplifiee( wild-
     node, dernier_noeud_parcouru(utilisateur) ) alors
9:     noeuds_precedents_possibles[] ← wildnodes[i]
10:   fin si
11: fin pour
12: //
13: // recherche des noeuds courants potentiels
14: noeuds_courants_possibles[] ← rechercher_noeuds_possibles(modele,requete)
15: pour  $i = 1$  to  $i = nb\_wildnodes$  faire
16:   si wildnode[i] = TOUS_LES_NOEUDS OR
     est_une_forme_simplifiee(wildnode[i],requete) alors
17:     noeuds_courants_possibles[] ← wildnodes[i]
18:   fin si
19: fin pour
20: //
21: // recherche de l'arc
22:  $i \leftarrow 0$ ; trouve ← FAUX
23: tantque  $i < taille(noeuds\_precedents\_possibles)$  AND trouve = FAUX
  faire
24:    $j \leftarrow 0$ 
25:   tantque  $j < taille(noeuds\_courants\_possibles)$  AND trouve =
    FAUX faire
26:     noeud_precedent ← noeuds_precedents_possibles[i]
27:     noeud_courant ← noeuds_courants_possibles[j]
28:     arc ← arc_existe(noeud_precedent,noeud_courant)
29:     si arc ≠ NULL alors
30:       trouve ← VRAI
31:     fin si
32:   fin tantque
33: fin tantque
```

5.4 Synthèse

Nous avons présenté dans ce chapitre une architecture permettant la mise en oeuvre de l'approche présentée au chapitre 4. Cette architecture est structurée comme un proxy : le système conseiller est placé entre l'utilisateur et le serveur Web cible. Deux modules ont été développés permettant d'offrir deux options de connexion à l'utilisateur. D'une part, une connexion de type proxy où l'utilisateur définit dans son navigateur le système conseiller comme étant son proxy. Dès lors, toutes ses requêtes passent par le système conseiller. D'autre part, une connexion classique (de type serveur Web) où l'utilisateur se connecte sur le système conseiller qui se charge de rediriger ses requêtes vers le serveur cible. Nous avons créé plusieurs modules pour suivre la navigation de l'utilisateur (module de diagnostic), produire des conseils (module d'élaboration des conseils) ou accélérer le fonctionnement du système conseiller (module cache, serveur Web du système conseiller).

Créer des modèles et des conseils est une tâche difficile. Nous verrons dans le chapitre suivant quels outils peuvent être proposés à l'expert pour l'aider dans cette tâche.

Chapitre 6

Assister la création des modèles d'utilisation

Les différentes étapes d'élaboration d'un système conseiller sont (cf. paragraphe 4.2) :

0. Création/adaptation/modification d'un système conseiller générique (si une telle architecture n'est pas disponible).

1. Identification et création des modèles de tâches.
 - (a) Analyse du site (structure et usages).
 - (b) Identification des tâches.
 - (c) Création des modèles de tâches.
2. Création des conseils.
3. Positionnement des conseils sur les modèles de tâches et création des modèles d'utilisation.
4. Instanciation du système conseiller générique : installation des modèles d'utilisation, adaptation réseau.
5. Analyse du fonctionnement du couple système conseiller - site cible : analyse des usages courants et si besoin (en cas, par exemple, de découverte d'usages non prévus et qui ne correspondent pas aux attentes des concepteurs du site cible) nouvelle itération à partir de l'étape 0 ou 1.

Les tâches 1 à 5 sont à la charge de celui que nous avons appelé l'expert (cf. paragraphe 4.2.2) et doivent être répétées pour chaque nouveau site cible.

Réaliser ces tâches peut être ardu. Cependant, la réalisation de certaines tâches peut être facilitée par des outils spécifiques. Dans ce chapitre, nous présentons des exemples d'outils que nous avons développés et qui permettent d'assister l'expert dans la réalisation des tâches 1.a, 1.c, 2, 3 et 4. Comme nous l'avons indiqué au paragraphe 4.2.2, analyser les usages d'un site Web et identifier des usages prototypiques (autres que ceux définis par les concepteurs du site) est un problème en soi. Nous n'avons pas examiné ce problème en tant que tel. Cependant, pour faire fonctionner nos prototypes et illustrer notre approche, nous avons été obligé de tenir le rôle d'expert et c'est dans ce contexte que nous avons étudié comment l'analyse des usages du site (étape 1.a) pouvait être assistée. Nous n'avons pas abordé, en revanche, l'assistance informatique qu'il est possible de proposer à l'expert pour identifier les tâches qui doivent être assistées (étape 1.b).

Les outils présentés ont été utilisés pour créer des modèles dans le cadre d'un contrat (cf. paragraphe 7.1.1) et dans le but d'illustrer notre approche. C'est un outil standard, Protégé 2000 [pro], un éditeur d'ontologies, qui a d'abord été utilisé. L'utilisation de Protégé 2000 a mis en valeur un certain nombre de problèmes. Nous avons alors exploré une nouvelle voie : un environnement ad-hoc a été élaboré. On trouvera trace de ces deux phases dans les paragraphes ci-dessous : pour chaque tâche, plusieurs exemples d'outils qu'il est possible de proposer à l'expert sont présentés.

6.1 Assister l'identification et la création des modèles de tâches (étape 1)

6.1.1 Assister l'analyse des usages du site (étape 1.a)

Ainsi que nous l'avons déjà indiqué (cf. paragraphe 2.4.2.2), les logs d'exécution d'un serveur Web sont une source d'information considérable pour qui s'intéresse aux usages d'un site Web. Pour l'expert qui doit répondre à des questions telles que : le modèle que je veux créer correspond-il à un usage réel du site ? ou plus précisément : y a-t-il réellement des passages de telle partie du site à telle partie du site, de telle page à telle page ? les logs sont, bien entendu, très riches en information et méritent d'être analysés.

Dans le cadre de nos travaux, nous avons signé une convention avec la fondation Apache et récupéré 16 mois de logs de leurs serveurs Web. Ces logs représentent une masse volumineuse d'information : par exemple, en juillet 2005 (mois présentant une fréquentation moyenne du site), il a été enregistré quelque 111 millions de requêtes, ce qui représente un volume de données de 18 Go. Avant toute

analyse, des pré-traitements doivent être effectués : d’une part, dans la convention signée avec la fondation Apache, nous nous sommes engagés à travailler sur des fichiers anonymisés (les adresses IP des utilisateurs du site sont transformées en un indice). D’autre part, ces recherches s’effectuent en tenant compte des sessions des utilisateurs. Une session d’un utilisateur est une visite d’un utilisateur dans un site particulier. Les recherches que nous effectuons se font dans chacune de ces sessions. Cependant, les logs, de par l’enregistrement séquentiel des requêtes, masquent les sessions. Un second traitement, de reconstruction cette fois, est à effectuer sur les logs avant de pouvoir les utiliser. Nous présentons, en annexe, le détail de ces pré-traitements effectués sur les logs.

Extraire de ces données des usages prototypiques est un travail à part entière et n’est pas l’objet de nos travaux. Cependant, dans le but d’illustrer et d’exemplifier notre approche, nous avons créé des outils pour aider l’expert dans plusieurs tâches d’analyse des logs. Ces tâches sont détaillées ci-dessous.

- Recherche d’une séquence. L’expert recherche une séquence de pages : par exemple, suite à l’analyse d’interviews ou de formulaires remplis par des utilisateurs, l’expert pressent que les utilisateurs parcourent successivement les pages A, B, C puis D. Il crée dans un modèle les noeuds correspondant à cette séquence qu’il veut valider. Après avoir sélectionné les noeuds dans le modèle, l’expert peut rechercher automatiquement cette séquence dans les fichiers de logs. En réponse, le système renvoie le nombre de séquences trouvées ainsi que plusieurs informations qui permettent de juger de la qualité du chiffre renvoyé (ces informations permettent notamment à l’expert de vérifier que le fichier de logs contient suffisamment d’enregistrements pour que les calculs effectués soient représentatifs) :
 - Le nombre d’utilisateurs trouvés dans le fichier de logs.
 - Le nombre de sessions trouvées dans le fichier de logs.
 - Le nombre de requêtes trouvées dans le fichier de logs.

Ainsi, si on recherche le nombre de passages de la page
“www.apache.org/index.html” à la page
“httpd.apache.org/index.html”, le 31 mars 2005, l’outil renvoie :

Etat : termine et OK
Nombre de sequences cibles trouvees : 9255
Informations complementaires :
Nombre d'utilisateurs : 167558

Nombre de sessions : 225493 Nombre de requetes HTTP : 1105350
--

- Recherche des pages accédées après une page donnée. L'expert s'interroge sur les pages les plus accédées après une URL donnée. Dans l'outil que nous avons développé, l'expert saisit l'URL de départ puis sélectionne le fichier de logs dans lequel il veut effectuer la recherche. Par exemple (les chiffres présentés dans ce paragraphe sont issus de l'analyse des logs de mars 2005 du serveur Apache) : les utilisateurs qui accèdent à la page httpd.apache.org/download.cgi vont ensuite :

- Pour 8% des sessions, sur la page archive.apache.org/dist/httpd.
- Pour 8% des sessions, sur httpd.apache.org/docs-2.0.
- etc.

De plus, comme dans l'exemple ci-dessus et comme pour tous les outils présentés ci-dessous, le système renvoie des informations (nombre d'utilisateurs, nombre de sessions, nombre de requêtes) permettant à l'expert de juger de la pertinence de l'information recueillie.

- Recherche des pages accédées après une page donnée, deuxième version. Cette deuxième version a été développée dans l'optique d'être utilisée dans des sites qui, comme le site Apache, présentent une structure en sous-sites (chaque projet a son propre site et la fondation Apache fédère l'ensemble de ces sites avec son nom de domaine apache.org). A la différence de l'outil proposé ci-dessus, la recherche des pages se limite au site de la page de départ (dans des sites comme celui de la fondation Apache, la recherche se limite donc à un sous-site, c'est-à-dire au site d'un projet). Par exemple (toujours d'après l'analyse des logs de mars 2005 du serveur Apache) : depuis la page httpd.apache.org/index.html, les utilisateurs accèdent à :

- Dans 35% des sessions, à la page httpd.apache.org/download.cgi.
- Dans 19% des sessions, à la page httpd.apache.org/docs-2.0.
- etc.

- Recherche des pages accédées avant une page donnée. A l'inverse des deux outils présentés ci-dessus, on recherche les pages parcourues avant une page donnée.
- Recherche de liens entre différents sous-sites. Un site complexe est parfois composé de sous-sites. Par exemple, le site Apache est structuré en sous-sites : httpd.apache.org, cocoon.apache.org, lenya.apache.org etc. Dans les

fichiers de logs on trouve les accès à tous ces sous-sites. Différents types de recherches sont proposés :

- Recherche des sous-sites parcourus dans une même session. Le système recherche les sous-sites parcourus simultanément au cours d'une même session. Un calcul statistique permet de donner des résultats tels que :
 - Dans 3% des sessions présentant un parcours dans deux sous-sites ou plus, les sites parcourus étaient jakarta.apache.org et struts.apache.org.
 - Dans 2% des sessions présentant un parcours dans deux sous-sites ou plus, les sites parcourus étaient jakarta.apache.org et archives.apache.org.
 - etc.
- Recherche de liens entre les sous-sites. Le système recherche les passages d'un sous-site à un autre. Les résultats obtenus sont, par exemple :
 - Dans 14% des sessions, l'utilisateur du sous-site cocoon.apache.org va passer sur le sous-site jakarta.apache.org.
 - Dans 11% des sessions, l'utilisateur du sous-site lenya.apache.org va passer sur le sous-site cocoon.apache.org.
 - etc.
- Recherche de liens entre un sous-site particulier et les autres sous-sites. Il est demandé à l'expert de spécifier un sous-site particulier. Le système cherche ensuite les sous-sites accédés depuis ce sous-site. Les résultats obtenus sont, par exemple : les utilisateurs qui accèdent au sous-site lenya.apache.org vont ensuite :
 - Dans 7% des sessions, dans le sous-site www.apache.org.
 - Dans 3% des sessions, dans le sous-site wiki.apache.org.
 - etc.

Comme on le voit, il est assez facile de créer des outils d'analyse proposant différentes informations utiles. Une méthodologie et des outils spécifiques restent cependant à construire. Il s'agit là de perspectives de recherche naturelles pour nos travaux.

6.1.2 Assister la création des modèles de tâches (étape 1.c)

Les étapes 1.a et 1.b de la séquence d'étapes menant à l'élaboration d'un système conseiller présentée plus haut ont pour objet de permettre à l'expert de ré-

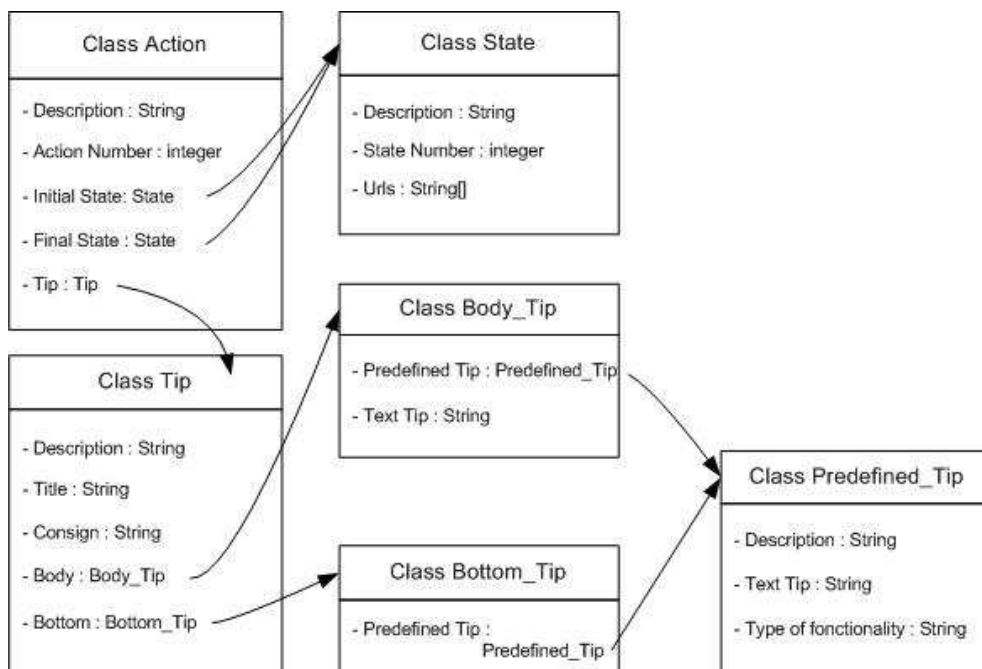


FIG. 6.1 – Les classes génériques

pondre à des questions telles que :

- Comment le site Web cible est-il organisé ?
- Comment le site est-il utilisé ? *i.e.*, quelles sont les pages les plus utilisées ? les moins utilisées ? quelles séquences sont observées ?
- Quelles sont les objectifs des utilisateurs lorsqu'ils accèdent au site ?
- Quels sont les usages prototypiques constatés du site ?
- Pour quelles tâches est-il souhaitable de proposer une aide ?

L'étape suivante (étape 1.c) consiste en la création proprement dite des modèles : création des noeuds et des arcs. Les outils présentés ci-dessous sont des exemples d'outils informatiques qui peuvent être développés pour assister l'expert pendant l'élaboration des modèles.

6.1.2.1 Approche 1 - Utilisation de Protégé 2000

Principe. Nous avons utilisé Protégé 2000 [pro] pour définir un ensemble de classes génériques qui, mises à disposition de l'expert, peuvent être instanciées et servir de base à son travail. Ces classes génériques modélisent les noeuds, les

arcs, les conseils et les fonctionnalités. La figure 6.1 présente la structure de ces classes génériques : la classe “State” (qui modélise les noeuds, c’est-à-dire les pages Web), la classe “Action” (qui modélise les requêtes HTTP qui permettent de passer d’une page Web à une autre. Ce passage est modélisé par un lien entre un état initial et un état final) et la classe “Tip” (qui modélise les conseils associés aux actions).

Pour créer des modèles de tâches, il suffit d’instancier les classes génériques en fonction d’un modèle d’utilisation particulier.

Apports. Cette approche permet à l’expert de bénéficier d’un cadre de travail : les relations entre les noeuds, les arcs, les conseils sont créées. L’expert peut ainsi concentrer son travail sur le fond : quelles tâches assister, quels noeuds choisir, quels liens établir entre eux, quels conseils donner. De plus, Protégé 2000 autorise l’exportation au format RDF Schema (RDFS) [BG02] des modèles. Le format RDFS est un format standard qui offre compatibilité et évolutivité. Une fois créés, les modèles sont d’ailleurs exportés dans ce format pour pouvoir être interprétés par le système conseiller.

Limites. L’utilisation de Protégé 2000 ne permet pas d’avoir un aperçu graphique du graphe : il est demandé à l’expert d’établir le lien entre deux noeuds en saisissant le numéro du noeud de départ puis celui du noeud d’arrivée. Cette gestion du graphe peut être source de difficultés pour des sites Web complexes où construire un modèle de tâche, amène à créer de nombreux noeuds et de nombreux arcs. La gestion des modèles avec Protégé 2000, si elle convient pour créer des graphes de taille limitée, est inconfortable et difficile lorsque le modèle s’agrandit.

La figure 6.2 présente un exemple d’instanciation des classes génériques avec Protégé 2000.

6.1.2.2 Approche 2 - Environnement ad hoc

Principe. Une autre possibilité est de développer un environnement informatique ad hoc pour créer puis manipuler les modèles. A titre d’exemple, nous avons développé un prototype d’un tel environnement. Le développement a été fait en Java [Jav] et nous nous sommes appuyés sur JGraph [JGr] pour tout ce qui touchait à la création et à la manipulation des noeuds et des arcs. Nous avons présenté au chapitre 4, un exemple de modèle d’utilisation créé avec cet environnement (cf. figure 4.4).

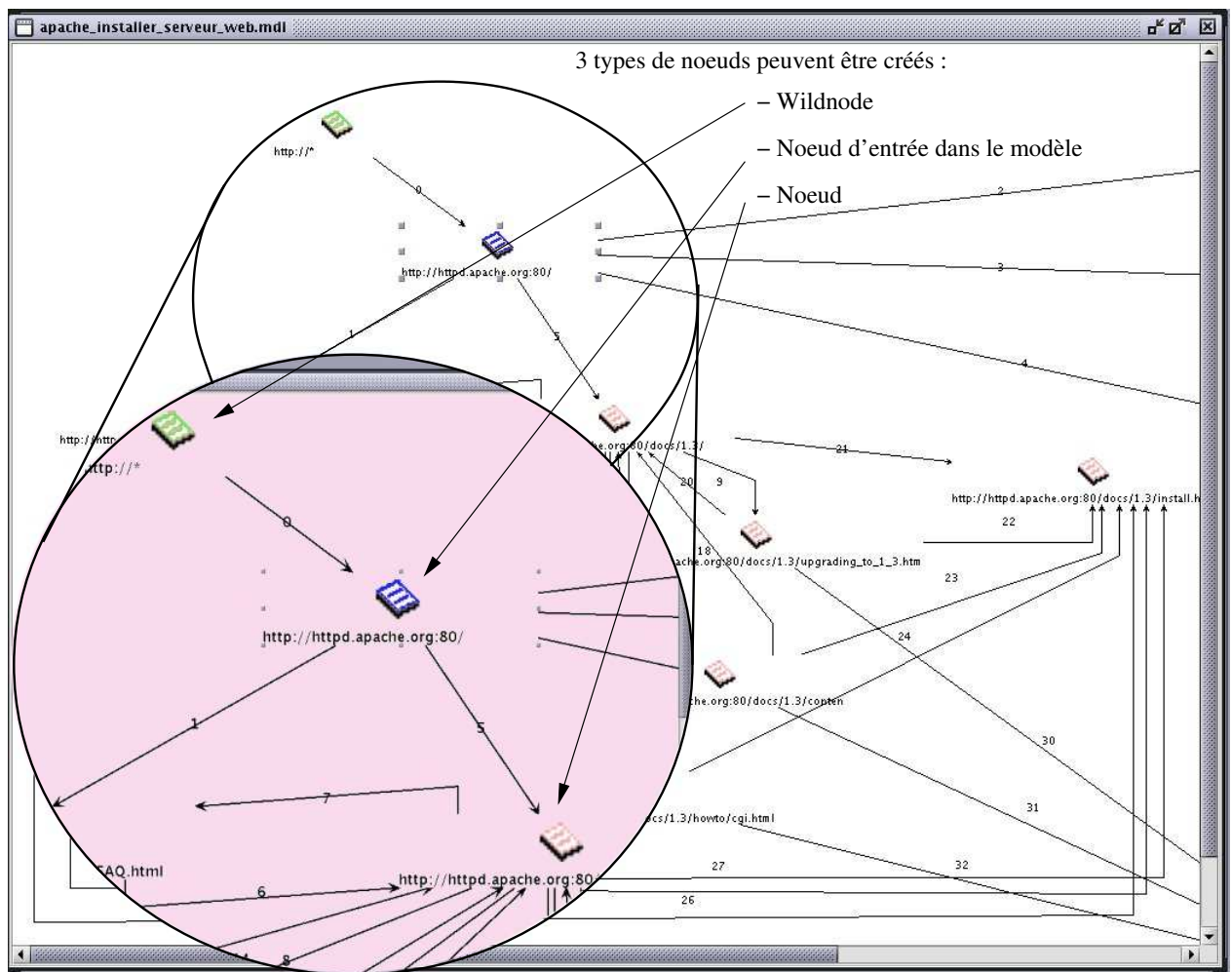


FIG. 6.4 – Les différents noeuds d'un modèle d'utilisation

Assister la tâche de création des modèles. Dans ce paragraphe, nous détaillons comment l’outil que nous avons développé assiste l’expert dans la tâche de création des modèles. Les tâches relatives à la manipulation des modèles (changement de nom, enregistrement, affichage, etc.) seront détaillées dans le paragraphe suivant. Plusieurs modes de création des modèles sont proposés :

- **Création manuelle.** La création manuelle des modèles propose à l’expert de créer manuellement les noeuds puis les arcs qui les relient : pour chaque noeud, l’expert va devoir spécifier un numéro puis les URLs qui lui sont associées.

Notons que le système vérifie que chaque URL est unique dans le modèle, vérification qui permet ensuite d’identifier, sans équivoque, le noeud sur lequel se trouve l’utilisateur.

Plusieurs types de noeuds peuvent être créés :

- Le noeud classique. Le noeud classique ne présente pas de caractéristique particulière ; l’expert renseigne simplement les champs numéro et URLs du formulaire de création. La figure 6.3 présente une copie écran de la fenêtre de création de ces noeuds.
- Le noeud d’entrée du modèle. La création du noeud d’entrée du modèle est semblable à celle du noeud classique. L’expert coche juste une case (cf. figure 6.3) indiquant au système que c’est vers ce noeud que doit être dirigé l’utilisateur choisissant le modèle.
- Le wildnode. Les wildnodes ont déjà été définis au paragraphe 5.3.6. Rappelons simplement qu’ils permettent de représenter plusieurs pages. Par exemple, le wildnode dont l’URL est “`http://www.apache.org/*`” représente l’ensemble des pages du site `www.apache.org`.

La figure 6.4 présente un modèle d’utilisation dans lequel ces différents noeuds ont été insérés.

- **Création assistée n°1.** Alors que le mode de création précédent était tout à fait manuel, le mode de création assistée n°1 tend à simplifier la création des noeuds : l’expert parcourt le site à l’aide de son navigateur et les URLs qu’il parcourt sont affichées en temps réel dans une fenêtre spécifique. Pour créer un noeud dans l’éditeur, l’expert fait simplement glisser les URLs affichées dans cette fenêtre vers la fenêtre d’édition. Comme dans le mode précédent, la création des arcs reste totalement manuelle.

La figure 6.5 présente un tel exemple de création.

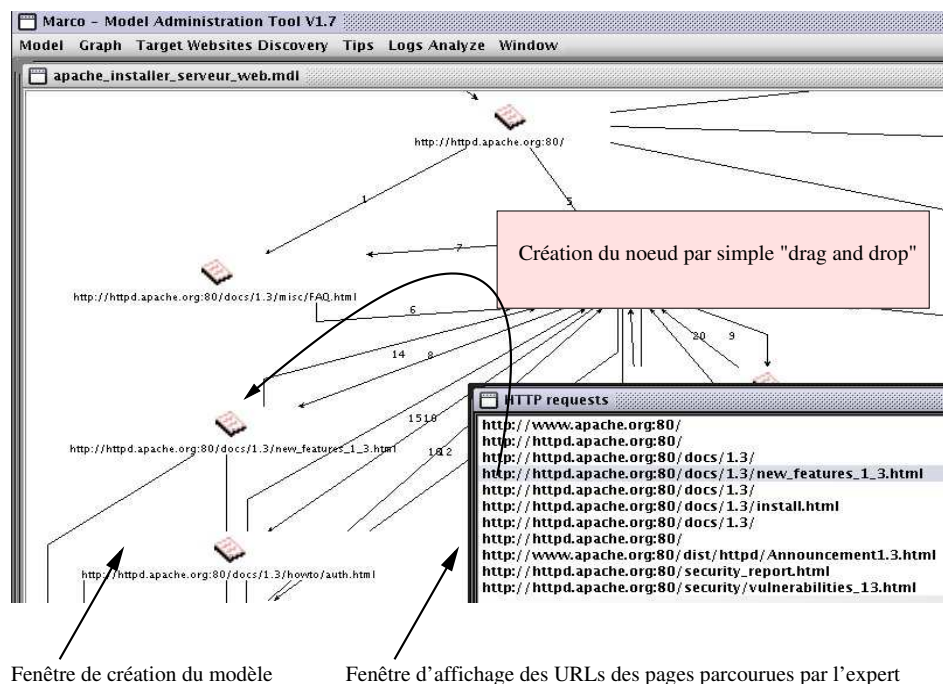


FIG. 6.5 – Création d'un noeud par simple "Drag and Drop"

- Création assistée n°2. Dans le mode de création assistée n°2 c'est le système conseiller qui crée le graphe du modèle d'utilisation : l'expert définit l'URL de la page qui servira de point de départ au parcours automatique, la profondeur de parcours souhaitée ainsi que l'algorithme qu'il souhaite utiliser pour organiser les nouveaux noeuds (par exemple, des algorithmes tels que "Spring Embedded Layout algorithm" ou encore "Circle Layout algorithm" sont proposés). Le système crée alors automatiquement le graphe (noeuds et arcs) et le présente à l'expert en utilisant l'algorithme choisi. L'expert peut ensuite simplifier le graphe et l'adapter à ses besoins.

La figure 6.6 présente la page permettant de saisir les différentes caractéristiques et de lancer l'exploration.

Assister les tâches de manipulation des modèles. D'autres activités de l'expert, proches des tâches de création de modèles, concernent plutôt la manipulation des modèles. Elles peuvent être, elles aussi, assistées à l'aide d'outils informatiques. Ainsi, à titre d'exemple, nous avons développé, dans l'environnement ad

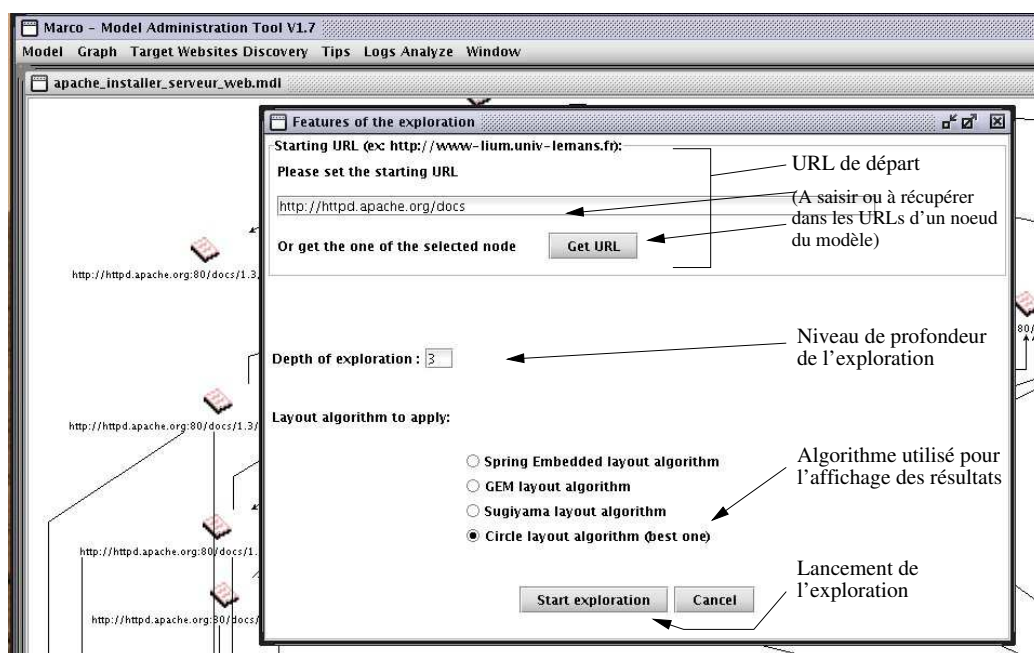


FIG. 6.6 – Exploration automatique du site cible

hoc, les fonctions suivantes :

- Ouverture, fermeture, enregistrement des modèles. Classiquement, il est possible d'ouvrir, de fermer, d'enregistrer les modèles et de changer leur nom.
- Gestion de l'affichage du graphe. L'étude de sites complexes risque d'amener l'utilisateur à construire des modèles d'utilisation contenant de nombreux noeuds et arcs. Dans le but d'aider l'expert à manipuler ces objets, nous proposons des techniques pour :
 - Réorganiser automatiquement les noeuds et les arcs. Plusieurs algorithmes permettant une réorganisation automatique du graphe sont proposés :
 - L'algorithme "Spring Embedded Layout algorithm".
 - L'algorithme "GEM Layout".
 - L'algorithme "Sugiyama Layout algorithm".
 - L'algorithme "Circle Layout algorithm".
 - Visualiser le graphe selon différentes échelles. L'affichage du graphe est configurable et permet de cibler une partie du modèle ou au contraire de le visionner dans son ensemble.

- Exportation sous forme d'image. Il est possible d'exporter le modèle sous la forme d'images au format jpeg (comme pour la visualisation, différents facteurs d'échelle sont proposés).

6.2 Assister la création des conseils (étape 2)

De par la taille des sites et la variété des objectifs possibles, la tâche de création de conseils peut être source de difficultés. Pour aider l'expert à construire ces conseils, il est possible de proposer des outils informatiques qui permettent de :

1. Éviter à l'expert la réécriture de conseil : des conseils similaires peuvent être pertinents à plusieurs endroits du modèle. Bénéficier de mécanismes qui vont éviter à l'expert de réécrire chaque conseil est donc judicieux.
2. Concevoir un objet conseil élaboré : cet objet doit pouvoir être scindé en plusieurs parties (en d'autres termes, on peut concevoir l'objet conseil comme un assemblage de conseils élémentaires). Par exemple, on peut imaginer que sur chaque conseil, l'expert veuille proposer un aperçu du modèle d'utilisation montrant le parcours et/ou la position de l'utilisateur. Il peut être intéressant de définir un conseil élémentaire assumant cette tâche, conseil élémentaire qui pourra être un composant de tous ou de certains des conseils élaborés pour le modèle.

Nous proposons ci-dessous deux approches qui permettent d'illustrer l'apport de l'outil informatique pour ce type de tâche.

6.2.1 Approche 1 - Utilisation de Protégé 2000

Principe. Nous avons utilisé Protégé 2000 pour proposer à l'expert une base de classes, *i.e.*, un environnement de travail, à partir duquel il puisse créer des conseils. Dans ce but, nous avons créé plusieurs classes (présentées ci-dessous) qui peuvent être instanciées pour créer des conseils.

La classe "Tip" est utilisée pour modéliser les conseils (cf. figure 6.1 présentée plus avant dans le cadre de l'utilisation de Protégé 2000 pour créer des modèles de tâches). Elle présente plusieurs champs textes :

- Une description du conseil proposé à l'usage du concepteur.
- Un titre qui sera présenté à l'utilisateur.
- Une consigne qui sera présentée à l'utilisateur.

et deux sous-classes: "Body_Tip" et "Bottom_Tip".

La classe “Body_Tip” permet de proposer texte et/ou fonctionnalités (via la classe “Predefined_Tip”).

La classe “Bottom_Tip” ne permet de proposer que des fonctionnalités (via la classe “Predefined_Tip”).

Enfin, la classe “Predefined_Tip” permet à l’expert de disposer de fonctionnalités, développées pour un site donné ou non (*i.e.*, fonctionnelles quel que soit le site cible) et intégrées au système conseiller. Ces fonctionnalités peuvent être adaptées au contexte à l’aide du champ “Text Tip”. Le champ “Type of functionality” est utilisé par le système conseiller lors de la création du conseil pour identifier la fonctionnalité à exécuter.

Ces différents champs et classes permettent de proposer à l’expert une structure de classes formatant le conseil. La partie haute du conseil (classe “Body_Tip”) permet de proposer à l’utilisateur une consigne puis du texte et différentes fonctionnalités. La partie basse du conseil (classe “Bottom_Tip”) peut proposer des fonctionnalités moins liées à la page courante, *i.e.*, des fonctionnalités qui manquent au site et qu’on pourra retrouver sur plusieurs pages.

La figure 6.7 présente un exemple de conseil créé avec Protégé 2000 et présenté à l’utilisateur (en fond, la page du site cible présentée à l’utilisateur simultanément au conseil).

Apports. Cette architecture de classes présente plusieurs avantages :

1. Le conseil est pré-formaté, ce qui présente deux avantages :
 - (a) Le travail sur la forme du conseil est en partie déjà effectué et l’expert peut se consacrer au conseil lui même.
 - (b) La présentation du conseil reste uniforme tout au long de la navigation de l’utilisateur, lui facilitant la lecture des conseils.
2. La création d’une classe “conseil” composée de sous-classes permet d’augmenter la réutilisabilité des objets.
3. La création d’une classe “conseil prédéfini” offre à l’expert la possibilité de disposer de fonctionnalités prédéfinies, développées ou non pour un site particulier.

Limites. L’inconvénient majeur de cette approche est sa rigidité : pour faciliter le travail de l’expert, une organisation en classes et en sous-classes a été proposée. Cependant, si elle permet une certaine réutilisabilité et un affichage formaté facilitant la prise en main du système conseiller par l’utilisateur, elle ne permet pas de proposer des conseils d’une grande variété.

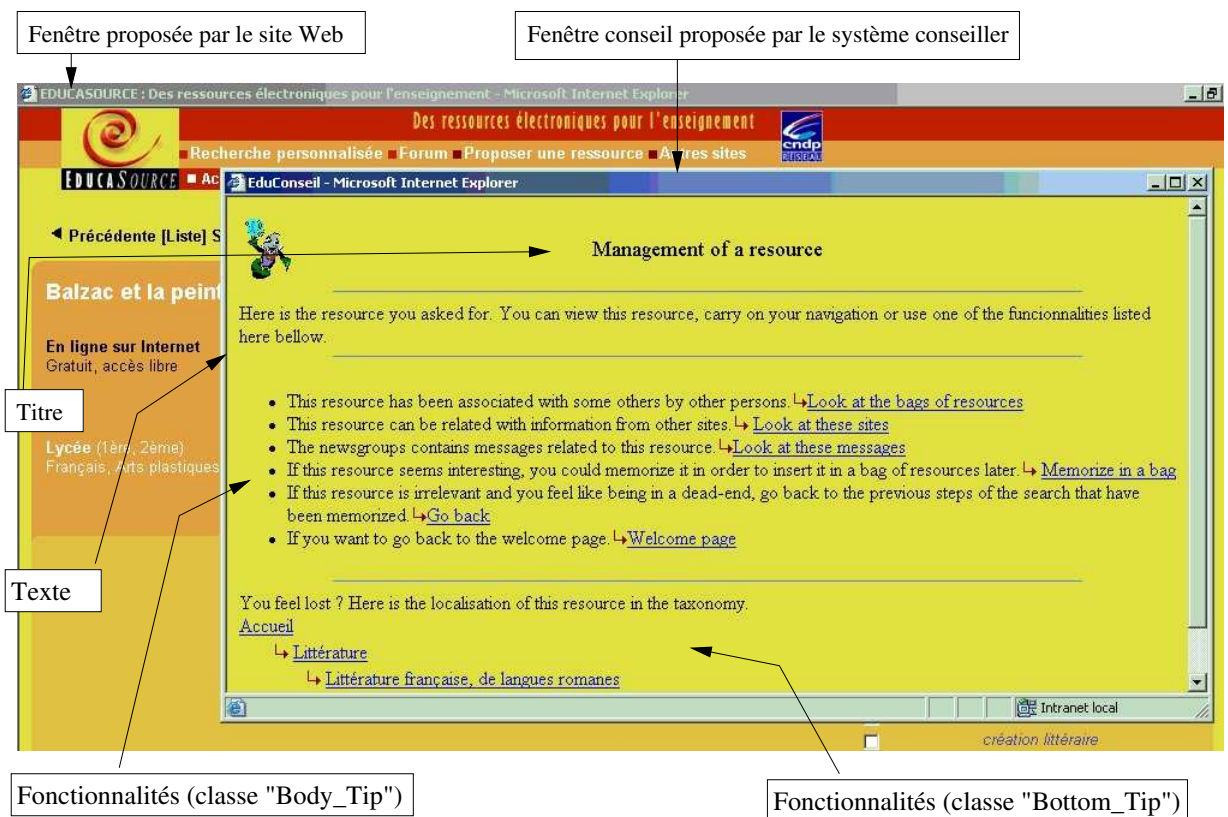


FIG. 6.7 – Exemple de conseil créé avec Protégé 2000

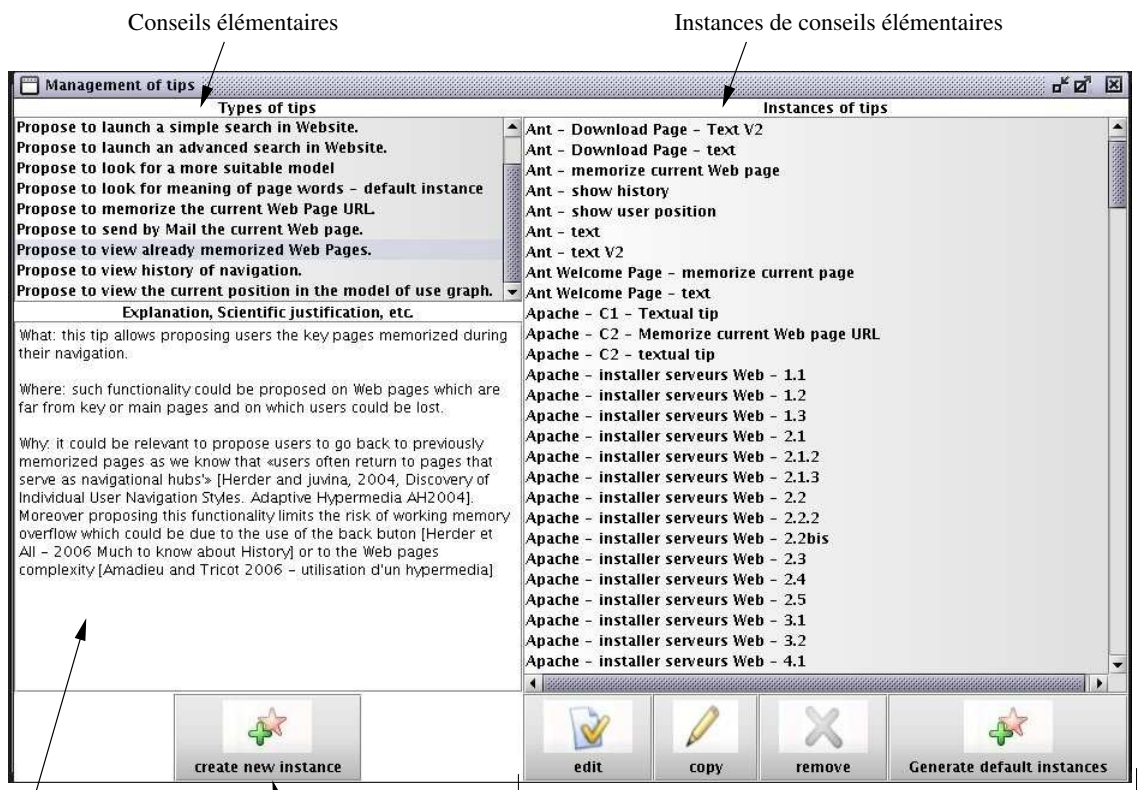
Un deuxième inconvénient de cette approche est la mise en oeuvre des fonctionnalités proposées à l'expert. Ces fonctionnalités sont intégrées au système conseiller. Ainsi, la création de nouvelles fonctionnalités et la modification des fonctionnalités déjà existantes, nécessitent (1) des compétences en programmation et (2) de modifier le code du système conseiller. Nous verrons dans le paragraphe suivant comment on peut pallier cet inconvénient.

6.2.2 Approche 2 - Environnement ad hoc

6.2.2.1 Principe

Il s'agit de proposer un mécanisme permettant à l'expert :

1. La création d'objets "Conseil" pouvant être instanciés, c'est-à-dire des objets qui peuvent être définis de façon générique par l'expert puis, qui seront



Création d'une instance d'un conseil élémentaire

Manipulation des instances

Informations relatives au conseil élémentaire sélectionné
 Ces champs permettent d'enregistrer des informations telles que le principe du conseil, son champ d'application ou encore les bases scientifiques sur lesquelles il repose.
 Ces informations ne sont pas obligatoirement renseignées.
 Elles servent uniquement de support à la réflexion de l'expert.

FIG. 6.8 – *Instanciations des conseils élémentaires*

précisés, détaillés, instanciés en fonction de situations données pour ensuite être proposés à l'utilisateur. En d'autres termes, l'objectif de cette approche est de permettre à l'expert de définir des conseils génériques, réutilisables, construits pour une situation typique (par exemple, l'arrivée sur une page générée par un moteur de recherche et présentant une liste de résultats) et qui seront instanciés en fonction d'une situation particulière.

2. La création de conseils élémentaires, sortes de briques qui pourront être réutilisés dans les conseils génériques. L'objectif est de faciliter la réutilisabilité.

6.2.2.2 Définitions

Conseils élémentaires. Les conseils élémentaires sont des objets Java définis par une interface (le terme interface définit, dans les langages objets, une description des interactions de l'objet avec son environnement). Ils sont chargés au démarrage de l'environnement ad hoc à disposition de l'expert. De nouveaux conseils élémentaires peuvent ainsi être développés et facilement intégrés au système conseiller. Chaque objet est obligatoirement créé avec des valeurs par défaut ce qui permet à l'expert de disposer immédiatement d'instances par défaut de ces conseils élémentaires (chaque conseil élémentaire a donc une instance par défaut). Ensuite, en fonction des besoins de l'expert, d'autres instances peuvent être créées. La figure 6.8 présente l'outil proposé à l'expert. Dans la partie gauche, sont présentés les conseils élémentaires. Les instances de ces objets sont présentées dans la partie droite.

Des exemples de conseils élémentaires sont :

- Proposer un texte.
- Proposer la mémorisation du lien courant.
- Proposer d'envoyer par mail un lien vers la page courante.
- Proposer un retour vers une page préalablement mémorisée.
- Proposer un retour vers la dernière page mémorisée.
- Proposer la visualisation de l'historique de navigation de l'utilisateur avec affichage miniature des pages parcourues.
- Proposer la visualisation de la définition d'un terme de la page.
- Proposer le lancement d'une recherche avancée (*i.e.*, avec des paramètres tels que la langue de la page, sa date de modification, le domaine de recherche, le format de la page recherchée, etc.).
- Proposer le lancement d'une recherche simple (où seuls des mots clés sont à saisir).
- Proposer la recherche d'un modèle d'utilisation plus conforme à la navigation de l'utilisateur que celui choisi en début de session.
- Proposer la visualisation de la position de l'utilisateur dans le modèle courant.

Pour permettre la construction de conseils élaborés qui s'adaptent automatiquement à leur environnement, il est proposé à l'expert, lors de la création du conseil élémentaire, d'utiliser des variables : ces variables peuvent être relatives à la page parcourue par l'utilisateur ou relatives au modèle d'utilisation choisi. Les variables que nous avons définies sont :

- Le nom du modèle courant (information relative à la page courante).

- L'URL de la page courante (information relative à la page courante).
- Le titre de la page courante (information relative à la page courante).
- L'indice de rappel (nombre de noeuds différents parcourus dans le modèle / nombre de noeuds du modèle) (information relative au modèle d'utilisation choisi).
- L'indice d'économie (nombre de noeuds différents parcourus dans le modèle / nombre de noeuds ouverts dans le modèle) (information relative au modèle d'utilisation choisi).
- etc.

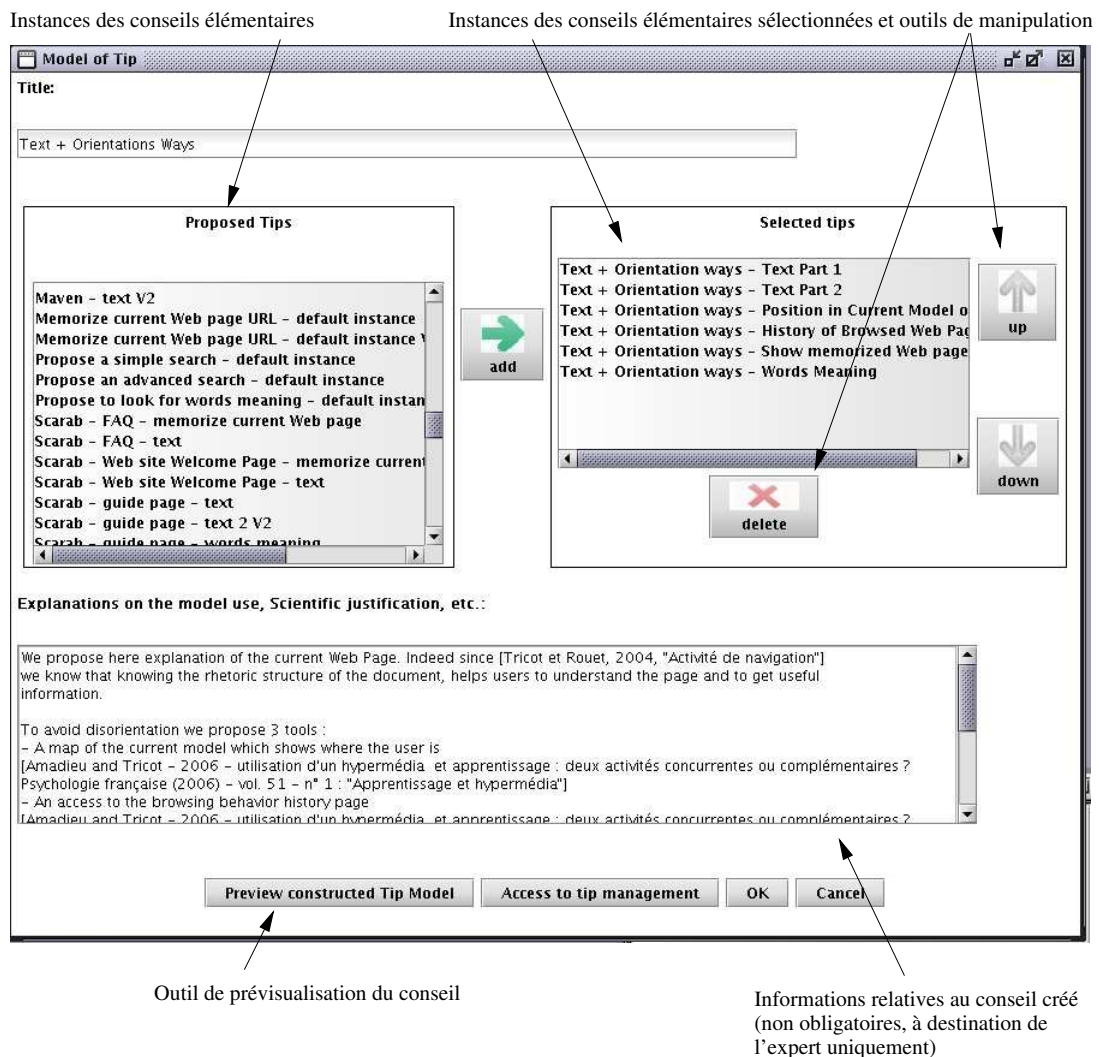


FIG. 6.9 – *Élaboration d'un conseil*

Conseils. Les conseils sont créés en associant plusieurs conseils élémentaires (plus précisément en associant des instances de conseils élémentaires). Cette association, illustrée par la figure 6.9, permet de construire un conseil générique. Ce conseil présente un texte initial, des fonctionnalités particulières et est destiné à être utilisé dans un type de situation particulière. Des exemples de ces situations sont :

- L'utilisateur accède à une page clé.
- L'utilisateur a parcouru une séquence particulière de pages (par exemple, il est revenu x fois sur la même page).
- L'utilisateur accède à une page proposant différentes orientations parmi lesquelles il doit faire un choix.
- etc.

Lorsque une telle situation est identifiée, *i.e.*, lorsque l'expert détermine qu'un des conseils génériques correspond à la situation courante, il peut créer une instance de ce conseil générique. C'est cette instance qui, une fois adaptée à la situation courante, sera proposée à l'utilisateur.

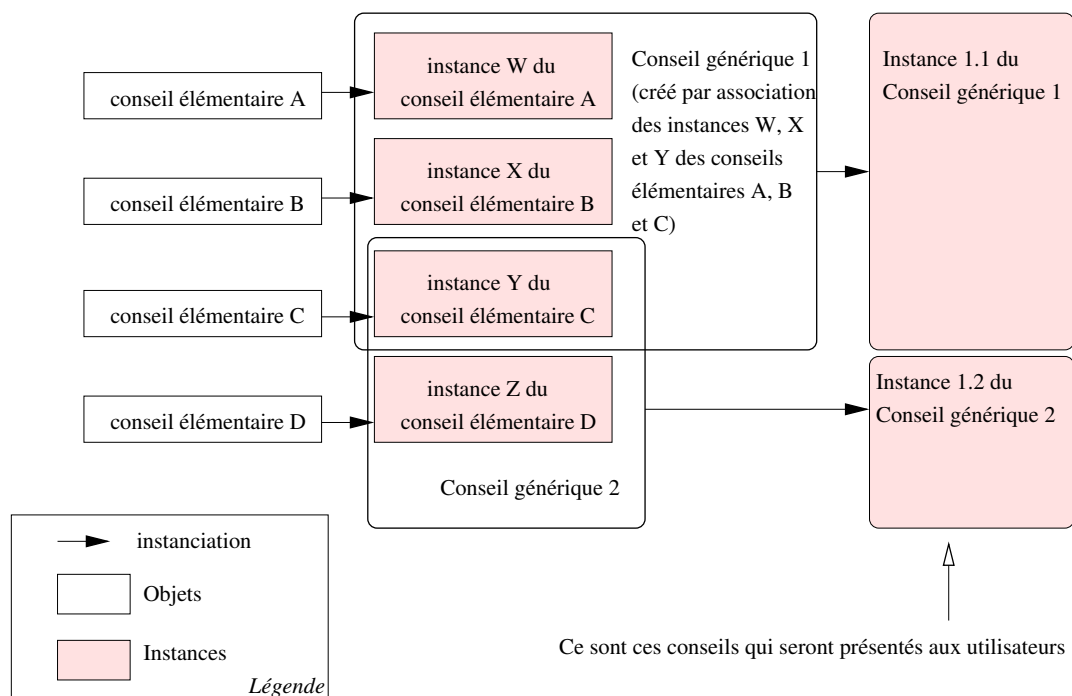


FIG. 6.10 – *Conseils élémentaires, conseils génériques et instances*

Synthèse. La figure 6.10 illustre l’articulation entre les conseils élémentaires, leurs instances, les conseils proprement dits et leurs instances.

Les figures 6.11, 6.12 et 6.13 présentent les différentes étapes nécessaires à la création d’un conseil par l’expert : imaginons un expert devant proposer un conseil aux utilisateurs accédant à une page clé (page présentant des informations importantes et orientant l’utilisateur). Il peut :

1. Rechercher, dans la liste des conseils génériques, un conseil proche de celui qu’il veut proposer aux utilisateurs.
2. Rechercher, parmi les instances du conseil générique choisi, celle qui lui paraît être proche du conseil qu’il veut proposer aux utilisateurs.
- 3.1. Si une instance convient, adapter l’instance, *i.e.*, modifier les instances de conseils élémentaires qui la composent.
- 3.2. Si aucune instance ne convient, créer une instance par défaut du conseil générique puis adapter cette instance par défaut, *i.e.*, modifier les instances de conseils élémentaires qui la composent.

1 – Choix, sélection d'un conseil générique puis création d'une nouvelle instance

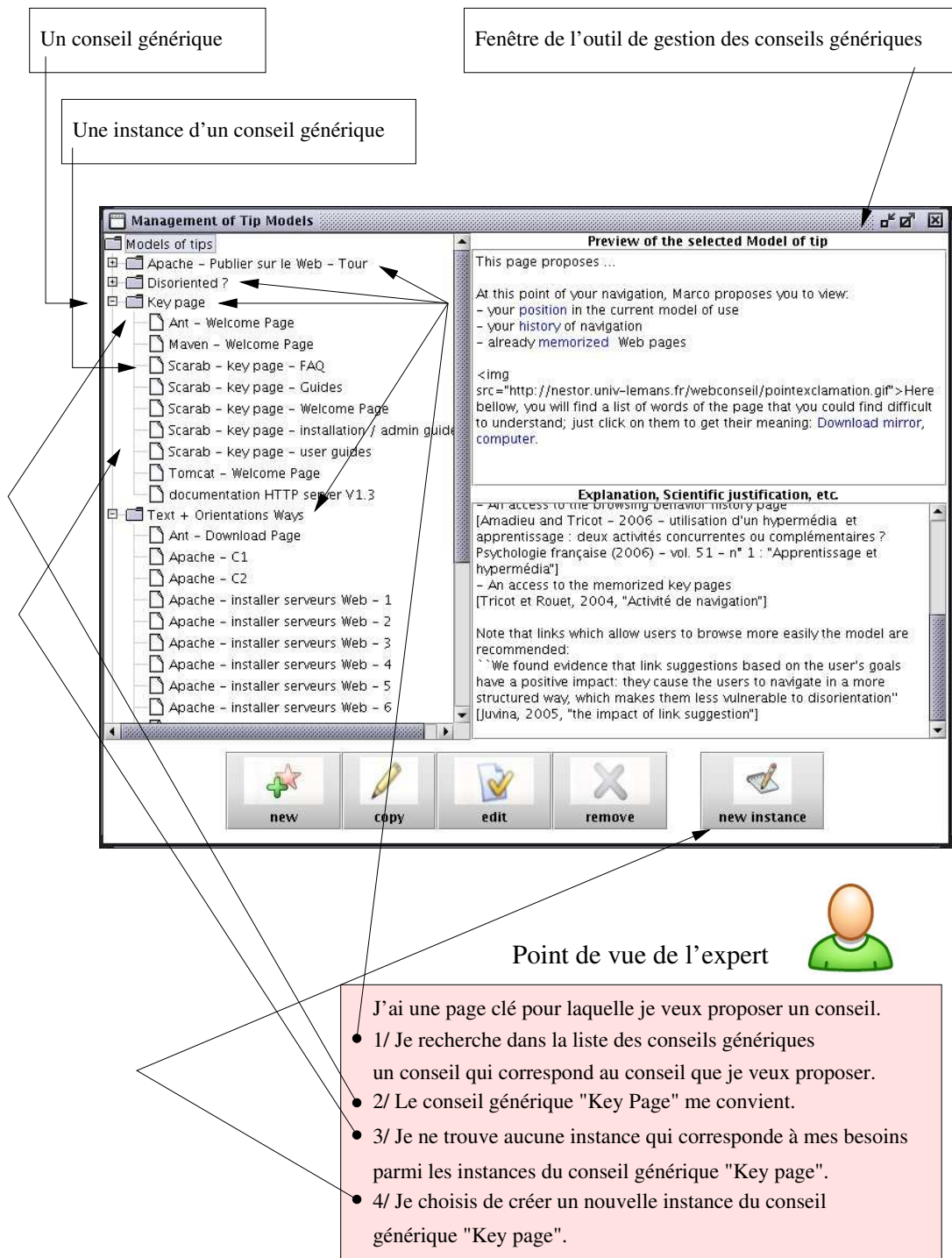


FIG. 6.11 – Créer un conseil - 1

2 – Adaptation de l'instance du conseil générique "Key page".

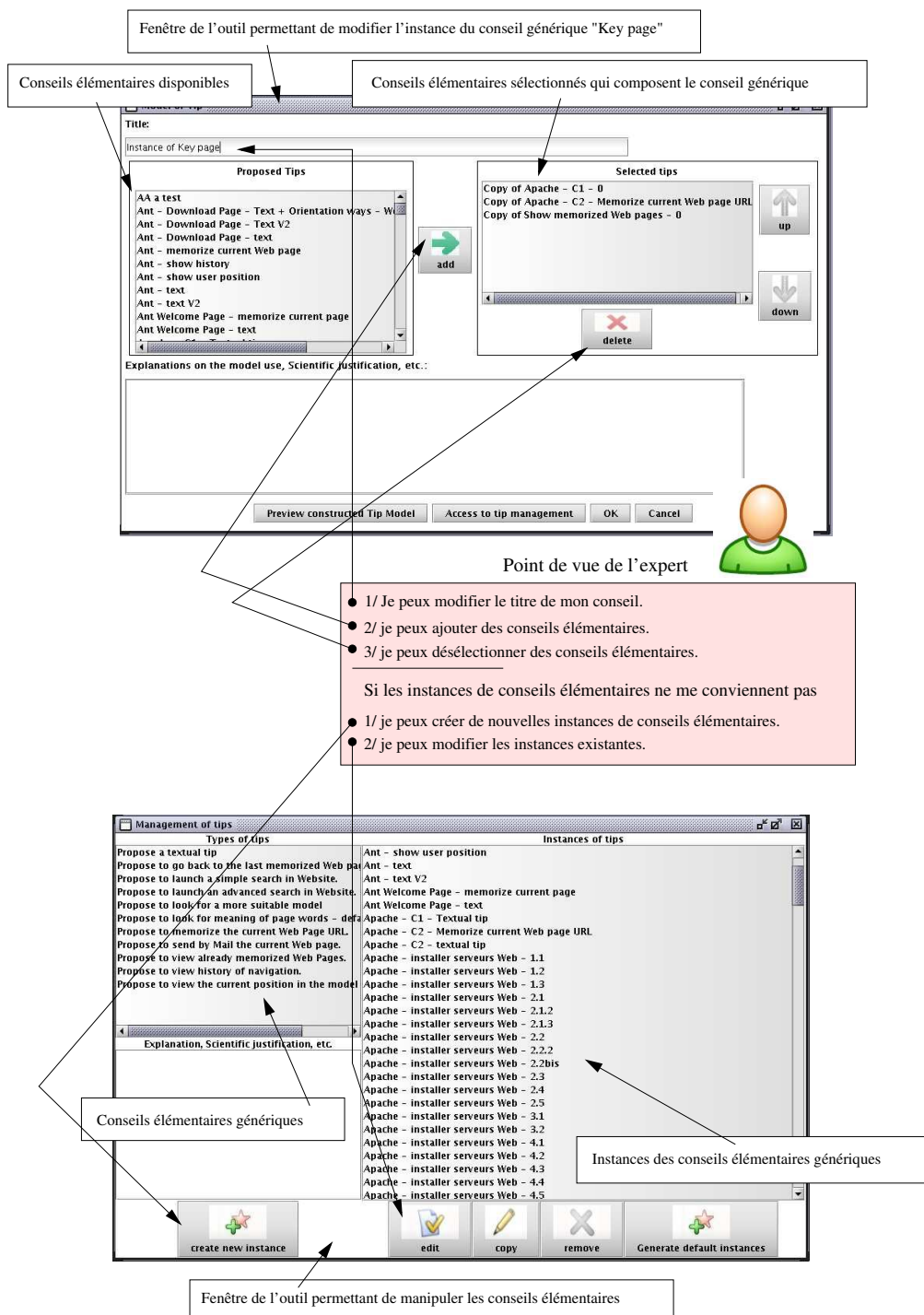
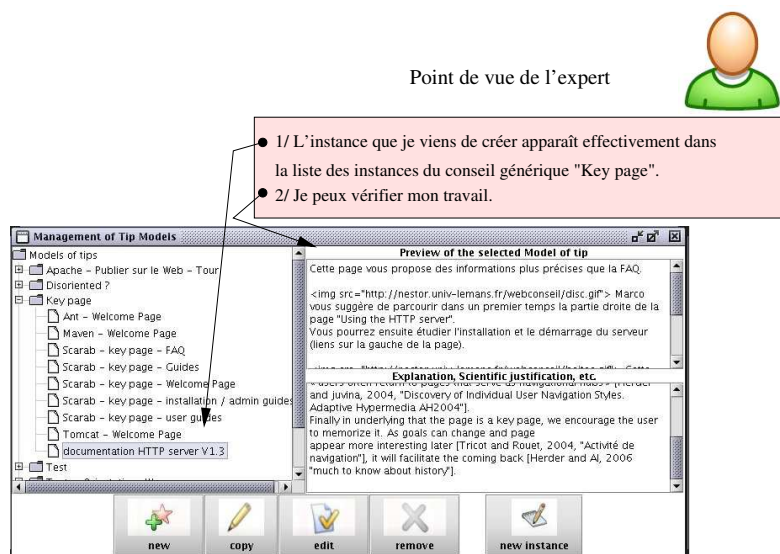


FIG. 6.12 – Créer un conseil - 2

3 – Conseils génériques et leurs instances



4 – Le conseil tel qu'il sera présenté à l'utilisateur

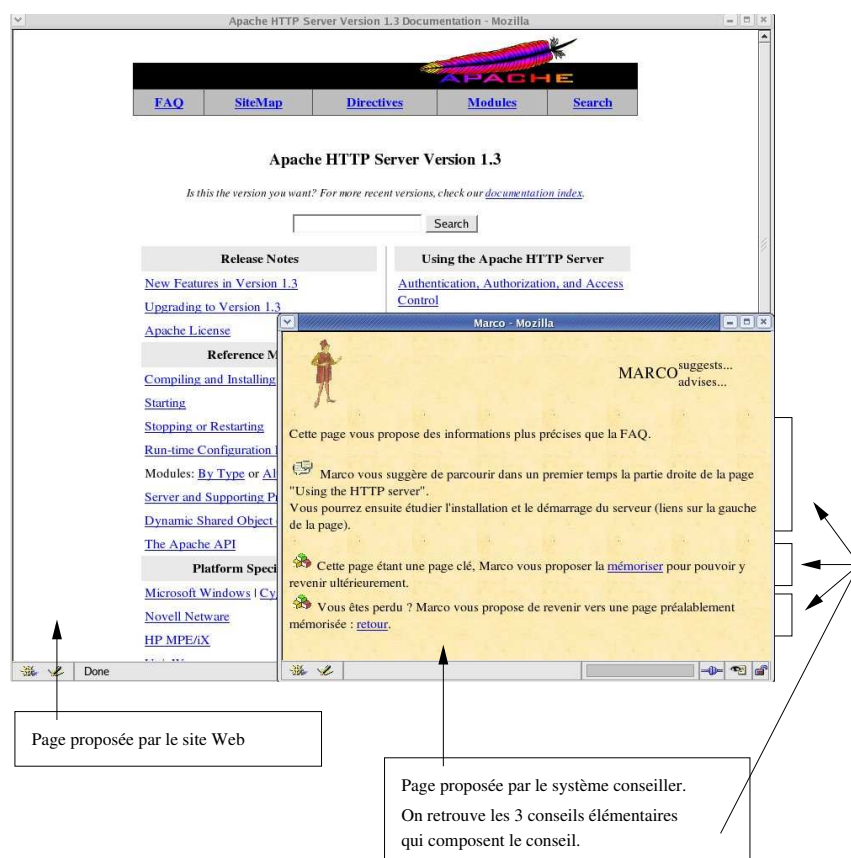


FIG. 6.13 – Créer un conseil - 3

6.3 Assister le positionnement des instances de conseil sur les arcs du modèle de tâches (étape 3)

Une fois le modèle de tâches et les instances de conseil construits, l'expert peut associer à chaque arc du modèle zéro ou une instance de conseil. Ces associations permettent de construire le modèle d'utilisation (cf. paragraphe 4.1.1). Différents types d'outils informatiques peuvent permettre ces associations. Comme pour les tâches précédentes, nous proposons deux exemples d'approches qu'il est possible d'envisager pour réaliser cette tâche.

6.3.1 Approche 1 - Utilisation de Protégé 2000

Principe. Une classe, nommée "action", a été créée. Elle modélise l'association d'un conseil et d'un arc (modélisé lui-même par une classe définissant le noeud de départ et le noeud d'arrivée). Deux autres champs de la classe "action" permettent de la préciser en donnant un numéro à l'action et un commentaire. La figure 6.14 présente la définition de la classe "action" avec Protégé 2000.

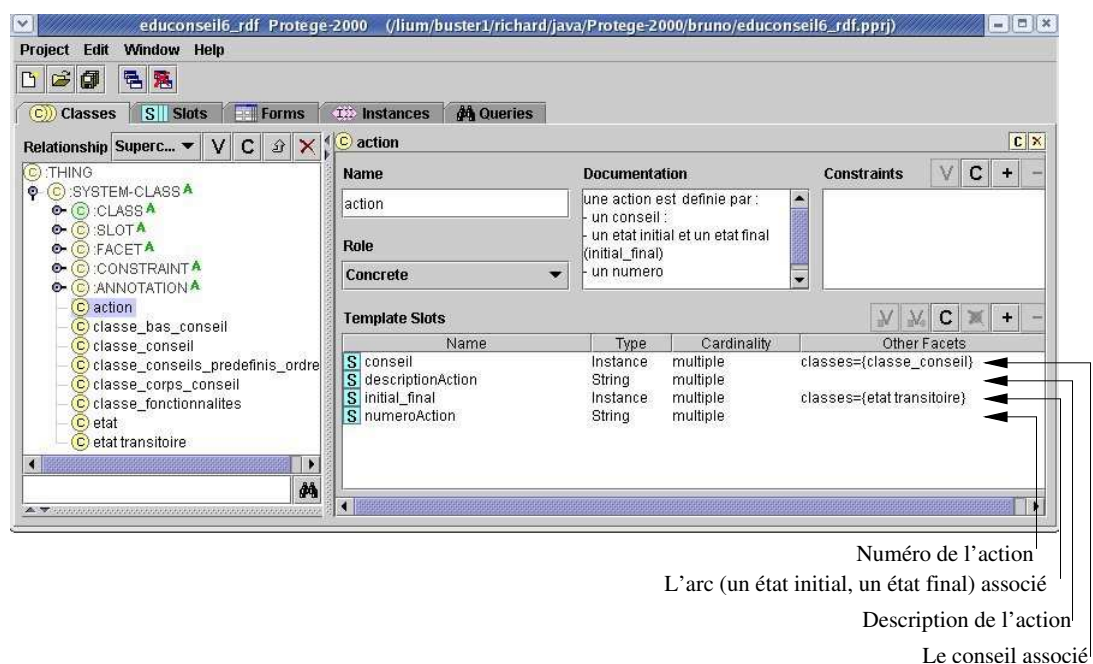


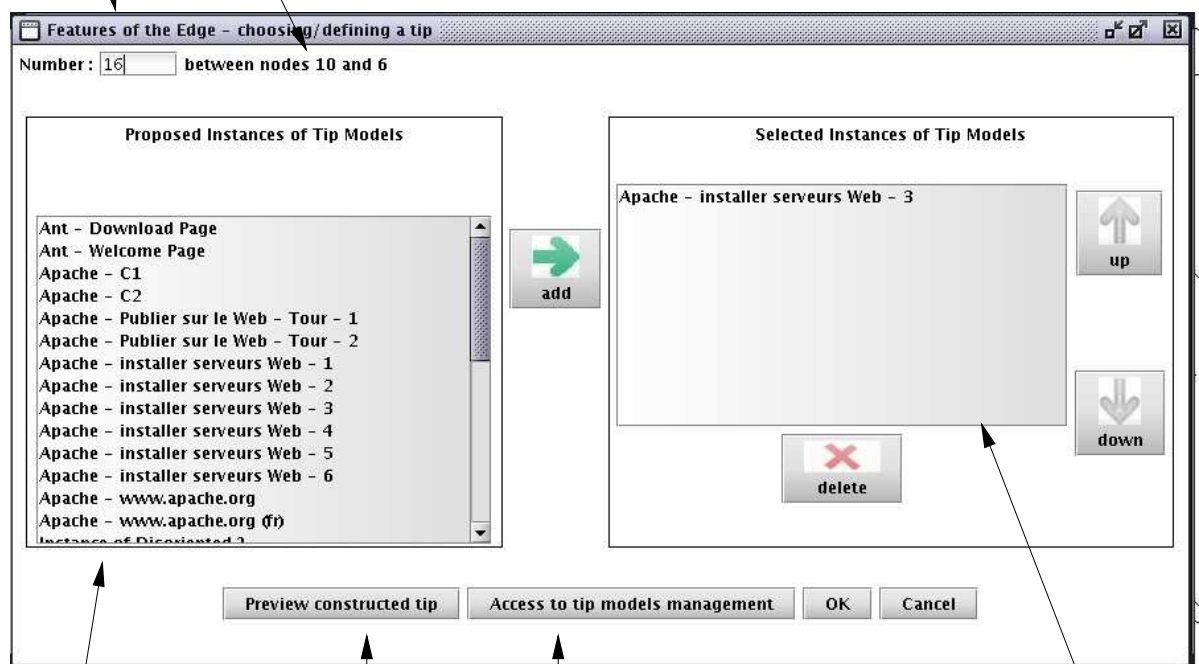
FIG. 6.14 – Positionnement d'un conseil sur un arc avec Protégé 2000

Limites. Les limites de cette approche viennent de son utilisabilité. Si lier un conseil avec un arc s’effectue simplement et rapidement, la tâche se complique dès que le nombre de noeuds, d’arcs et de conseils augmente. Dès lors, une version avec une interface graphique devient pertinente.

6.3.2 Approche 2 - Environnement ad hoc

Cette deuxième approche propose à l’expert une interface graphique lui présentant les noeuds et les arcs les reliant. La copie d’écran présentée figure 6.4 (page 127) propose un aperçu d’un modèle d’utilisation qu’il est possible de créer avec l’environnement ad hoc. On y observe des noeuds et des arcs. La figure 6.15 présente l’outil permettant d’associer un conseil à un arc par simple double click sur celui-ci.

Numéro de l’arc et rappel de son positionnement



Instances de conseils génériques
(pouvant être associées à un arc)

Prévisualisation du conseil

Accès à l’outil de gestion des conseils

Instances sélectionnées

FIG. 6.15 – Positionnement d’un conseil sur un arc avec l’environnement ad hoc

6.4 Assister l’instanciation d’un système conseiller générique (étape 5)

L’instanciation d’un système conseiller générique consiste en la prise en compte des modèles d’utilisation par le système conseiller (cf. paragraphe 4.2.5). Deux exemples d’approches qu’il est possible de proposer sont présentées ci-dessous.

6.4.1 Approche 1 - Utilisation de Protégé 2000

Comme nous l’avons déjà expliqué (cf. paragraphe 6.1.2) Protégé 2000 permet d’exporter les modèles construits au format RDFS. Ce format est un standard. Il était donc intéressant d’intégrer au système conseiller un module d’analyse du format RDFS lui permettant ainsi d’importer des modèles au format RDFS élaborés avec Protégé 2000 ou avec un autre outil proposant une exportation au format RDFS.

L’instanciation d’un système conseiller générique, lorsque les modèles sont construits avec Protégé 2000, s’effectue comme suit :

1. Exportation au format RDFS des modèles prêts à être utilisés par les utilisateurs.
2. Importation, c’est-à-dire lecture et analyse de ces modèles par le système conseiller.

6.4.2 Approche 2 - Environnement ad hoc

Pour que les modèles puissent être présentés aux utilisateurs, ils doivent être exportés sur le système conseiller. L’environnement ad hoc assiste cette tâche en proposant de :

- Lister, supprimer les modèles déjà exportés.
- Lister, supprimer les modèles créés avec l’environnement ad hoc.
- Exporter sur le système conseiller les modèles créés dans l’environnement ad hoc.

6.4.3 Définition des propriétés des modèles

Indépendamment de l’approche adoptée, il peut être intéressant d’associer aux modèles des informations permettant à l’utilisateur de faire son choix. En effet,

lorsqu'ils accèdent au système conseiller, les utilisateurs doivent choisir parmi les modèles d'utilisation celui qui semble le plus en rapport avec le but qu'ils veulent réaliser en accédant au site. Ce choix est important puisqu'il va conditionner la suite du processus : si l'utilisateur choisit un modèle qui ne correspond pas à ses besoins (*i.e.*, le but que le modèle permet de réaliser n'est pas celui de l'utilisateur), les conseils proposés ne seront pas adaptés.

Dans l'environnement ad hoc, nous avons donné à l'expert la possibilité de définir ces informations. Ainsi, l'expert peut renseigner, pour chaque modèle, des informations telles que :

- Le(s) but(s) que le modèle permet d'atteindre.
- La taille du modèle.
- Les caractéristiques du site Web cible : description générale, structure (hiérarchique, en réseau, etc.), fonctionnalités qu'il est nécessaire de savoir manipuler.

La figure 6.16 propose un extrait de la page de garde d'un système conseiller où les éléments de description permettant à l'utilisateur de faire son choix sont présentés.

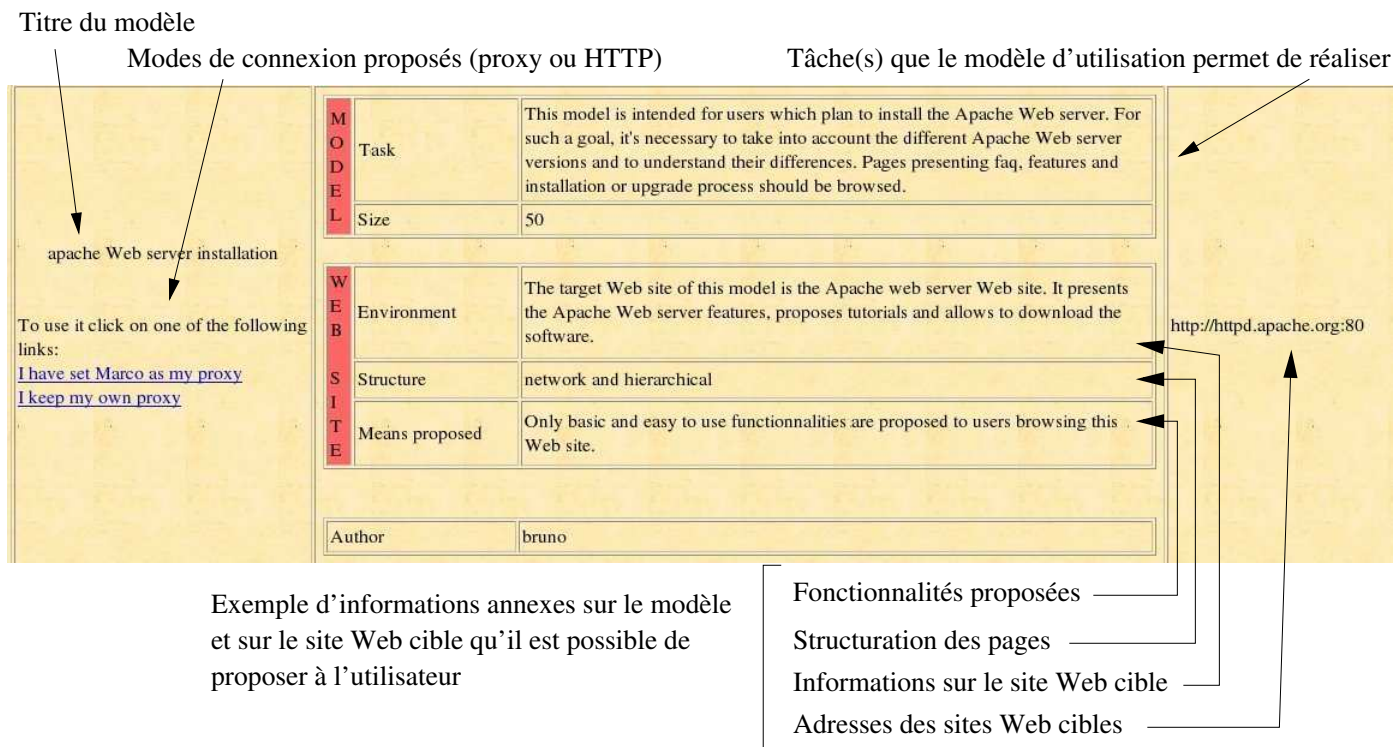


FIG. 6.16 – Extrait de la page de garde du système conseiller

6.5 Synthèse

Élaborer un système conseiller requiert plusieurs étapes. Les étapes de création des modèles et des conseils sont des tâches difficiles qui peuvent cependant être assistées et facilitées par l'outil informatique. Nous avons proposé deux approches illustrant l'aide qu'il est possible d'apporter à l'expert en charge de ces tâches.

La première approche est fondée sur Protégé 2000 et sur l'exportation au format RDFS des modèles. Cette approche offre un cadre de travail relativement rigide à l'expert et devient de plus en plus complexe à utiliser au fur et à mesure que la taille du modèle augmente (ce qui est rapidement le cas dès lors qu'on s'intéresse à des sites Web complexes).

La seconde approche est fondée sur un environnement ad hoc que nous avons développé pour explorer les différentes possibilités d'assistance de l'expert dans la tâche de construction de systèmes conseillers. Cet environnement graphique assiste la création de modèles de tâches et le positionnement des conseils sur les arcs. Des mécanismes d'objets et d'instances de ces objets ont été mis en place pour éviter la réécriture de conseils et encourager la création d'un ensemble cohérent de conseils.

Chapitre 7

Mise en oeuvre

Nous avons détaillé dans les chapitres précédents les développements que nous avons effectués tout au long de ce travail de thèse. Le but de ces développements était d'illustrer la faisabilité de notre approche. Deux orientations ont été proposées : l'une fondée sur Protégé 2000 et la construction de modèles au format RDFS, l'autre fondée sur des outils et des modèles ad hoc, le système conseiller évoluant en fonction de ces approches.

Pour illustrer la faisabilité de chacune de ces orientations et tester le fonctionnement de notre architecture, nous avons construit différents modèles et fait fonctionner les différentes versions de l'architecture. Ce sont ces modèles et des copies d'écran illustrant leur mise en oeuvre que nous présentons dans ce chapitre.

Les différentes étapes pour construire un système conseiller sont (cf. paragraphe 4.2) :

0. Création/adaptation/modification d'un système conseiller générique.

1. Identification et création des modèles de tâches.

- (a) Analyse du site (structure et usages).
- (b) Identification des tâches.
- (c) Création des modèles de tâches.

2. Création des conseils.

3. Positionnement des conseils sur les modèles de tâches et création des modèles d'utilisation.

4. Instanciation du système conseiller générique : installation des modèles d'utilisation, adaptation réseau.

5. Analyse du fonctionnement du couple système conseiller - site cible : analyse des usages courants et si besoin (en cas, par exemple, de découverte d'usages non prévus et qui ne correspondent pas aux attentes des concepteurs du site cible) nouvelle itération à partir de l'étape 0 ou 1.

L'architecture Marco répond à l'étape 0. Nous allons nous intéresser plus particulièrement aux étapes 1 à 4, l'étape 5 étant hors du cadre de cette thèse.

7.1 Premier terrain applicatif : Educasource

Nous avons abordé ce terrain applicatif dans le cadre d'un contrat du L.I.U.M. avec l'un des services du ministère français de l'éducation et de la recherche (P. Tchounikine était le porteur de ce projet). Il s'agissait d'étudier la faisabilité de l'association d'un système conseiller au site Web Educasource.

7.1.1 Présentation du site

Educasource [Edu] est un site du ministère français de l'éducation et de la recherche. Nous présentons figure 7.1 sa page d'accueil. Ce site est à destination des enseignants. Il propose un ensemble de descripteurs de ressources mises à disposition des enseignants. Chaque descripteur présente les informations suivantes :

- Auteur,
- Résumé,
- Domaine,
- Mots clés,
- Lien vers la ressource.

La figure 7.2 présente un exemple de descripteur.

Deux modes de navigation permettent d'accéder à ces descripteurs :

- Navigation via les moteurs de recherche. Deux moteurs de recherche sont proposés. Ils permettent :
 1. Une recherche simple : l'utilisateur fournit quelques mots clés
 2. Une recherche avancée : l'utilisateur fournit
 - Des mots clés. Ces mots clés peuvent être recherchés selon différents modes : "au moins un mot", "tous les mots" ou "l'expression exacte"

Moteur de recherche (recherche simple ou avancée)

Forum



Arborescence thématique

FIG. 7.1 – Page d'accueil du site Educasource

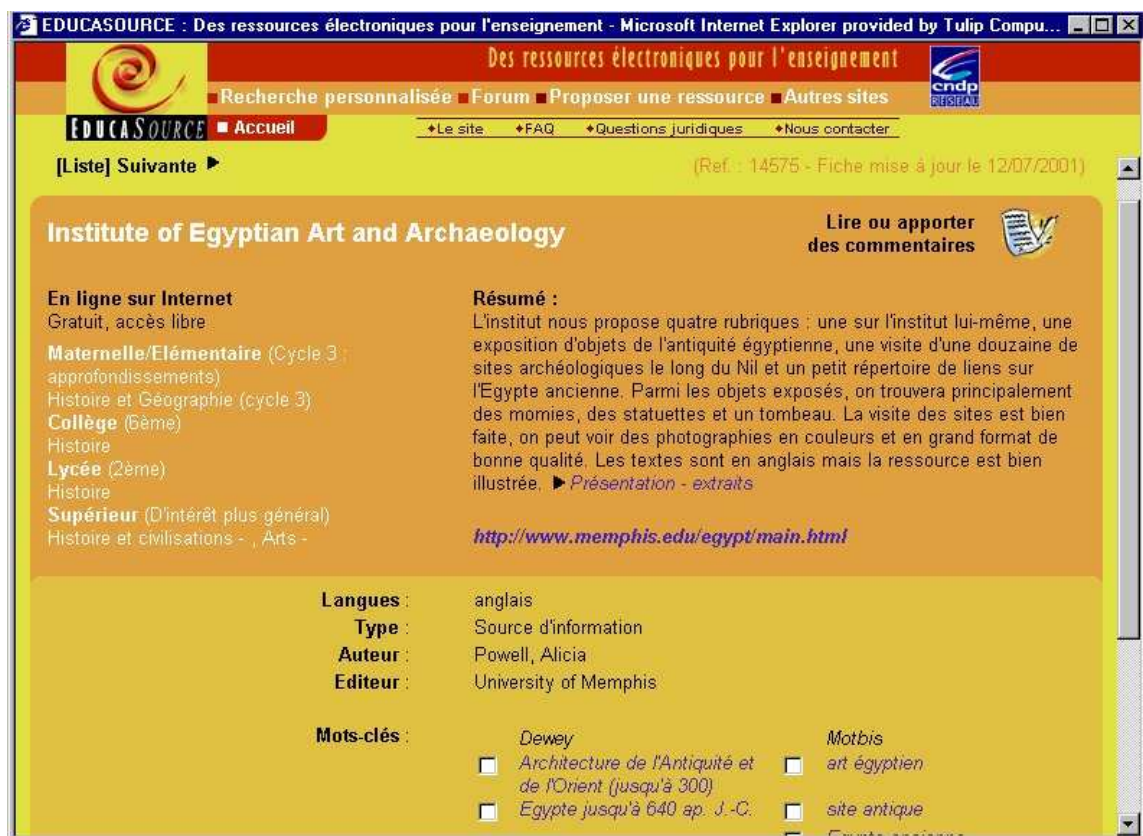


FIG. 7.2 – Exemple de descripteur proposé par le site Educasource

- Des informations précisant le but de sa recherche : ressources pédagogiques souhaitées, sources d'informations envisagées, degré, niveau du public visé, discipline, type de support ou encore langue visée.

A l'issue de la recherche, classiquement, une liste de pointeurs (URLs) vers les descripteurs trouvés par le moteur de recherche est proposée.

- Navigation via la taxonomie. Les descripteurs sont classés et présentés en fonction de la taxonomie de Dewey. Il est possible pour l'utilisateur de parcourir cette taxonomie jusqu'à atteindre le descripteur recherché.

Enfin, un forum est proposé aux utilisateurs. Organisé par thèmes, il permet aux utilisateurs d'échanger des informations en fonction de ces thèmes. De nouveaux thèmes peuvent être proposés par les utilisateurs.

Notons que le site Educasource n'existe plus sous cette forme, les objectifs des concepteurs ayant changé radicalement pour des raisons sans rapport avec le travail présenté ici. Le prototype que nous avons créé n'a donc pas été exploité.

7.1.2 Analyse du site (étape 1.a)

Après une année d'utilisation, les usages du site ont été analysés par un cabinet de conseil indépendant. Les résultats de cette analyse nous ont été fournis et ont formé la base de notre travail. Cette analyse a principalement relevé deux dysfonctionnements (par rapport aux attentes des concepteurs) :

- Sur-utilisation des moteurs de recherche. Il apparaît une sur-utilisation des moteurs de recherche par rapport à la navigation via la taxonomie. Or, comme nous l'avons déjà indiqué dans l'introduction, l'utilisation de moteurs de recherche présente des inconvénients : d'une part, définir les mots clés représentant l'objet de la recherche n'est pas facile (la difficulté augmentant lorsque l'utilisateur a peu de connaissances dans le domaine de l'objet recherché et/ou lorsque la recherche est complexe), d'autre part, les moteurs de recherche proposent souvent, comme résultats de la recherche, des listes qui sont trop longues ou trop courtes, laissant à l'utilisateur soit la charge d'extraire les items qui pourraient lui convenir, soit la nécessité de reformuler sa requête.

Enfin, la sur-utilisation des moteurs de recherche entraîne une sous-utilisation de la taxonomie qui a pourtant des atouts intéressants, notamment le fait qu'elle facilite la sérendipité.

- Sous-utilisation du forum. Peu de messages sont déposés dans le forum alors que celui-ci est destiné à être un lieu de vie facilitant les échanges entre utilisateurs du site.

L'objet de notre travail exploratoire a consisté à étudier comment aborder ces dysfonctionnements par l'ajout au site d'un système conseiller. L'architecture que nous avons proposée est celle décrite dans les paragraphes précédents, à savoir un système conseiller épiphyte dont le fonctionnement est fondé sur des modèles d'utilisation. Les modèles créés visent deux objectifs :

- Au niveau des fonctionnalités de recherche, (1) encourager la recherche via la taxonomie et (2) assister les utilisateurs qui choisissent d'utiliser les moteurs de recherche (en leur proposant, par exemple, une aide pour étendre leur recherche ou au contraire une aide pour trier, structurer les listes de descripteurs de ressources obtenues).
- Au niveau des fonctionnalités de communication entre les utilisateurs du site, encourager l'utilisation du forum en proposant des fonctionnalités incitatives pouvant amener les utilisateurs à communiquer entre eux.

Ces deux objectifs illustrent le fait qu'un système conseiller et des modèles d'utilisation peuvent viser plusieurs objectifs. L'utilisation des modèles permet, d'une part, d'aplanir certaines difficultés rencontrées par les utilisateurs (par exemple, en conseillant les utilisateurs sur des pages jugées difficiles ou en expliquant certaines fonctionnalités). Elle permet, d'autre part, de favoriser certains comportements ou de tenter d'en éviter d'autres en fonction de ce que les concepteurs jugent opportuns ou contraires à leurs attentes.

7.1.3 Identification des tâches (étape 1.b)

Les tâches à assister sont, dans ce cas, directement données par l'énoncé du problème :

- Utilisation libre du site. Il s'agit d'assister un utilisateur n'ayant pas décidé, *a priori*, quel mode de recherche il allait choisir. La recherche via la taxonomie est encouragée. La recherche via les moteurs de recherche est assistée. Enfin, des conseils favorisant la communication entre utilisateurs (comportement recherché par les concepteurs) sont proposés, ainsi que des conseils favorisant l'ouverture des utilisateurs vers d'autres sites jugés intéressants par les concepteurs du site Educasource.

- Utilisation des moteurs de recherche. Des outils sont proposés pour faciliter leur utilisation. Comme pour la tâche précédente, des conseils favorisant la communication entre utilisateurs sont proposés, ainsi que des conseils favorisant l'ouverture des utilisateurs vers d'autres sites.
- Utilisation du forum. Des outils sont proposés pour encourager et faciliter l'utilisation du forum.

7.1.4 Création des modèles de tâches (étape 1.c)

Pour chacune des tâches décrites ci-dessus des modèles de tâches ont été construits. La figure 7.3 présente une représentation sous forme de graphe du modèle de tâches correspondant à la tâche "Utilisation des moteurs de recherche". Ces modèles ont été construits "à la main", par analyse du site, et en utilisant l'approche fondée sur Protégé 2000.

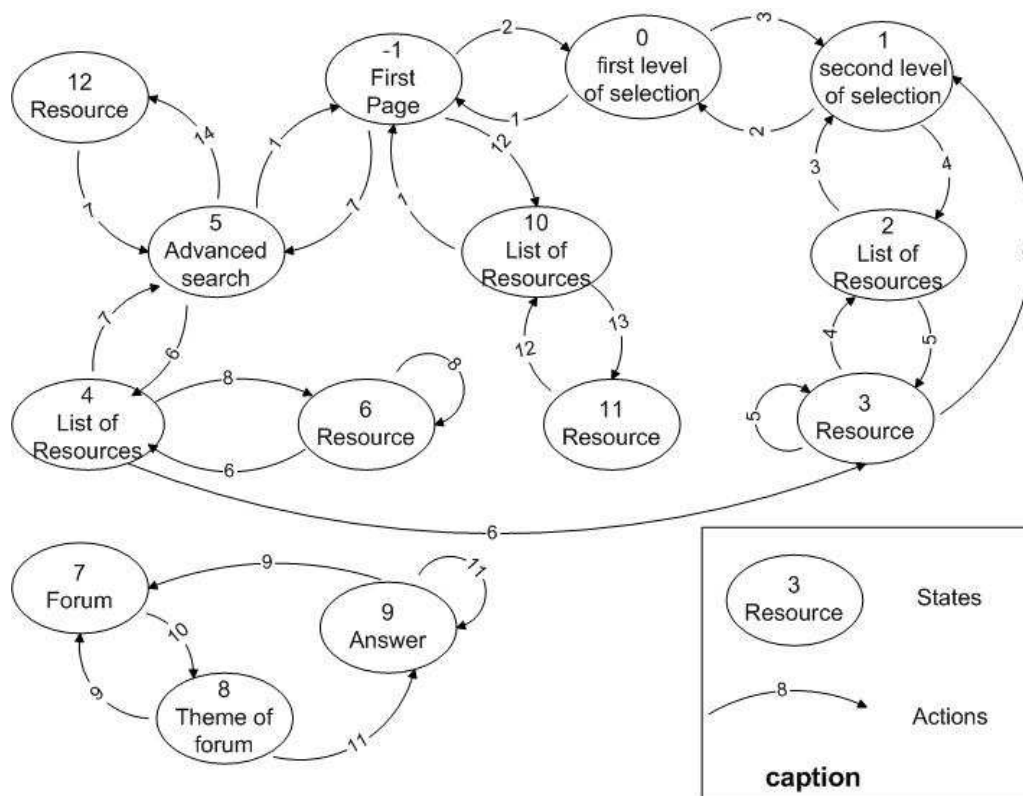


FIG. 7.3 – Le modèle de tâches construit pour le site Educasource et correspondant à la tâche "Utilisation des moteurs de recherche".

7.1.5 Création des conseils (étape 2)

Différents conseils intégrant textes, liens et fonctionnalités ont été élaborés. Ils correspondent aux conseils élémentaires présentés dans le chapitre précédent. Des exemples de ces conseils sont :

- Aides à la navigation :
 - Présentation de la position de l'utilisateur dans la taxonomie.
 - Offre de mémorisation des liens vers des descripteurs susceptibles d'être intéressants. La consultation peut être effectuée plus tard, laissant l'utilisateur se concentrer sur ses opérations de recherche.
 - Rappel des choix effectués par l'utilisateur au cours de sa navigation et possibilité de retour à un point de décision.
 - Rappel tout au long de la navigation des différentes fonctionnalités à disposition (fonctionnalités offertes par le site cible et non accessibles sur la page courante ou fonctionnalités offertes par le système conseiller).
 - Offre de mémorisation du descripteur courant. L'utilisateur peut ainsi se constituer une base de descripteurs qui pointent vers des ressources qu'il juge intéressantes.
 - Offre de mémorisation de la position courante dans la taxonomie. Il est ainsi possible de proposer à l'utilisateur de revenir vers ces positions.
- Aides à la recherche :
 - Extension automatique des listes résultats de recherche lorsque celles ci sont trop courtes (le système conseiller génère automatiquement des recherches supplémentaires à partir des résultats de la recherche initiale et en présente les résultats).
 - Restructuration automatique (par exemple, par domaine ou par mot clé) des listes résultats de recherche lorsque celles ci sont trop longues.
 - Extension des recherches vers des parties du site cible non encore visitées.
 - Extension des recherches vers des sites jugés intéressants par les concepteurs du site cible.
 - Proposition de compléments d'information. Par exemple, affichage du nombre de descripteurs qu'une branche de la taxonomie permet d'atteindre.

- Promotion du dialogue entre utilisateurs :
 - Recherche automatique dans le forum des messages susceptibles d'intéresser l'utilisateur (le système conseiller recherche automatiquement dans le forum des messages qui pourraient intéresser l'utilisateur et propose à l'utilisateur des liens vers ces messages).
 - Offre de création et de gestion de paniers de ressources : les ressources jugées intéressantes par un utilisateur peuvent être regroupées en un panier qu'il est ensuite possible de décrire puis de partager avec les autres utilisateurs.
 - Recherche automatique de paniers de ressources susceptibles d'intéresser l'utilisateur.
 - Proposition d'un espace d'échanges présentant les différents paniers créés et partagés par les utilisateurs du site.
- etc.

Ces conseils ont été créés à l'aide de Protégé 2000 dont l'utilisation a été décrite au chapitre précédent.

Les figures 7.4 et 7.5 présentent des exemples de conseil dont peuvent bénéficier les utilisateurs. On y retrouve quelques uns des conseils et des fonctionnalités présentés ci-dessus.

7.1.6 Positionnement des conseils sur les modèles de tâches (étape 3)

Les conseils ont été positionnés sur les arcs des modèles de tâches. Ce mécanisme a été présenté au chapitre précédent et illustré par la figure 6.14 (cf. page 142).

7.1.7 Instanciation d'un système conseiller générique (étape 4)

Comme nous l'avons déjà expliqué, avec Protégé 2000 l'instanciation d'un système conseiller générique consiste à (1) exporter au format RDFS des modèles d'utilisation puis (2) importer ces modèles dans le système conseiller.

Dès lors, les modèles sont présentés aux utilisateurs sur la page de garde du système conseiller (la figure 6.16, présentée au chapitre précédent (page 146), proposait un exemple de page de garde).



FIG. 7.4 – Exemple de conseil proposé par le système conseiller

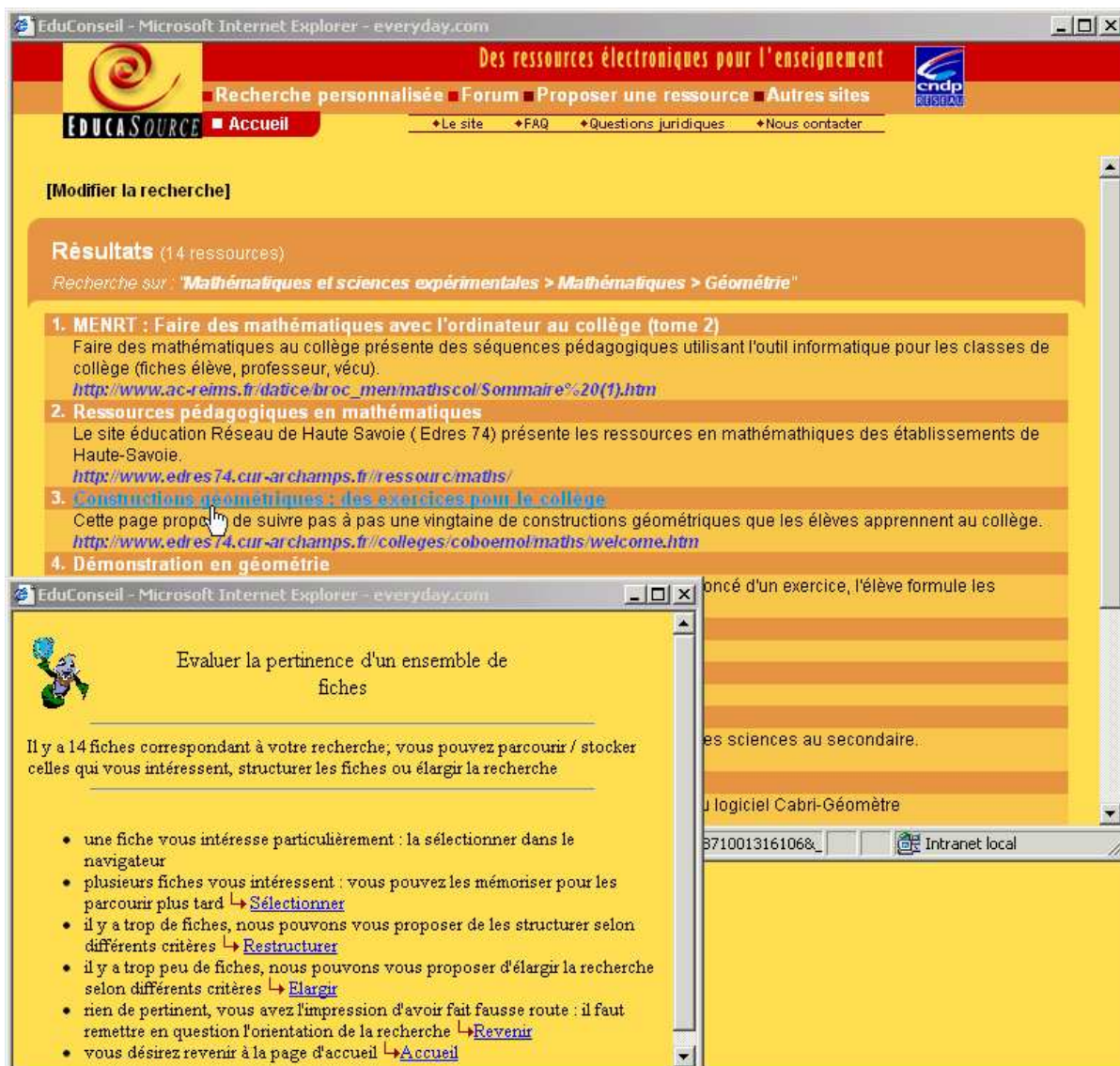


FIG. 7.5 – Exemple de conseil proposé par le système conseiller

7.2 Second terrain applicatif : Apache

7.2.1 Présentation du site

Le site de la fondation Apache est un site extrêmement complexe : il comporte de nombreuses pages (en mai 2006, il présentait près de 300 000 pages), présente une structure large (le site est organisé en sous-projets et la cinquantaine de projets qui sont représentés, forme une arborescence large) et profonde et de nombreuses fonctionnalités (moteur de recherche, forum, listes de diffusion, accès sécurisé, gestion des versions, etc.). Le site est conçu autour d'un site Web central dont l'URL est <http://www.apache.org>. Ce site propose des pointeurs vers les projets hébergés par la fondation Apache. Ces projets sont nombreux (une cinquantaine) et variés : on trouve ainsi des projets tels que HTTPD (serveur Web), SpamAssassin (lutte contre le SPAM) ou Tomcat (conteneur de servlet J2EE). Ces projets sont fondés sur une approche visant à favoriser le travail collaboratif et le développement de logiciels libres et open source. Nous avons choisi ce site pour mettre à l'essai la faisabilité de notre approche sur un site de grande envergure et pour lequel nous avons pu, sur la base d'une convention avec la fondation Apache, disposer des logs.

7.2.2 Identification et création des modèles de tâches (étape 1)

7.2.2.1 Analyse du site (étape 1.a)

Analyse de la structure du site. La structure du site de la fondation Apache est hiérarchique et cloisonnée : chaque projet développe un site Web propre avec une interface qui lui est propre. On n'observe que peu de liens transversaux entre les projets : le serveur principal (dont l'URL est <http://www.apache.org>) propose des liens vers chacun des projets, et les sites Web des projets proposent eux aussi un retour vers le site principal. Cependant, peu de projets proposent des liens vers d'autres projets. Les projets qui proposent de tels liens sont souvent des projets qui ont éclaté en plusieurs sous projets (par exemple, Lenya qui fut un sous projet de Cocoon et qui est maintenant un projet à part entière). Pourtant, certains projets sont tournés vers un même objectif. Par exemple, HTTPD et Cocoon proposent chacun une approche particulière pour permettre la publication d'informations sur le Web.

Analyse des usages du site. Pour analyser les usages du site Apache, nous avons choisi d'étudier les logs du serveur, cette solution nous paraissant la plus

abordable par rapport à nos moyens et au temps imparti. Dans ce but, ainsi que nous l'avons indiqué au paragraphe 6.1.1, nous avons signé une convention avec la fondation Apache pour disposer des enregistrements de l'activité de leurs serveurs Web pendant une période suffisamment longue (16 mois) pour être représentative.

Comme nous l'avons expliqué ci-dessus, la structure du site est principalement verticale : les utilisateurs accèdent au site principal (dont l'URL est <http://www.apache.org>) puis "descendent" dans les sites des projets qui les intéressent. Il y a peu de possibilités pour un utilisateur de naviguer transversalement : un utilisateur intéressé par plusieurs projets trouvera peu de liens lui permettant de passer d'un projet à un autre. Cette caractéristique est d'ailleurs confirmée par l'analyse des logs : en 2005 seules 4,34% des sessions ont donné lieu à un parcours sur 3 sites ou plus de la fondation Apache (le détail des calculs présentés dans cette section est donné en annexe).

Pourtant, prenons l'exemple d'un utilisateur accédant au site Apache et cherchant à installer un système avec lequel il puisse publier des informations sur le Web. L'analyse du site de la fondation Apache montre que différentes possibilités lui sont offertes, dont les plus populaires sont :

- Installer HTTPD, le serveur Web de la fondation Apache
- Installer Cocoon, une plate-forme fondée sur Jakarta Tomcat et sur laquelle peuvent venir se greffer d'autres produits de la fondation Apache, par exemple, Forrest ou Lenya

Bien que ce type de navigation semble correspondre à la logique de l'offre d'Apache, de telles navigations sont peu fréquentes. Le schéma présenté figure 7.6 détaille les différentes navigations que l'on peut observer autour de ces projets. Deux points peuvent être soulignés :

1. Plus de la moitié (56%) des entrées sur le site de la fondation Apache se font via les sites `httpd` (19%), `www` (17%) et Jakarta (20%) ; ces données s'expliquent par le fait que `httpd` et `jakarta` sont les produits phares de la fondation Apache. Dans notre cadre de travail on observe que les utilisateurs intéressés par l'installation d'un système leur permettant de publier sur le Web se dirigent principalement vers HTTPD et très peu vers Cocoon.
2. Une fois sur le site de Cocoon, on observe peu de mouvements vers les sites Lenya et Forrest, pourtant directement liés.

Pour tenter de mieux comprendre ces navigations, nous les avons extraites puis analysées. Pour mener cette analyse, nous nous sommes fondés sur des travaux [Her02], [Her03] [HJ04] [HvD04] [HWOM06], [JvO05], [JH05] et [JO06]

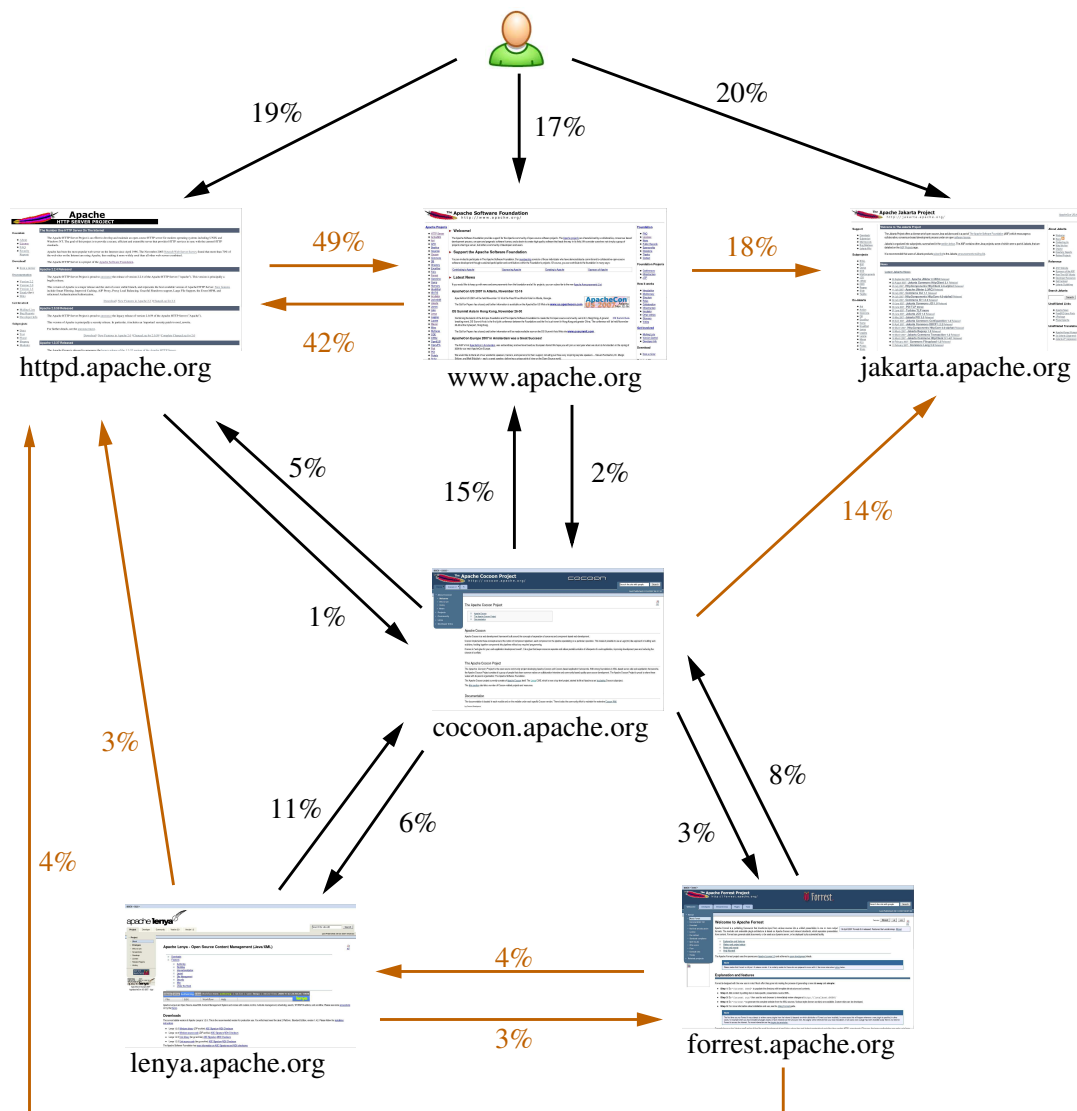


FIG. 7.6 – Navigations entre les projets WWW, HTTPD, Jakarta, Cocoon, Lenya et Forrest

qui ont montré qu’il était possible, à partir des logs d’un serveur Web, de mettre en évidence des navigations caractéristiques et en particulier des navigations qualifiées de “flimsy navigation” caractérisant des utilisateurs qui semblent perdus dans le site.

De telles navigations ont les caractéristiques suivantes :

- Un nombre de pages visitées faible. Pour la prise en compte de ce paramètre,

il convient de le relativiser par rapport au nombre moyen de pages visitées par session.

- Une densité des pages visitées élevée. La densité des pages visitées est calculée par la formule [Rau92] :

$$\frac{\text{nombre de liens parcourus}}{(\text{nombre de pages parcourues}) * (\text{nombre de pages parcourues} - 1)}$$

- Un temps médian de lecture élevé. Le temps médian de lecture se calcule selon le principe suivant. Soit une liste de n pages lues par un utilisateur d'un site Web au cours d'une session. A chaque page est associé un temps de lecture. Une fois cette liste triée en fonction du temps de lecture, le temps médian de lecture est le temps de lecture associé à la page $n/2$.
- Une distance moyenne faible entre les pages parcourues et connectées. La distance moyenne d'un graphe est la moyenne des plus petites distances entre toutes les paires de sommets [BKM⁺00]. La distance moyenne entre les pages connectées d'un graphe est la moyenne des plus petites distances entre toutes les paires connectées de sommets.
- Un nombre de cycles faible. Le nombre de cycles est calculé par la formule :

$$\text{nombre de liens parcourus} - \text{nombre de pages visitees} + 1$$

- Un taux de consultation de la première page du site élevé. Ce taux est calculé par la formule :

$$\frac{\text{nombre d'accès à la première page du site}}{\text{nombre de pages visitees}}$$

Pour analyser les logs et rechercher les navigations du type “flimsy navigation”, nous avons développé un outil spécifique. La figure 7.7 présente un exemple de données que le programme produit.

En utilisant ces indicateurs et en analysant les logs de la fondation Apache de l'année 2005 et en particulier les sessions d'utilisateurs parcourant les sites httpd.apache.org et cocoon.apache.org, on observe un nombre élevé (plus de 20%) de sessions qu'il est possible de qualifier de “flimsy navigation”.

En conclusion, on observe que peu d'utilisateurs vont naviguer au cours d'une même session sur les sites httpd.apache.org, cocoon.apache.org, lenya.apache.org et forrest.apache.org. Pourtant ces sites qui proposent plusieurs solutions à un problème unique, publier sur le Web, devraient être plus fréquemment parcourus simultanément. L'analyse des logs des serveurs Web montre que les utilisateurs qui s'y risquent rencontrent des problèmes puisqu'un nombre élevé de navigations présente les caractéristiques de navigations d'utilisateurs perdus.

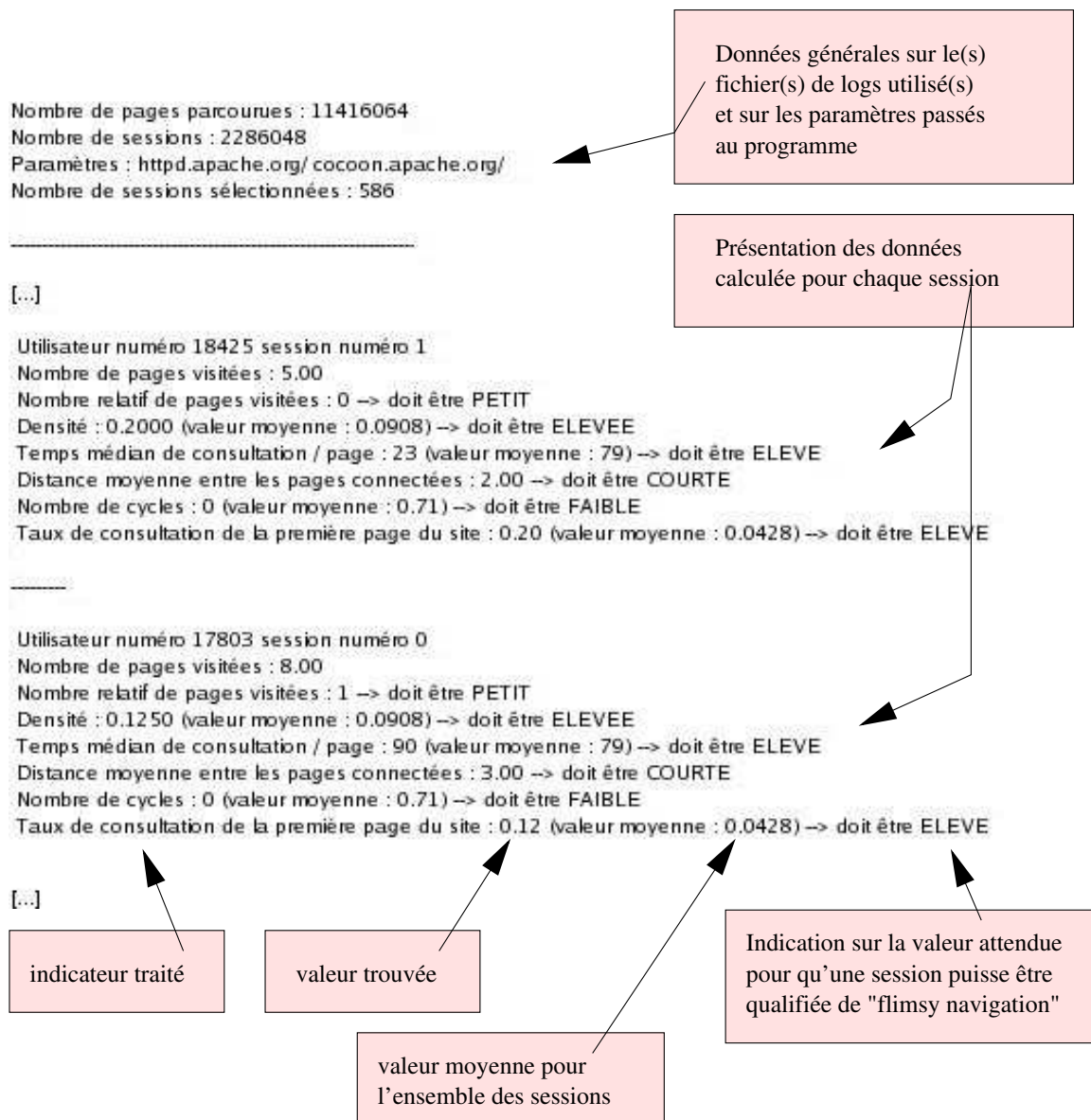


FIG. 7.7 – Sortie proposée par l’outil de recherche des navigations du type “flimsy navigation”

7.2.2.2 Identification des tâches (étape 1.b)

C’est sur la base de cette analyse que nous avons travaillé. Cette analyse est superficielle mais elle nous a permis de définir un problème “théorique” pertinent pour nos travaux et nous permet donc d’en tester la faisabilité.

Nous proposons d'assister la tâche précédemment évoquée : assister l'utilisateur voulant publier des informations sur le Web mais ne sachant quel outil utiliser. Comme nous l'avons déjà dit, la fondation Apache propose plusieurs outils : HTTPD ou Cocoon associé à Lenya ou à Forrest. Il s'agit principalement de proposer à l'utilisateur des explications sur chacun de ces outils, de faciliter la navigation dans chaque site et d'offrir des passerelles entre ces sites.

7.2.2.3 Création des modèles de tâches (étape 1.c)

Le modèle de tâches décrit ci-dessus est présenté figure 7.8. Cette figure illustre le fait que la taille des modèles de tâches devient rapidement importante (il ne s'agit pourtant que d'une copie d'écran partielle). Il propose à des utilisateurs inexpérimentés une assistance dans la découverte des différents projets de la fondation Apache permettant de publier des informations sur le Web. C'est l'environnement ad hoc présenté au chapitre précédent qui a été utilisé pour sa création.

FIG. 7.8 – Copie d'écran partielle du modèle “Utilisation des outils de la fondation Apache pour publier sur le Web”

7.2.3 Création des conseils (étape 2)

Les conseils que nous avons définis proposent une aide à la compréhension des pages jugées difficiles, une aide à la navigation et une aide à la recherche.

- Aide à la compréhension des pages. Il s'agit, par exemple, de :
 - Souligner l'existence de liens particuliers jugés utiles
 - Proposer la définition de termes jugés difficiles
- Aide à la navigation. Il s'agit d'assister l'utilisateur dans le parcours de chacun des sites évoqués plus haut et de faciliter le passage d'un site à un autre. Par exemple, :
 - Présentation de la position de l'utilisateur dans le modèle
 - Rappel de l'existence de pages en rapport avec sa recherche
 - Guidage vers les pages clés des sites
 - Proposition de mémorisation de pages jugées intéressantes par l'utilisateur
 - Retour en cas de désorientation vers des pages préalablement jugées intéressantes et mémorisées
- Aide à la recherche. La recherche dans ces sites est très peu encouragée, voire même difficile. Par exemple :
 - httpd.apache.org propose un d'outil de recherche uniquement lors de la consultation de la documentation.
 - cocoon.apache.org, lenya.apache.org et forrest.apache.org proposent un outil de recherche fondé sur Google. Cependant, aucune option avancée de recherche n'est possible (ainsi, il n'est pas possible (à moins de connaître la syntaxe des requêtes de Google) de rechercher des pages ne contenant pas un terme précis ou des pages ayant un titre particulier ou encore écrites ou modifiées depuis une date précise). De plus, la recherche peut être un facteur de désorientation : la recherche s'effectue dans la page courante. Si plusieurs recherches successives sont lancées sans pour autant satisfaire l'utilisateur, ce dernier pourra éprouver des difficultés à retourner à la page à partir de laquelle il a lancé la recherche.

Nous proposons, pour faciliter la recherche, plusieurs fonctionnalités, par exemple :

- Restructuration des listes résultats trop longues
- Assistance lorsque les listes résultats sont trop courtes :
 - Aide à l'extension des termes utilisés (par exemple, en recherchant automatiquement des synonymes ou en proposant des définitions)
 - Extension de la recherche aux sites proches (par exemple, au site `cocoon.apache.org` quand une recherche infructueuse a été lancée dans le site `lenya.apache.org`)
- Retour vers la page précédant le lancement de la recherche en cas de recherche infructueuse ou de désorientation
- Retour vers des pages clés du site en cas de recherche infructueuse ou de désorientation

Les figures 7.9 et 7.10 présentent des exemples de conseils donnés aux utilisateurs du modèle décrit dans cette section.

7.2.4 Positionnement des conseils sur les modèles de tâches (étape 3)

Dans l'environnement ad hoc (ici l'environnement Marco), cette tâche est assistée : un simple click sur les arcs permet d'en définir les propriétés et notamment d'associer des conseils définis précédemment. La figure 6.15 présentée au chapitre précédent (page 143), illustre cette fonctionnalité.

7.2.5 Instanciation d'un système conseiller générique (étape 4)

Dans l'environnement ad hoc, l'instanciation d'un système conseiller générique s'effectue en deux étapes : (1) définition des propriétés du modèle et (2) exportation des modèles.

L'étape 1 consiste en la saisie de données précisant le modèle :

- Données concernant le modèle :
 - Nom du modèle : "Apache - publier sur le Web"

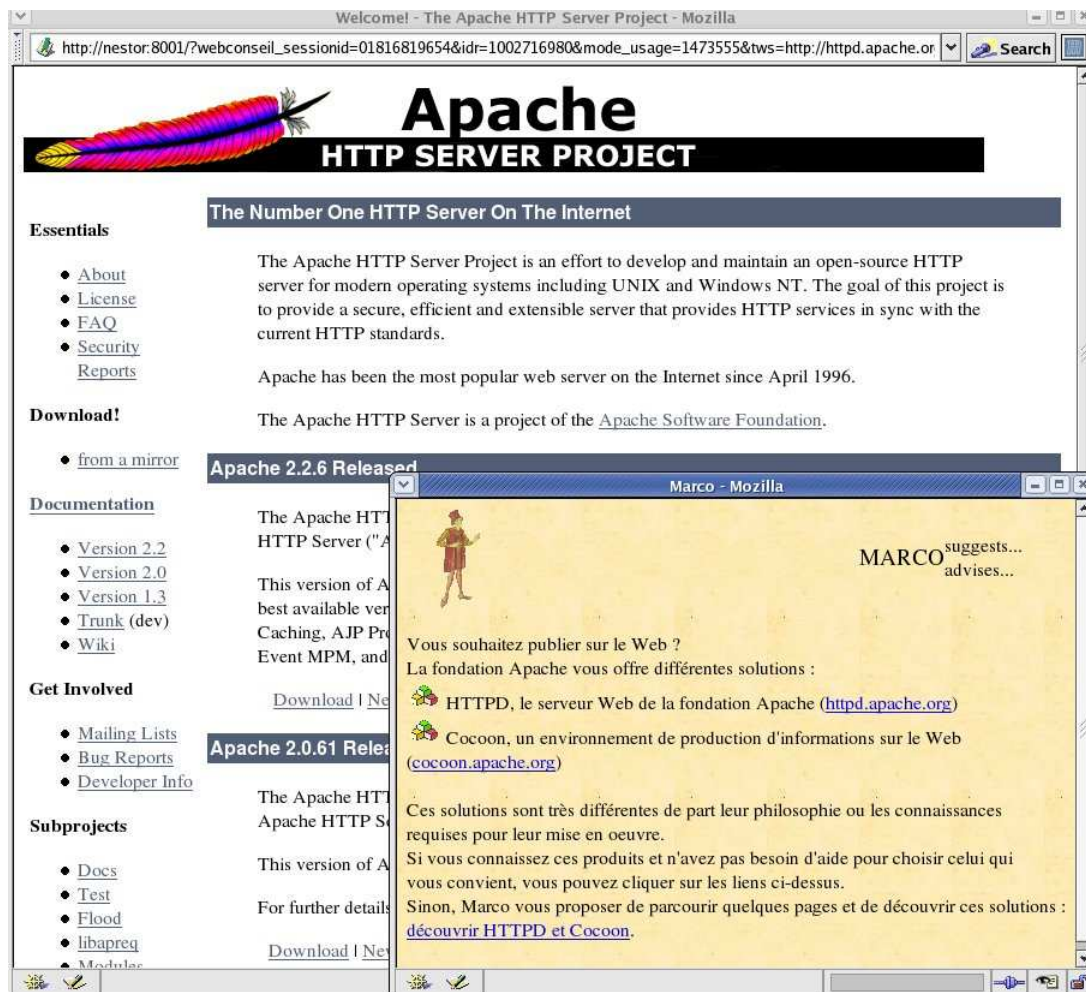


FIG. 7.9 – Exemple de conseils proposés à un utilisateur du modèle “Je veux publier sur le Web”

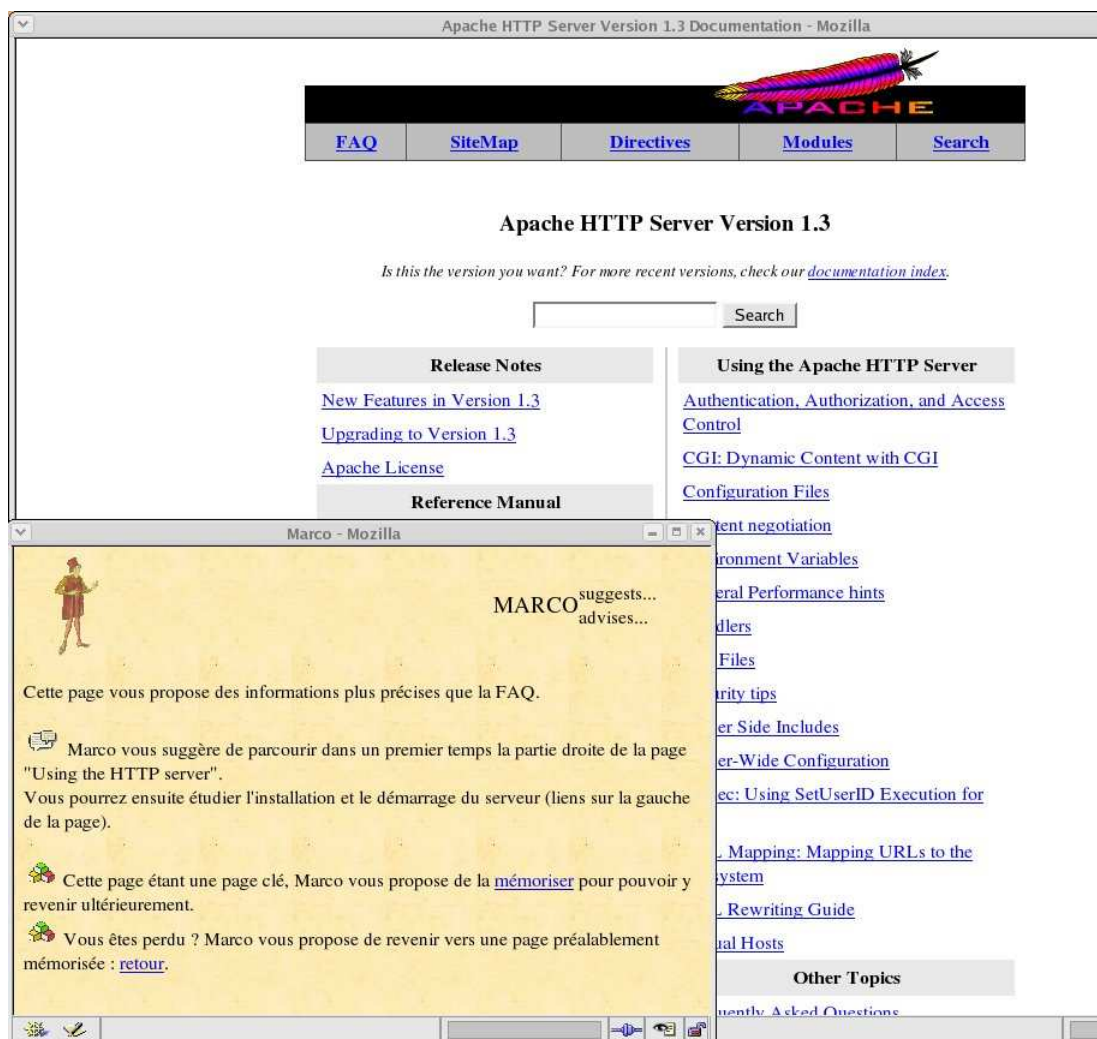


FIG. 7.10 – Exemple de conseils proposés à un utilisateur du modèle “Je veux publier sur le Web”

- Informations relatives au modèle et à destination de l'utilisateur : "l'objectif de ce modèle d'utilisation est d'aider un utilisateur voulant publier des informations sur le Web, à faire son choix entre les différentes solutions logicielles proposées par la fondation Apache. Il conviendra donc de faciliter la navigation de l'utilisateur dans le site de la fondation Apache et la découverte des pages présentant les outils, proposés par la fondation Apache, et permettant la publication d'informations sur le Web".
 - Taille du modèle : "50 noeuds"
- Données concernant le site Web cible :
 - Description : "le site de la fondation Apache regroupe une cinquantaine de projets qui permettent le développement de logiciels open source".
 - Structure du site : "le site Apache présente une architecture hiérarchique : le site principal, celui de la fondation (www.apache.org), donne accès aux sites des projets. On n'observe que peu de passerelles entre les sites des différents projets".
 - Fonctionnalités particulières proposées : "aucune : les fonctionnalités proposées dans ce site sont classiques et ne présentent pas de difficultés particulières d'utilisation".

Chapitre 8

Conclusions et perspectives

8.1 Bilan des travaux

Les utilisateurs du Web rencontrent de nombreuses difficultés en naviguant. Les systèmes conseillers représentent une solution pour tenter de faire face à ces difficultés et faciliter les navigations des utilisateurs. Nous avons présenté les différentes approches mises en oeuvre par ces systèmes, que ce soit au niveau des techniques utilisées ou du type d'aide proposé.

Nous avons défini une approche fondée sur la construction de systèmes conseillers à partir d'une architecture générique épiphyte. Cette approche est originale de par (1) la nature de l'architecture proposée et (2) la nature de l'aide apportée.

L'architecture générique épiphyte définie permet une séparation effective entre les données du serveur Web pour lequel une aide est proposée et les données propres au système conseiller. Cette séparation franche entre les systèmes autorise l'évolution indépendante de chacun des systèmes.

L'aide apportée aux utilisateurs est fondée sur la tâche que l'utilisateur souhaite réaliser en abordant un site. Pour offrir une telle aide à l'utilisateur, le système conseiller analyse la navigation de l'utilisateur en fonction de modèles d'utilisation créés par des experts et propose les conseils associés aux modèles. Cette approche impose une analyse détaillée des usages du site Web en question mais présente plusieurs avantages :

1. Le système conseiller ne propose pas une aide ponctuelle, *i.e.*, une aide associée uniquement à une page, mais une aide liée à une tâche et qui va s'étendre sur de nombreuses pages du site.
2. Le système conseiller crée et offre aux utilisateurs plusieurs vues du ou des

sites étudiés. En effet, le système conseiller peut être considéré comme un méta-système : chaque modèle d'utilisation lie des pages d'un ou de plusieurs sites et propose (1) un parcours dans le site en fonction d'un objectif donné, (2) des conseils spécifiques au parcours et à la navigation de l'utilisateur et (3) des fonctionnalités appropriées. La création d'un nouveau modèle d'utilisation peut donc être perçue comme la création d'un nouveau site dédié à un objectif donné.

Notre approche se fonde sur l'utilisation d'un système conseiller. Nous avons (1) défini un processus d'élaboration d'un système conseiller en proposant différentes étapes, de l'analyse du site Web cible à l'instanciation d'un système conseiller générique en passant par les créations des modèles de tâches et des conseils, et (2) proposé une architecture et des outils de construction fondés sur l'approche précédemment définie. Les modules composant cette architecture ont été présentés en détail. Enfin, (3), parmi les tâches nécessaires à l'élaboration d'un système conseiller, il nous est apparu que la création des modèles d'utilisation était particulièrement délicate. Nous avons proposé deux approches informatiques pour assister cette tâche, la deuxième offrant une assistance plus poussée à l'expert.

Deux prototypes ont été construits qui nous ont permis d'illustrer et de montrer la faisabilité de notre approche.

Nous avons utilisé dans l'introduction et au paragraphe 3.6 la grille d'analyse Platon 1 [TBB⁺04] comme support à l'analyse nos travaux. La grille d'analyse proposant les résultats de la recherche, présentée dans le tableau ci-dessous (tableau 8.1), a déjà été présentée dans l'introduction (tableau 1.1). Elle permet, en effet, de donner un aperçu clair et structuré des résultats de la recherche. Ce tableau trouve cependant naturellement sa place dans ce chapitre consacré au bilan des travaux.

8.2 Positionnement des travaux

Les systèmes conseillers, présentés au chapitre 2, ont pour objectifs de rendre accessible à l'utilisateur des informations qu'il pourrait avoir du mal à obtenir et/ou de limiter la charge cognitive nécessaire à l'utilisation du site. Pour réaliser ce ou ces objectifs, plusieurs types d'aide peuvent être proposés : apporter directement de l'information supplémentaire à l'utilisateur, simplifier la navigation de l'utilisateur en jouant sur les liens des pages ou sur leur contenu ou encore guider l'utilisateur. C'est dans la catégorie des systèmes proposant une aide de type guidage que l'on peut situer nos travaux. Nous avons, en effet, choisi de proposer cette forme d'aide qui couvre les autres types d'aide qu'il est possible de propo-

[C] Résultat de la recherche	
[C.1] Nature des résultats	<p>Les résultats que nous proposons sont :</p> <ul style="list-style-type: none"> - Une contribution à la définition de la notion de système conseiller (définition, éléments de classification, exemples, méthode de conception). - Des éléments de méthodologie de conception de systèmes conseillers. - Une architecture informatique générique, mise en oeuvre de l'approche proposée. - Des outils de support à la conception de systèmes conseillers. - Une preuve de concept par des prototypes opérationnels.
[C.2] Aspects généraux et génériques des résultats	<p>Plusieurs éléments de nos travaux sont réutilisables :</p> <ul style="list-style-type: none"> - Nous avons analysé les travaux sur les systèmes conseillers et présenté les différents types d'aide qu'ils permettent de proposer. La grille d'analyse proposée peut permettre de positionner d'autres travaux, que ce soit en fonction de l'aide proposée ou des techniques utilisées. - Nous avons proposé des éléments de méthodologie de conception de systèmes conseillers en précisant, notamment, les différentes étapes nécessaires à l'élaboration d'un système conseiller. Ces éléments peuvent être utilisés et adaptés, pour permettre la constructions de systèmes conseillers dans d'autres contextes.
[C.3] Types de validation des résultats	<p>Nous avons utilisé, comme méthode d'évaluation, des prototypes qui ont été construits comme "preuves de concept" : leur élaboration a validé la faisabilité de notre approche et de l'architecture proposée.</p>

TAB. 8.1 – *Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Résultat de la recherche*

ser : pour guider l'utilisateur, il est possible de lui présenter des informations supplémentaires et/ou de simplifier sa navigation. Cependant, on peut observer que nos travaux se situent en marge du courant majeur des recherches sur les systèmes conseillers puisque peu de travaux prennent en compte, explicitement, l'objectif de l'utilisateur lorsqu'il aborde le site.

8.3 Limites

Les principales limites actuelles du travail présenté dans ce mémoire sont, à notre sens, les suivantes :

1. Nous n'avons pas pu faire de mises à l'essai effectives. Seules des preuves de concept ont été proposées, qui permettent de démontrer la faisabilité de notre approche. Dans le cas de notre premier contexte d'application, le site a été stoppé, empêchant la phase de mise à l'essai. Dans le cas d'Apache, le fait que ce contexte ait été abordé tardivement dans la thèse et qu'il soit particulièrement complexe n'a pas permis d'effectuer cette phase.
2. Nous n'avons pas abordé toutes les difficultés posées par la construction semi-automatique de modèles. La difficulté de construction des modèles est apparue clairement au cours de nos travaux, et a motivé, entre autres, le fait de chercher à travailler sur un site très complexe et très utilisé comme celui de la fondation Apache. Pour pouvoir traiter les données des logs de ce site, nous avons abordé les techniques d'anonymisation des données, de simplification de données ou encore de reconstruction des sessions. Pour l'analyse proprement dite des données et en particulier la construction automatique des modèles d'utilisation des techniques comme, par exemple, les réseaux de neurones, auraient pu être utilisées. Cependant, extraire des informations des logs des serveurs Web est un sujet de recherche à part entière (comme en témoignent les nombreuses publications du domaine data mining). Le temps imparti et les limites de notre problématique ont réduit notre spectre de travail, et nous avons utilisé de simples calculs statistiques pour analyser ces logs.

8.4 Perspectives

Nous envisageons plusieurs développements à ces travaux :

- Nous avons décidé de laisser l'utilisateur choisir le modèle qui lui paraît le plus approprié. Nous avons pris cette décision pour éviter de commettre des erreurs d'appréciation : proposer ou imposer à l'utilisateur un modèle qui ne correspond pas à la tâche qu'il envisage de réaliser est risqué. A court terme, les conseils ne seront pas appropriés et l'utilisateur ne sera pas aidé. A plus long terme, l'utilisateur n'utilisera plus le système conseiller. Cependant, ce choix peut être remis en cause. Il peut être intéressant d'étudier quelles approches, quelles techniques peuvent permettre d'analyser le comportement de l'utilisateur et lui proposer (ou de modifier automatiquement), avec une marge d'erreur relativement limitée, le modèle qui semble correspondre à la tâche qu'il veut réaliser.
- Nous avons indiqué que la phase de construction des modèles d'utilisation était délicate, et précisé qu'il s'agissait d'une des limites actuelles de ce travail. En effet, dès lors que le site Web est complexe, l'expert doit manipuler de nombreux noeuds, de nombreux arcs et de nombreux conseils. Nous avons montré dans quelle mesure un outil informatique pouvait assister l'expert dans cette tâche. Nous envisageons d'aller plus loin en utilisant les logs des serveurs Web comme support à la création des modèles. Il s'agit d'analyser les logs, d'en dégager des comportements prototypiques (par exemple, on observe que de nombreuses navigations échouent sur une page donnée ou que des pages sont systématiquement parcourues ensemble au cours d'une même session) et de construire automatiquement, à partir de ces comportements prototypiques, un modèle de tâches. Resterait à l'expert la tâche, non négligeable, d'y associer des conseils.
- Nous avons élaboré plusieurs prototypes qui ont été construits comme "preuves de concept" de la faisabilité de notre approche, de l'architecture proposée et des outils proposés. En revanche, nous n'avons pas effectué de tests d'utilisabilité et d'usage de prototypes. Il sera intéressant de mettre en oeuvre ces tests pour analyser aussi bien l'aide proposée à l'expert lors de la construction des modèles (facilité à rechercher, à sélectionner et à composer les conseils élémentaires et les conseils génériques) que celle proposée à l'utilisateur du site Web et du système conseiller. A ce niveau, une perspective est de s'intéresser au bénéfice que peuvent présenter les conseils. Juger de la qualité des conseils, du bénéfice qu'en retirent ou non les utilisateurs est un travail à part entière. Ce type de travail peut, par exemple, s'effectuer à l'aide de formulaire à faire remplir aux utilisateurs à l'issue de

leur navigation ou à l'aide d'interviews. Il nécessite du temps et des compétences particulières, et se trouve, en toute hypothèse, en marge de notre problématique.

- Nous avons proposé à l'expert d'élaborer des conseils par assemblage de conseils élémentaires. Une telle structuration des conseils peut être vue comme les prémisses d'un langage d'expression de conseils. Il serait intéressant de creuser cette piste.

Annexe A

**Notre approche vue sous l'angle de
l'approche méthodologique Platon-1
[TBB⁺04]**

[A] Définition du projet de recherche	
[A.1] Objectifs de la recherche	Contribution à l'élaboration des systèmes conseillers fondés sur les principes suivants : - Prise en compte de la tâche de l'utilisateur. - Aide de type guidage. - Installation du système conseiller sans modification du site cible. - Site cible complexe.
[A.2] Contraintes sur la construction du système informatique	(1) Élaborer une architecture fonctionnelle. (2) Disposer d'un prototype utilisable.
[A.3] Finalités du système informatique	Prototype de système conseiller permettant (1) de montrer la faisabilité de notre approche et (2) de l'illustrer.

TAB. A.1 – *Notre problématique vue sous l'angle de l'approche méthodologique Platon-1 - Définition du projet de recherche*

[B] Cadre théorique de la recherche	
[B.6] Théorie de la conception	Nous avons adopté un processus de conception itératif : après avoir défini la problématique, nous avons proposé une approche puis conçu une première architecture sur laquelle nous nous sommes fondés pour construire un prototype illustrant l'approche et sa faisabilité. La construction puis l'analyse de ce prototype nous ont conduit à faire évoluer notre approche et proposer une deuxième architecture, puis un deuxième prototype fondé sur la nouvelle architecture.

TAB. A.2 – *Notre problématique vue sous l'angle de l'approche méthodologique Platon-1 - Cadre théorique de la recherche*

[C] Résultat de la recherche	
[C.1] Nature des résultats	<p>Les résultats que nous proposons sont :</p> <ul style="list-style-type: none"> - Une contribution à la définition de la notion de système conseiller (définition, éléments de classification, exemples, méthode de conception). - Des éléments de méthodologie de conception de systèmes conseillers. - Une architecture informatique générique, mise en oeuvre de l'approche proposée. - Des outils de support à la conception de systèmes conseillers. - Une preuve de concept par des prototypes opérationnels.
[C.2] Aspects généraux et génériques des résultats	<p>Plusieurs éléments de nos travaux sont réutilisables :</p> <ul style="list-style-type: none"> - Nous avons analysé les travaux sur les systèmes conseillers et présenté les différents types d'aide qu'ils permettent de proposer. La grille d'analyse proposée peut permettre de positionner d'autres travaux, que ce soit en fonction de l'aide proposée ou des techniques utilisées. - Nous avons proposé des éléments de méthodologie de conception de systèmes conseillers en précisant, notamment, les différentes étapes nécessaires à l'élaboration d'un système conseiller. Ces éléments peuvent être utilisés et adaptés, pour permettre la constructions de systèmes conseillers dans d'autres contextes.
[C.3] Types de validation des résultats	<p>Nous avons utilisé, comme méthode d'évaluation, des prototypes qui ont été construits comme "preuves de concept" : leur élaboration a validé la faisabilité de notre approche et de l'architecture proposée.</p>

TAB. A.3 – Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Résultat de la recherche

[D] Cycle de vie de la recherche	
[D.1] Contexte de lancement de la recherche	Un contrat industriel a été signé entre le LIUM et le ministère de l'Education Nationale et de la Recherche. L'objectif de ce contrat était la proposition d'une approche et d'un prototype visant à étudier l'impact potentiel de l'association d'un système conseiller au site Web Educasource.
[D.2] Historique de la recherche	J'ai commencé à travailler en tant qu'ingénieur d'étude sur cette problématique. Ce travail s'est ensuite poursuivi, sous la direction de P. Tchounikine, dans le cadre d'une thèse.

TAB. A.4 – *Notre approche vue sous l'angle de l'approche méthodologique Platon-1 - Cycle de vie de la recherche*

Annexe B

Pré-traitements effectués sur les logs

Nous présentons ci-dessous les codes des pré-traitements effectués sur les logs : anonymisation, simplification puis extraction des sessions.

```
/*
 *
 *
 * ANONYMISATION DES DONNEES
 *
 *
 * exemple d'enregistrement que l'on
 * retrouve dans les fichiers de logs :
 * archive.apache.org 193.52.29.131 - bob [17/Jul/2005:00:00:07 +0200]
 *      "GET /dist/ HTTP/1.1" 200 4213 "http://www.eu.apache.org/dist/"
 *      "Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.8)
 *      Gecko/20050511 Firefox/1.0.4"
 *
 * données après modifications :
 * archive.apache.org 11111 - 22222 [17/Jul/2005:00:00:07 +0200]
 *      "GET /dist/ HTTP/1.1" 200 4213 "http://www.eu.apache.org/dist/"
 *      "Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.8)
 *      Gecko/20050511 Firefox/1.0.4"
 *
 */

private static void anonymiser(String[] nom_fichier_logs,
```

```

int nb_fichiers_a_traiter)

{
String ligne;
String ip;
String ip_anonymisee;

String remote_user;
String identifiant;
String identifiant_anonymise;

String web;
String fin_ligne;
int indice0;
int indice;
int indice2;
int indice3;
int indice_fichier;
String nom_fichier_sortie;

try
{
    indice_fichier = 0;
    while (indice_fichier < nb_fichiers_a_traiter )
    {
        // gestion des fichiers
        // entrée
        BufferedReader fichier_entree =
            new BufferedReader(
                new FileReader(nom_fichier_logs[indice_fichier]));
        System.out.println("Lecture du fichier de logs
            (" + nom_fichier_logs[indice_fichier] + ")");
        // sortie
        nom_fichier_sortie =
            nom_fichier_logs[indice_fichier] + ".Anonymized";
        File test = new File(nom_fichier_sortie);
        if (test.exists())
        {
            System.out.println(
                "Enregistrement des données dans fichier "
                + nom_fichier_sortie + " IMPOSSIBLE : le fichier existe");
            System.exit(0);
        }
    }
}

```

```

}
indice_fichier++;

FileOutputStream fichier_sortie =
    new FileOutputStream(nom_fichier_sortie);
nb_ligne = 0;

while ((ligne = fichier_entree.readLine()) != null)
{
    // extraction site Web cible
    indice0 = ligne.indexOf(" ");
    web = ligne.substring(0, indice0);

    // extraction IP
    indice = ligne.indexOf(" ", indice0+1);
    ip = ligne.substring(indice0+1, indice);
    // anonymisation de l'IP
    ip_anonymisee = anonymiserIP(ip);

    // extraction nom distant
    indice2 = ligne.indexOf(" ", indice + 1);
    remote_user = ligne.substring(indice + 1, indice2);

    // extraction identifiant utilisateur authentifié
    indice3 = ligne.indexOf(" ", indice2 + 1);
    identifiant = ligne.substring(indice2 + 1, indice3);
    // anonymisation de l'identifiant
    identifiant_anonymise = anonymiserIdentifiant(identifiant);

    // récupération de la fin de la ligne
    // (pas de modif à faire)
    fin_ligne = ligne.substring(indice3 + 1);
    fichier_sortie.write((web + " " + ip_anonymisee + " "
        + remote_user + " " + identifiant_anonymise
        + " " + fin_ligne + "\n").getBytes());

    // System.out.println(ip_anonymisee+" "
        +remote_user+" "+identifiant_anonymise+" "+fin_ligne);

    fichier_entree.close();
    fichier_sortie.close();
}

```

```

        }
    }
    catch (FileNotFoundException e)
    {
        System.err.println(
            "ouverture impossible du fichier des utilisateurs.");
    }
    catch (NoSuchElementException e)
    {
        System.out.println("pb fichier des utilisateurs");
    }
    catch (IOException e)
    {
        System.out.println("pb lecture fichier des utilisateurs");
    }

    System.out.println("Anonymisation OK");
}

/**
 * @param identifiant de l'utilisateur
 * @return identifiant anonymisé
 */
private static String anonymiserIdentifiant(String identifiant)
{
    String retour;

    if (!identifiant.equals("-"))
    {
        // recherche de l'identifiant dans la table
        retour = (String) ids.get(identifiant);

        if (retour == null)
        {
            retour = ""+indiceID;
            indiceID++;
            ids.put(identifiant, retour);
        }

        return retour;
    }
}

```

```

else
{
    return identifiant;
}
}

/**
 * @param IP
 * @return IP anonymisé
 */
private static String anonymiserIP(String ip)
{
    String retour;

    // recherche de l'IP dans la table
    retour = (String) ips.get(ip);

    if (retour == null)
    {
        retour = ""+indiceIP;
        indiceIP++;
        ips.put(ip, retour);
    }

    return retour;
}

```

```

/*
 *
 * SIMPLIFICATION DES DONNEES
 *
 * exemple :
 * archive.apache.org 11111 - 22222 [17/Jul/2005:00:00:07 +0200]
 *      "GET /dist/ HTTP/1.1" 200 4213 "http://www.eu.apache.org/dist/"
 *      "Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.8)
 *      Gecko/20050511 Firefox/1.0.4"
 */
private static void simplifier(String nom_fichier_logs,
                               String nom_fichier_sortie)
{
    String ligne;
    String requete;
    int indice0;
    int indice;
    int sup;

    sup = 0;
    try
    {
        // gestion des fichiers
        // entrée
        BufferedReader fichier_entree = new BufferedReader(
            new FileReader(nom_fichier_logs));
        System.out.println("Lecture du fichier de logs ("
            + nom_fichier_logs + ")");

        File test = new File(nom_fichier_sortie);
        if (test.exists())
        {
            System.out.println(
                "Enregistrement des données dans fichier "
                + nom_fichier_sortie + " IMPOSSIBLE : le fichier existe");
            System.exit(0);
        }

        // sortie

```

```

FileOutputStream fichier_sortie =
    new FileOutputStream(nom_fichier_sortie);

while ((ligne = fichier_entree.readLine()) != null)
{
    // extraction de la requete
    indice0 = ligne.indexOf("\");
    indice = ligne.indexOf("\", indice0+1);
    requete = ligne.substring(indice0, indice);
    if (requete.indexOf(".jpg") > 0
        || requete.indexOf(".gif") > 0
        || requete.indexOf(".css") > 0
        || requete.indexOf(".png") > 0
        || requete.indexOf("robots.txt") > 0
        || requete.indexOf(".rss") > 0
        || requete.indexOf(".js") > 0
        || requete.indexOf(".ico") > 0
        || requete.indexOf(".dtd") > 0
        || requete.startsWith("\POST")
        || requete.startsWith("\SEARCH")
        || requete.startsWith("\OPTIONS")
        || requete.startsWith("\HEAD")
        || requete.startsWith("\GET http")
        || requete.startsWith("\GET ftp") )
    {
        sup++;
    }
    else
    {
        requete = ligne.substring(indice);
        // suppression des robots et autres vérificateurs
        if (requete.indexOf("check_http/") > 0
            || requete.indexOf("msnbot") > 0
            || requete.indexOf("ia_archiver") > 0
            || requete.indexOf("Googlebot") > 0
            || requete.indexOf("swish-e spider") > 0
            || requete.indexOf("Echoping") > 0
            || requete.indexOf("Bookmark Renewal Check Agent") > 0
        )
        {
            sup++;
        }
    }
}

```



```

        else
        {
            // on ne garde que les requêtes terminées avec succès
            if (requete.startsWith("\n 200 "))
                fichier_sortie.write((ligne + "\n").getBytes());
            else
                sup++;
        }
    }

    fichier_entree.close();
    fichier_sortie.close();
}
catch (FileNotFoundException e)
{
    System.err.println("ouverture impossible du fichier des utilisateurs");
}
catch (NoSuchElementException e)
{
    System.out.println("pb fichier des utilisateurs");
}
catch (IOException e)
{
    System.out.println("pb lecture fichier des utilisateurs");
}

System.out.println("Simplification OK - lignes supprimées : "+sup)
}

```

```

/*
 *
 * EXTRACTION DES SESSIONS
 *
 */

public class ExtraireSessions
{
    // durée d'une session en minutes
    static int DUREE_SESSION = 30;
    // suppressions dans un soucis de clarté
    [...]

    // paramètres
    // [1..n-1] : fichiers à traiter
    // n        : fichier de sortie
    public static void main(String[] args)
    {

        // table des utilisateurs recensés
        // key : ip+" "+type_navigateur
        // value : adresse dans la table requetes
        // de la première requete
        Hashtable utilisateurs;

        // requetes des utilisateurs
        //[0] : numero de session
        //[1] : la requete
        //[2] : date
        String[][] requetes;

        // pointeur vers la requete suivante
        int[] requete_svte;

        // pointeur vers la dernière requete
        Hashtable dernière_requete;

```

```

// initialisations
utilisateurs = new Hashtable();
requetes = new String[MAX_REQUETES][3];
requete_svte = new int[MAX_REQUETES];
dernière_requete = new Hashtable();

// suppressions dans un soucis de clarté
[...]

extraireSessions(nom_fichier_logs,
                  nb_fichiers_a_traiter,
                  utilisateurs,
                  requetes,
requete_svte, dernière_requete);
enregistrementDonnées(nom_fichier_logs,
                      nb_fichiers_a_traiter,
                      utilisateurs, requetes,
                      requete_svte, dernière_requete);
}

/*
* exemple :
* archive.apache.org 11111 - 22222 [17/Jul/2005:00:00:07 +0200]
*      "GET /dist/ HTTP/1.1" 200 4213 "http://www.eu.apache.org/dist/"
*      "Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.8)
*      Gecko/20050511 Firefox/1.0.4"
*/
private static void extraireSessions(
    String[] nom_fichier_logs,
    int nb_fichiers_a_traiter,
    Hashtable utilisateurs,
    String[][] requetes,
    int[] requete_svte,
    Hashtable dernière_requete)
{
String ligne;

String nomsite;
String ip;
String date;

```

```

String retour;
String requete;
int indice;
int indice1;
int indice2;
int indice3;
int indice_fichier;
int indice_requetes;
int nbRequetesTraitees;

// on va lire les fichier de logs
indice_fichier = 0;
indice_requetes = 0;

nbRequetesTraitees = 0;

// lecture des différents fichiers
while (indice_fichier < (nb_fichiers_a_traiter - 1))
{
    try
    {
        System.out.println("Lecture du fichier de logs (" +
            nom_fichier_logs[indice_fichier] + ")");

        BufferedReader fichier_entree =
            new BufferedReader(new FileReader(
                nom_fichier_logs[indice_fichier]));

        indice_fichier++;

        // lecture d'un fichier
        while ((ligne = fichier_entree.readLine()) != null)
        {
            indice = ligne.indexOf(" ");

            if (indice >= 0)
            {
                // extraction du nom du site
                nomsite = ligne.substring(0, indice);
                indice1 = ligne.indexOf(" ", indice + 1);
            }
        }
    }
}

```

```

if (indice1 >= 0)
{
    // extraction de l'IP
    ip = ligne.substring(indice + 1, indice1);
    indice2 = ligne.indexOf("[", indice1 + 1);
    if (indice2 >= 0)
    {
        indice3 = ligne.indexOf("]", indice2 + 1);

        if (indice3 >= 0)
        {
            // extraction de la date
            date = ligne.substring(indice2 + 1, indice3);

            int indice4 = ligne.indexOf("\\"", indice3 + 1);

            if (indice4 >= 0)
            {
                int indice5 = ligne.indexOf("\\"", indice4 + 1);

                if (indice5 >= 0)
                {
                    // extraction de la requête
                    requete = ligne.substring(indice4 + 1, indice5);

                    // simplification de la requete
                    indice2 = requete.indexOf(" ");

                    if (indice2 >= 0)
                    {
                        indice3 = requete.indexOf(" ", indice2 + 1);

                        if (indice3 >= 0)
                        {
                            requete = requete.substring(indice2 + 1, indice3);
                        }
                    }
                }

                // extraction du navigateur
                indice = ligne.indexOf("\\"", indice5 + 1);
            }
        }
    }
}

```

```

if (indice >= 0)
{
    indice1 = ligne.indexOf("\"", indice + 1);

    if (indice1 >= 0)
    {
        indice2 = ligne.indexOf("\"", indice1 + 1);

        if (indice2 >= 0)
        {
            indice3 = ligne.indexOf("\"", indice2 + 1);

            if (indice3 >= 0)
            {
                String navigateur =
                    ligne.substring(indice2 + 1, indice3);

                // traitements
                retour = (String)
                    utilisateurs.get(ip + " " + navigateur);

                if (retour == null)
                // l'utilisateur n'existe pas ds la table
                {
                    // on ajoute l'utilisateur dans la table utilisateurs
                    utilisateurs.put(ip + " "
                        +navigateur, indice_requetes + "");

                    // on ajoute l'utilisateur la table dernière_requete
                    dernière_requete.put(ip + " "
                        +navigateur, indice_requetes + "");

                    // on met à jour le numero de session
                    // dans la table requete
                    requetes[indice_requetes][0] = "0";
                }
            }
        }
    }
    else // l'utilisateur est connu...
    {
        // on recherche sa dernière requete
        int indice_dernière_requete =
            (new Integer((String)

```

```

dernière_requete.get(ip + " "
+ navigateur))).intValue();

// on met à jour la table requete suivante
requete_svte[indice_dernière_requete] =
    indice_requetes;

// on met à jour la table dernière_requete
dernière_requete.remove(ip + " " +navigateur);
dernière_requete.put(ip + " " +
navigateur, indice_requetes + "");

// on compare les dates pour voir
// si on est tjs dans la meme session
if (!memeSession(
requetes[indice_dernière_requete][2],date))
{
    // on incremente le numero de session
    requetes[indice_requetes][0] = " " +
        (new Integer((String)
        requetes[indice_dernière_requete][0]).intValue() +1);
}
else
{
    // on conserve le meme numero de session
    requetes[indice_requetes][0] =
        requetes[indice_dernière_requete][0];
}
}

// on met à jour la table requete suivante
requete_svte[indice_requetes] = -1;

// on met à jour la table requete
requetes[indice_requetes][1] = nomsite +requete;
requetes[indice_requetes][2] = date;

indice_requetes++;

if (indice_requetes == MAX_REQUETES)
{

```

```

        System.out.println("indice_requetes == MAX_REQUETES
        => sortie");
        System.exit(0);
    }

    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }
    else {System.out.println("error fichier log : "+ligne);}
    }

    fichier_entree.close();
}
catch (FileNotFoundException e)
{
    System.err.println("ouverture impossible du fichier de logs.");
    indice_fichier = nb_fichiers_a_traiter;
}
catch (NoSuchElementException e)
{
    System.out.println("pb fichier de logs");
    indice_fichier = nb_fichiers_a_traiter;
}
catch (IOException e)
{

```



```

        System.out.println("pb lecture fichier de logs");
        indice_fichier = nb_fichiers_a_traiter;
    }
}
}

/**
 * @param date_courante
 * @param date_precedente
 * format des dates : 01/Jun/2005:00:03:46 +0200
 * @return
 */
private static boolean memeSession(String date_precedente,
                                    String date_courante)
{
    int indice;
    boolean retour;
    Date dc;
    Date dp;
    Date dc2;

    indice = date_precedente.indexOf(" ");
    retour = false;

    if (indice >= 0)
    {
        date_precedente = date_precedente.substring(0, indice);
        indice = date_courante.indexOf(" ");

        if (indice >= 0)
        {
            date_courante = date_courante.substring(0, indice);

            DateFormat dateFormat =
                new SimpleDateFormat("dd/MMM/yyyy:HH:mm:ss", Locale.US);

            try
            {
                dc = dateFormat.parse(date_courante);
            }
        }
    }
}

```

```

        GregorianCalendar cal = new GregorianCalendar();
        cal.setTime(dc);

        long dcl = cal.getTimeInMillis();
        dcl = dcl - (DUREE_SESSION * 60 * 1000);
        dc2 = new Date(dcl);

        dp = dateFormat.parse(date_precedente);
        retour = dc2.before(dp);
    }
    catch (ParseException e)
    {
        e.printStackTrace();
    }
}

return retour;
}

private static String calculDuree(
    String date_precedente,
    String date_courante)
{
    Date dc;
    long dcl;
    Date dp;
    long dpl;
    GregorianCalendar cal;
    long retour;
    String retour_s;
    Long retour_l;

    DateFormat dateFormat =
        new SimpleDateFormat("dd/MMM/yyyy:HH:mm:ss", Locale.US);
    retour_s = "-1";
    try
    {
        cal = new GregorianCalendar();

```

```

        dc = dateFormat.parse(date_courante);
        cal.setTime(dc);
        dcl = cal.getTimeInMillis();

        dp = dateFormat.parse(date_precedente);
        cal.setTime(dp);
        dpl = cal.getTimeInMillis();

        retour = (dcl - dpl)/1000;
        retour_l = new Long(retour);
        retour_s = (new Integer(retour_l.intValue())).toString();
    }
    catch (ParseException e)
    {
        e.printStackTrace();
    }

    return retour_s;
}

}

```

Annexe C

Exemples de statistiques effectuées sur les logs

Sites utilisés comme entrée		
Calculs sur 2005		
jakarta.apache.org	20,22	%
httpd.apache.org	18,81	%
www.apache.org	16,82	%
mail-archives.apache.org	9,01	%
struts.apache.org	5,40	%
ant.apache.org	5,00	%
xml.apache.org	4,17	%
ws.apache.org	2,73	%
archive.apache.org	2,61	%
spamassassin.apache.org	2,11	%
logging.apache.org	1,89	%
maven.apache.org	1,77	%
portals.apache.org	1,11	%
cocoon.apache.org	1,04	%
wiki.apache.org	1,02	%
incubator.apache.org	1,01	%
lucene.apache.org	0,85	%
db.apache.org	0,82	%
myfaces.apache.org	0,51	%
lenya.apache.org	0,34	%
geronimo.apache.org	0,33	%
forrest.apache.org	0,29	%
james.apache.org	0,27	%
ajax.apache.org	0,27	%
xmlbeans.apache.org	0,27	%
apr.apache.org	0,23	%
directory.apache.org	0,20	%
xmlgraphics.apache.org	0,17	%
ibatis.apache.org	0,16	%
avalon.apache.org	0,15	%
beehive.apache.org	0,12	%
excalibur.apache.org	0,10	%
gump.apache.org	0,05	%
tomcat.apache.org	0,05	%
perl.apache.org	0,04	%
xerces.apache.org	0,01	%
commons.apache.org	0,01	%
bugs.apache.org	0,01	%
search.apache.org	0,01	%
xalan.apache.org	0,01	%
tcl.apache.org	0,00	%

FIG. C.1 – Sites utilisés comme entrée

Mois (2005)	+2sess	+ 3 sess	Nb sessions	Nb requêtes (après simplification des données)
1	0,17		172 965,00	524 803,00
2	0,4		199 622,00	807 705,00
3	21,88	4,71	2 381 063,00	11 853 901,00
4	22,47	5,26	5 884 534,00	28 332 869,00
5	22,63	4,9	5 513 253,00	25 922 961,00
6	20,26	4,15	5 976 288,00	28 022 993,00
7	20,14	4,19	5 256 560,00	25 466 676,00
8	20,28	4,09	5 485 671,00	26 479 031,00
9	20	3,95	5 560 305,00	26 786 572,00
10	17,89	3,45	2 962 281,00	13 946 073,00
11	5,14		675 638,00	2 766 507,00
12	6,81		532 763,00	2 094 367,00
			40 600 943,00	193 004 458,00
	Moyenne (sans 1,2, 11 et 12)			
	20,69	4,34	4877494,38	23351384,5

FIG. C.2 – Sites parcourus ensemble lors d'une même session

FROM WWW (05 2005)			
www.apache.org	->	www.apache.org:	
www.apache.org	->	httpd.apache.org:	40,94 %
www.apache.org	->	jakarta.apache.org:	17,79 %
www.apache.org	->	xml.apache.org:	6,22 %
www.apache.org	->	ws.apache.org:	4,37 %
www.apache.org	->	archive.apache.org:	3,87 %
www.apache.org	->	ant.apache.org:	3,37 %
www.apache.org	->	struts.apache.org:	2,32 %
www.apache.org	->	maven.apache.org:	2,23 %
www.apache.org	->	wiki.apache.org:	1,81 %
www.apache.org	->	mail-archives.apache.org:	1,79 %
www.apache.org	->	logging.apache.org:	1,43 %
www.apache.org	->	cocoon.apache.org:	1,30 %
www.apache.org	->	incubator.apache.org:	1,21 %
www.apache.org	->	portals.apache.org:	1,09 %
www.apache.org	->	db.apache.org:	1,07 %
www.apache.org	->	lenya.apache.org:	0,92 %
www.apache.org	->	spamassassin.apache.org:	0,89 %
www.apache.org	->	lucene.apache.org:	0,88 %
www.apache.org	->	geronimo.apache.org:	0,88 %
www.apache.org	->	apr.apache.org:	0,79 %
www.apache.org	->	myfaces.apache.org:	0,72 %
www.apache.org	->	directory.apache.org:	0,70 %
www.apache.org	->	xmlbeans.apache.org:	0,65 %
www.apache.org	->	james.apache.org:	0,64 %
www.apache.org	->	excalibur.apache.org:	0,55 %
www.apache.org	->	forrest.apache.org:	0,51 %
www.apache.org	->	xmlgraphics.apache.org:	0,41 %
www.apache.org	->	gump.apache.org:	0,34 %
www.apache.org	->	avalon.apache.org:	0,23 %
www.apache.org	->	bugs.apache.org:	0,03 %
www.apache.org	->	commons.apache.org:	0,02 %
www.apache.org	->	xerces.apache.org:	0,01 %
www.apache.org	->	xalan.apache.org:	0,01 %

FIG. C.3 – Sites parcourus après *www.apache.org*

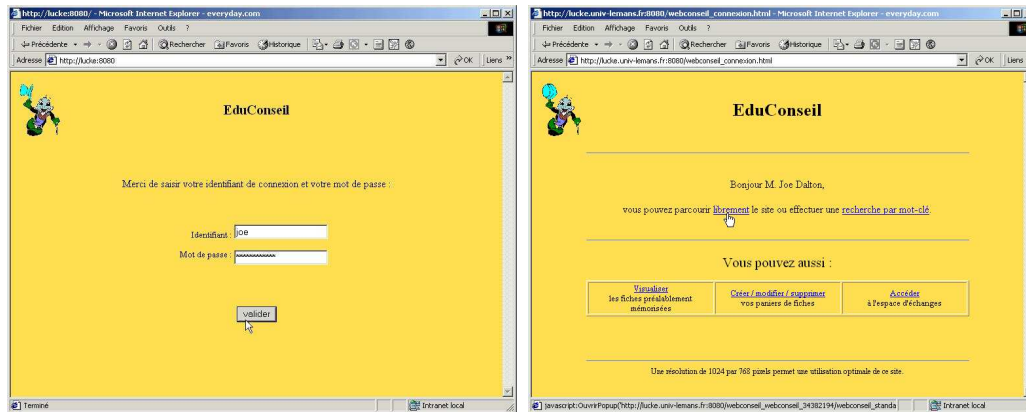
7	ajax.apache.org	->	cocoon.apache.org/	0,01	%
81	<u>ant.apache.org Somme</u>			3,77	%
145	<u>apr.apache.org Somme</u>			3,16	%
239	<u>archive.apache.org Somme</u>			0,67	%
241	<u>avalon.apache.org Somme</u>			0,24	%
243	<u>bugs.apache.org Somme</u>			0,03	%
671	<u>cocoon.apache.org Somme</u>			25,77	%
673	<u>commons.apache.org Somme</u>			0,00	%
714	<u>db.apache.org Somme</u>			2,13	%
761	<u>directory.apache.org Somme</u>			1,49	%
805	<u>excalibur.apache.org Somme</u>			1,65	%
871	<u>forrest.apache.org Somme</u>			2,95	%
896	<u>geronimo.apache.org Somme</u>			1,25	%
906	<u>gump.apache.org Somme</u>			1,07	%
1104	<u>httpd.apache.org Somme</u>			4,53	%
1187	<u>incubator.apache.org Somme</u>			1,29	%
1267	<u>issues.eu.apache.org Somme</u>			0,54	%
1733	<u>jakarta.apache.org Somme</u>			8,43	%
1759	<u>james.apache.org Somme</u>			0,84	%
1838	<u>lenya.apache.org Somme</u>			4,33	%
1876	<u>logging.apache.org Somme</u>			0,81	%
1894	<u>lucene.apache.org Somme</u>			0,67	%
2141	<u>mail-archives.apache.org Somme</u>			0,86	%
2211	<u>maven.apache.org Somme</u>			1,09	%
2238	<u>myfaces.apache.org Somme</u>			0,81	%
2331	<u>portals.apache.org Somme</u>			2,29	%
2348	<u>spamassassin.apache.org Somme</u>			0,69	%
2414	<u>struts.apache.org Somme</u>			1,71	%
2764	<u>wiki.apache.org Somme</u>			3,01	%
2852	<u>ws.apache.org Somme</u>			1,88	%
3072	<u>www.apache.org Somme</u>			15,00	%
3074	<u>www.apache.orgGET Somme</u>			0,00	%
3076	<u>xalan.apache.org Somme</u>			0,00	%
3080	<u>xerces.apache.org Somme</u>			0,01	%
3330	<u>xml.apache.org Somme</u>			5,67	%
3360	<u>xmlbeans.apache.org Somme</u>			0,73	%
3372	<u>xmlgraphics.apache.org Somme</u>			0,65	%

FIG. C.4 – Sites parcourus avant cocoon.apache.org

Annexe D

Exemple de parcours du site Educasource et de fonctionnement du système conseiller

Exemple de session : parcours libre de type « recherche par thème »



Joe Dalton s'identifie auprès du système conseiller. Son but est de rechercher des ressources éducatives en géométrie pour son enseignement en collège. Joe Dalton choisit d'utiliser un parcours libre.



Sur la page d'accueil, la fenêtre de conseil indique les différentes possibilités. Joe Dalton sélectionne le seul thème qui l'intéresse : "Mathématiques et sciences expérimentales".

EDUCASOURCE : Des ressources électroniques pour l'enseignement - Microsoft Internet Explorer - everyday.com

Des ressources électroniques pour l'enseignement

Recherche personnalisée Forum Proposer une ressource Autres sites

EDUCASOURCE Accueil

Le site FAQ Questions juridiques Nous contacter

Catégories

dans "Mathématiques et sciences expérimentales"

Rechercher Ok

Accueil > Mathématiques et sciences expérimentales (1340)

■ Animaux	(158)
■ Astronomie	(157)
■ Chimie	(144)
■ Mathématiques	(249)
■ Paléontologie, paléobotanique	(28)
■ Physique	(183)
■ Plantes	(65)
■ Sciences de la terre	(280)
■ Sciences de la vie	(220)
■ Théorie, enseignement, histoire, usuels	(96)

EduConseil - Microsoft Internet Explorer - everyday.com

Evaluer la pertinence d'une liste de sous-thèmes

Nous avons ici 10 sous-thèmes pour le thème Mathématiques et sciences expérimentales; il s'agit de préciser encore les choses en s'orientant vers un sous-thème particulier ou de choisir une recherche transversale par mot clé.

- un des sous-thèmes semble approprié : le sélectionner dans le navigateur
- plusieurs sous-thèmes semblent appropriés : nous pouvons vous proposer d'en sélectionner plusieurs, de partir sur un des thèmes et de revenir sur les autres plus tard [Sélectionner](#)
- le thème général semble intéressant mais les sous-thèmes proposés ne vous guident : identifiez un mot-clé dénotant ce que vous cherchez dans le navigateur
- rien de pertinent, vous avez l'impression d'avoir fait fausse route : il faut remettre en question l'orientation de la recherche [Revenir](#)
- vous désirez revenir à la page d'accueil [Accueil](#)

Intranet local

Joe Dalton est intéressé par deux sous-thèmes. Il a la possibilité de les sélectionner pour que le système les mémorise.

EduConseil - Microsoft Internet Explorer - everyday.com

Des ressources électroniques pour l'enseignement

EduConseil

Catégories

dans "Mathématiques et sciences expérimentales"

Accueil > Mathématiques et sciences expérimentales (1340)

<input type="checkbox"/> Animaux	(158)
<input type="checkbox"/> Astronomie	(157)
<input type="checkbox"/> Chimie	(144)
<input checked="" type="checkbox"/> Mathématiques	(249)
<input type="checkbox"/> Paléontologie, paléobotanique	(28)
<input type="checkbox"/> Physique	(183)
<input type="checkbox"/> Plantes	(65)
<input type="checkbox"/> Sciences de la terre	(280)
<input type="checkbox"/> Sciences de la vie	(220)
<input checked="" type="checkbox"/> Théorie, enseignement, histoire, usuels	(96)

Sélectionnez les thèmes qui vous semblent appropriés puis [validez](#) votre sélection

© Centre National de Documentation Pédagogique

Les deux sous-thèmes sont mémorisés par le système conseiller.

EDUCASOURCE : Des ressources électroniques pour l'enseignement - Microsoft Internet Explorer - everyday.com

Des ressources électroniques pour l'enseignement

Recherche personnalisée Forum Proposer une ressource Autres sites

EDUCA SOURCE Accueil Le site FAQ Questions juridiques Nous contacter

Catégories dans "Mathématiques et sciences expérimentales" Rechercher [] Ok

Accueil > Mathématiques et sciences expérimentales (1340)

■ Animaux	(158)
■ Astronomie	(157)
■ Chimie	(144)
■ Mathématiques	(249)
■ Paléontologie, paléobotanique	(28)
■ Physique	(183)
■ Plantes	(65)
■ Sciences de la terre	(280)
■ Sciences de la vie	(220)
■ Théorie, enseignement, histoire, usuels	(96)

EduConseil - Microsoft Internet Explorer - everyday.com

Evaluer la pertinence d'une liste de sous-thèmes

Nous avons ici 10 sous-thèmes pour le thème Mathématiques et sciences expérimentales; il s'agit de préciser encore les choses en s'orientant vers un sous-thème particulier ou de choisir une recherche transversale par mot clé.

- un des sous-thèmes semble approprié : le sélectionner dans le navigateur
- plusieurs sous-thèmes semblent appropriés : nous pouvons vous proposer d'en sélectionner plusieurs, de partir sur un des thèmes et de revenir sur les autres plus tard [↳ Sélectionner](#)
- le thème général semble intéressant mais les sous-thèmes proposés ne vous guident pas : identifiez un mot-clé dénotant ce que vous cherchez dans le navigateur
- rien de pertinent, vous avez l'impression d'avoir fait fausse route : il faut remettre en question l'orientation de la recherche [↳ Revenir](#)
- vous désirez revenir à la page d'accueil [↳ Accueil](#)

Intranet local

Joe part sur le premier sous-thème "Mathématiques".

EDUCASOURCE : Des ressources électroniques pour l'enseignement - Microsoft Internet Explorer - everyday.com

Des ressources électroniques pour l'enseignement

Recherche personnalisée Forum Proposer une ressource Autres sites

EDUCA SOURCE Accueil Le site FAQ Questions juridiques Nous contacter

Catégories dans "Mathématiques et sciences expérimentales > Mathématiques" Rechercher [] Ok

Vous pouvez aussi faire une sélection avec les cases à cocher Voir la sélection

Accueil > Mathématiques et sciences expérimentales > **Mathématiques (249)**

<input type="checkbox"/> Algèbre et théorie des nombres	(9)
<input type="checkbox"/> Analyse	(7)
<input type="checkbox"/> Arithmétique	(13)
<input type="checkbox"/> Généralités	(192)
<input checked="" type="checkbox"/> Géométrie	(47)
<input type="checkbox"/> Principes généraux des mathématiques	(12)
<input type="checkbox"/> Probabilités, mathématiques appliquées	(8)
<input type="checkbox"/> Topologie	(1)

EduConseil - Microsoft Internet Explorer - everyday.com

Evaluer la pertinence d'une liste de sous-sous-thèmes

Nous avons ici 8 sous-thèmes pour le thème Mathématiques et sciences expérimentales et Mathématiques; il s'agit de préciser encore les choses en s'orientant vers un sous-thème particulier ou de choisir une recherche transversale par mot clé.

- un des sous-thèmes semble approprié : le sélectionner dans le navigateur, vous accédez aux fiches correspondantes
- plusieurs sous-thèmes semblent appropriés :
 - vous pouvez visualiser toutes les fiches correspondant à ces sous-thèmes en les cochant
 - vous pouvez en sélectionner plusieurs, partir sur un des thèmes et revenir sur les autres plus tard.
- le thème général semble intéressant mais les sous-thèmes proposés ne vous guident pas : identifiez un mot-clé dénotant ce que vous cherchez dans le navigateur
- rien de pertinent, vous avez l'impression d'avoir fait fausse route : il faut remettre en question

Intranet local

Dans la liste des sous-thèmes proposée, il n'est intéressé que par "Géométrie".



Son choix le conduit à un trop grand nombre de fiches. Joe choisit de restructurer ces 47 fiches.



Le système propose à Joe une structuration par les codes Dewey trouvés dans les 47 fiches.

Joe s'intéresse à l'enseignement secondaire et suis le lien correspondant.

EduConseil - Microsoft Internet Explorer - everyday.com

Des ressources électroniques pour l'enseignement

Recherche personnalisée Forum Proposer une ressource Autres sites

EDUCASOURCE Accueil

Le site FAQ Questions juridiques Nous contacter

[Modifier la recherche]

Résultats (14 ressources)

Recherche sur : "Mathématiques et sciences expérimentales > Mathématiques > Géométrie"

- 1. MENRT : Faire des mathématiques avec l'ordinateur au collège (tome 2)**
Faire des mathématiques au collège présente des séquences pédagogiques utilisant l'outil informatique pour les classes de collège (fiches élève, professeur, vécu).
[http://www.ac-reims.fr/dadice/broc_men/mathscol/Sommaire%20\(1\).htm](http://www.ac-reims.fr/dadice/broc_men/mathscol/Sommaire%20(1).htm)
- 2. Ressources pédagogiques en mathématiques**
Le site éducation Réseau de Haute Savoie (Edres 74) présente les ressources en mathématiques des établissements de Haute-Savoie.
<http://www.edres74.cur-archamps.fr/ressources/maths/>
- 3. Constructions géométriques : des exercices pour le collège**
Cette page propose de suivre pas à pas une vingtaine de constructions géométriques que les élèves apprennent au collège.
<http://www.edres74.cur-archamps.fr/colleges/coboemol/maths/welcome.htm>
- 4. Démonstration en géométrie**

Évaluer la pertinence d'un ensemble de fiches

Il y a 14 fiches correspondant à votre recherche, vous pouvez parcourir / stocker celles qui vous intéressent, structurer les fiches ou élargir la recherche

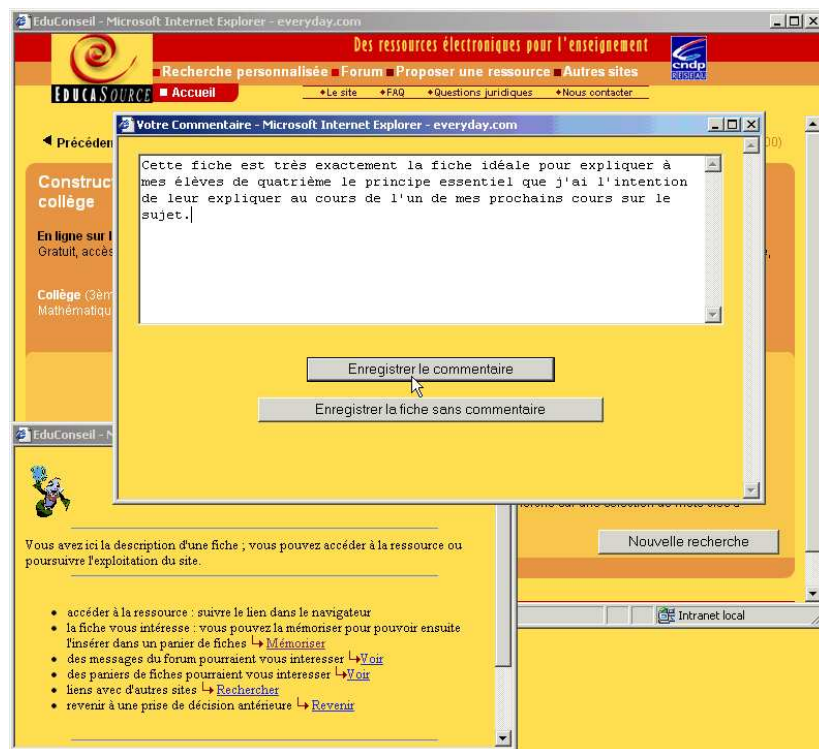
- une fiche vous intéresse particulièrement : la sélectionner dans le navigateur
- plusieurs fiches vous intéressent : vous pouvez les mémoriser pour les parcourir plus tard → [Sélectionner](#)
- il y a trop de fiches, nous pouvons vous proposer de les structurer selon différents critères → [Restructurer](#)
- il y a trop peu de fiches, nous pouvons vous proposer d'élargir la recherche selon différents critères → [Élargir](#)
- rien de pertinent, vous avez l'impression d'avoir fait fausse route : il faut remettre en question l'orientation de la recherche → [Revenir](#)
- vous désirez revenir à la page d'accueil → [Accueil](#)

Intranet local

La restructuration réduit le nombre de fiches à 14. L'une d'elles intéresse Joe, il l'affiche.



Cette fiche intéresse vraiment Joe, il la mémorise dans son espace de travail.



Joe insère un commentaire décrivant la fiche qu'il mémorise.



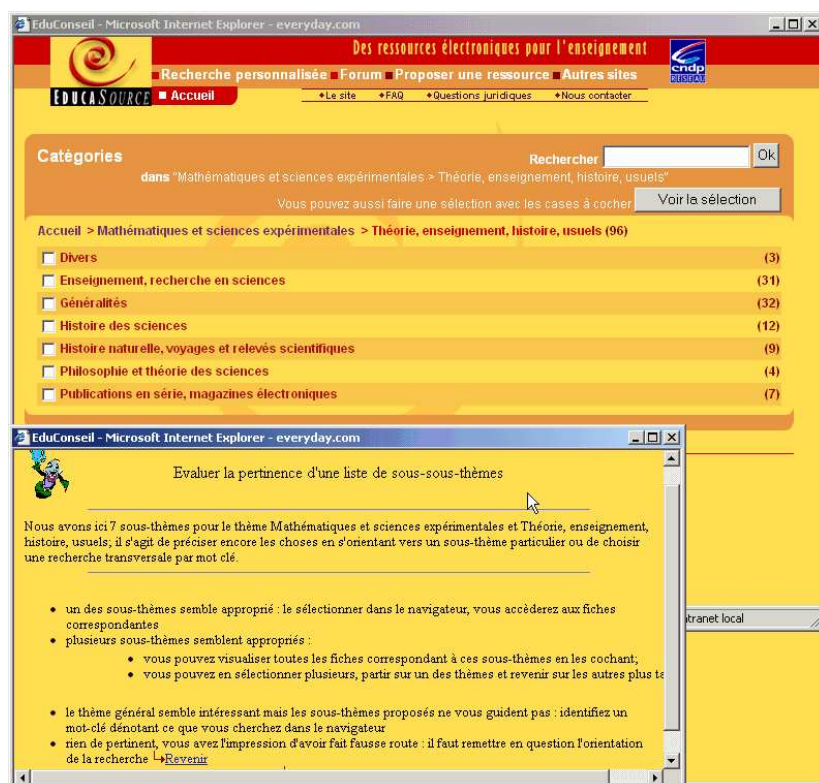
Le système conseiller indique à Joe qu'un des messages du Forum semble également intéressant.



Joe suit ce lien et accède à l'article du Forum. Joe veut maintenant revenir sur le choix précédent du sous-thème mémorisé par le système.



Le système conseiller lui affiche la liste des sous-thèmes qu'il avait sélectionné Joe peut maintenant sélectionner le second sous-thème qu'il avait choisi : "théorie, enseignement, histoire, usuels".



Et le parcours de Joe reprend où il l'avait interrompu.
Etc, etc.

Bibliographie

- [AAM01] S. AGUZZOLI, P. AVESANI et P. MASSA : Compositional recommender systems using case-based reasoning approach. *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, Radisson Hotel New Orleans, LA - USA, 2001.
- [ABHV04] L. AROYO, P. De BRA, G.J. HOUBEN et R. VDOVJAK : Embedding information retrieval in adaptive hypermedia: Ir meets aha! *Hypermedia*, 10(1):53–76, 2004.
- [ACT01] L. ARDISSONO, L. CONSOLE et I. TORRE : An adaptive system for the personalized access to news. *AI Communications*, 4(3): 129–147, 2001.
- [ADF⁺02] F. ABBATTISTA, M. DEGEMMIS, N. FANIZZI, O. LICCHELLI, P. LOPS, G. SEMERARO et F. ZAMBETTA : Learning user profiles for content-based filtering in e-commerce. *In Proceedings AI*AI Workshop su Apprendimento Automatico: Metodi e Applicazioni.*, Sienna, Italy, 2002.
- [Ama] Amazon. <http://www.amazon.com/>.
- [Apa] The apache foundation. <http://www.apache.org/>.
- [ARD05] R. ANDONIE, J.E. RUSSO et R. DEAN : Crossing the rubicon for an intelligent advisor. *Beyond Personalization 2005*, pages 7–12, 2005.
- [AT06] F. AMADIEU et A. TRICOT : Utilisation d’un hypermédia et apprentissage : deux activités concurrentes ou complémentaires. *Psychologie Française*, pages 5–23, 2006.
- [BAB⁺03] P. De BRA, A. AERTS, B. BERDEN, B. de LANGE, B. ROUSSEAU, T. SANTIC, D. SMITS et N. STASH : Aha! the adaptive hypermedia architecture. *In HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 81–84, New York, NY, USA, 2003. ACM Press.
- [BB04] P. BRUSILOVSKY et P. De BRA : Introduction. *New Review of Hypermedia and Multimedia*, 10:1–3, 2004.

- [BBS03] P. BRUSILOVSKY, P. De BRA et T. SANTIC : A flexible layout model for a web-based adaptive hypermedia architecture. *Proceedings of AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems*, pages 77–86, 2003.
- [BC98] P. De BRA et L. CALVI : 2l670: a flexible adaptive hypertext courseware system. In *HYPERTEXT '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems*, pages 283–284, New York, NY, USA, 1998. ACM Press.
- [BCGP02] P. BUONO, M. F. COSTABILE, S. GUIDA et A. PICCINNO : Integrating user data and collaborative filtering in a web recommendation system. In *Revised Papers from the nternational Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 315–321, London, UK, 2002. Springer-Verlag.
- [BEBH01] C. BAILEY, S. R. EL-BELTAGY et W. HALL : Link augmentation: A context-based approach to support adaptive hypermedia. In *OHS-7/SC-3/AH-3*, pages 239–251, 2001.
- [Bel00] N.J. BELKIN : Helping people find whate they don’t know. *Communications of the ACM*, 43:58–61, 2000.
- [BES97] P. BRUSILOVSKY, J. EKLUND et E. W. SCHWARZ : Adaptive navigation support in educational hypermedia on the world wide web. In *INTERACT*, pages 278–285, 1997.
- [BG02] D. BRICKLEY et R.V. GUHA : Rdfs: Resource description framework schema. *W3C Working Draft*, November 2002.
- [BHY96] R. D. BURKE, K. J. HAMMOND et B. C. YOUNG : Knowledge-based navigation of complex information spaces. In *Proceedings of the American Association for Artificial Intelligence Conference. Madison, WI:AAAI Press*, pages 132–139, 1996.
- [BKM⁺00] A. BRODER, R. KUMAR, F. MAGHOUL, P. RAGHAVAN, S. RAJAGOPALAN, R. STATA, A. TOMKINS et J. WIENER : Graph structure in the web. In *Proceedings of the 9th international World Wide Web conference on Computer networks: the international journal of computer and telecommunications netowrking*, pages 309–320, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.
- [BL98] P. BRUSILOVSKY et L.PESIN : Adaptive navigation support in educational hypermedia: An evaluation of the isis-tutor. *Journal of Computing and Information Technology*, 6(1):27–38, 1998.

- [BL00] J. BORGES et M. LEVENE : Data mining of user navigation patterns. In *WEBKDD '99: Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, pages 92–111, London, UK, 2000. Springer-Verlag.
- [BM02] P. BRUSILOVSKY et M. T. MAYBURY : From adaptive hypermedia to the adaptive web. *Commun. ACM*, 45(5):30–33, 2002.
- [BPAG05] J. BLANCHARD, B. PETITJEAN, T. ARTIÈRES et P. GALLINARI : Un système d’aide à la navigation dans des hypermédias. In *EGC 2005*, pages 281–292, Paris, janvier 2005.
- [Bru96] P. BRUSILOVSKY : Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.
- [Bru01] P. BRUSILOVSKY : Adaptive hypermedia. *Adaptive hypermedia. User Modeling and User-Adapted Interaction*, 11:87–110, 2001.
- [Bru04] P. BRUSILOVSKY : Adaptive navigation support: From adaptive hypermedia to the adaptive web and beyond. 2(1):7–23, 2004.
- [BS97] M. BALABANOVIC et Y. SHOHAM : Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
- [BSS06] P. De BRA, D. SMITS et N. STASH : The design of aha! In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 171–195, New York, NY, USA, 2006. ACM Press.
- [Bur99] R. BURKE : Integrating knowledge-based and collaborative-filtering recommender systems. In *Proceedings of the Workshop on AI and Electronic Commerce. AAAI 99*, 1999.
- [Bur02a] R. BURKE : Hybrid recommender systems: Survey and experiments. *User Modelling and user User-Adapted Interaction*, 12:331–370, 2002.
- [Bur02b] R. BURKE : Interactive critiquing for catalog navigation in e-commerce. *Artif. Intell. Rev.*, 18(3-4):245–267, 2002.
- [CCS05] C.C. CHEN, M.C. CHEN et Y. SUN : Pva: a self-adaptive personal view agent system. *ACM SIGKDD'01*, pages 257–262, 2005.
- [CGM⁺99] M. CLAYPOOL, A. GOKHALE, T. MIRANDA, P. MURNIKOV, D. NETES et M. SARTIN : Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*, 1999.
- [Che05] A. CHEVALIER : Evaluer un site web : les concepteurs et les utilisateurs parviennent-ils à identifier les problèmes d’utilisabilité?

- RSTI-RIA, Revue des Sciences et Technologies de l'Information*, 19/2005:319–337, 2005.
- [CJ03] M. CHEVALIER et C. JULIEN : Interface adaptative et coopérative pour l'aide à la Recherche d'Information sur le Web . *Information - Interaction - Intelligence*, Volume 3(2):47–73, 2003. ISBN 2-85428-614-6.
- [CLWB01] M. CLAYPOOL, P. LE, M. WASED et D. BROWN : Implicit interest indicators. *In IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, 2001. ACM Press.
- [CMS99] R. COOLEY, B. MOBASHER et J. SRIVASTAVA : Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.
- [Com] Comscore. <http://www.comscore.com/>.
- [CPM02] P. A. CHAMPIN, Y. PRIÉ et A. MILLE : Une approche fondée sur les usages pour l'assistance à l'utilisateur sur le web sémantique. *Actes de la conference Francophone de Reconnaissance de Formes et d'Intelligence Artificielle*, pages 633–642, 2002.
- [CPM03] P. A. CHAMPIN, Y. PRIÉ et A. MILLE : Musette: Modelling uses and tasks for tracing experience. *WS5: From Structured Cases to Unstructured Problem Solving Episodes For Experience-Based Assistance at ICCBR'03*, pages 279–294, 2003.
- [CRP00] M. CRAMPES, S. RANWEZ et M. PLANTIER : Ontology-supported and ontology-driven conceptual navigation on the world wide web. *In Proceedings of HyperText 2000, ACM HT2000*, pages 191–199. ACM Press, 2000.
- [CV95] C. CORTES et V. VAPNIK : Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [CVG06] A. CHEVALIER et L. VUILLEMIN-GIOARÀ : Aide à la prise en compte de contraintes liées aux utilisateurs lors de la conception d'un hypermédia : une étude auprès de concepteurs de sites web - supporting to consider user constraints during hypermedia design: a study with web designers. *Psychologie Française*, pages 51–72, 2006.
- [Dek05] E. DEKOVEN : Off-topic recommendations. *Beyond Personalization 2005*, pages 31–36, 2005.
- [Dmo] dmoz open directory project, project home page <http://dmoz.org>. <http://dmoz.org>.
- [EB98] J. EKLUND et P. BRUSILOVSKY : The value of adaptivity in hypermedia learning environments: A short review of empirical

- evidence. In *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia*, pages 11–17, 1998.
- [Edu] Educasource. <http://www.educasource.education.fr>.
- [ELPV04] M. EIRINAKI, C. LAMPOS, S. PAULAKIS et M. VAZIRGIANNIS : Web personalization integrating content semantics and navigational patterns. In *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 72–79, New York, NY, USA, 2004. ACM Press.
- [EV03] M. EIRINAKI et M. VAZIRGIANNIS : Web mining for web personalization. *ACM Trans. Inter. Tech.*, 3(1):1–27, 2003.
- [EVV03] M. EIRINAKI, M. VAZIRGIANNIS et I. VARLAMIS : Sewep: using site semantics and a taxonomy to enhance the web personalization process. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, New York, NY, USA, 2003. ACM Press.
- [EZ03] E. EGYED-ZSIGMOND : *Gestion des connaissances dans une base de documents multimédias*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 2003.
- [FBH00] X. FU, J. BUDZIK et K.J. HAMMOND : Mining navigation history for recommendation. *Proceedings of Intelligent User Interface*, pages 106–112, 2000.
- [FCS01] Y. FU, M. CREADO et M. SHIH : Adaptive web sites by web usage mining. In *IC'2001: Proceedings of the International Conference on Internet Computing*, pages 28–34, Las Vegas, NA, USA, 2001.
- [FRFMS00] L. FRANCISCO-REVILLA et III F. M. SHIPMAN : Adaptive medical information delivery combining user, task and situation models. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 94–97, New York, NY, USA, 2000. ACM Press.
- [GFMG05] M. GRGAR, B. FORTUNA, D. MLADENIC et M. GROBELNIK : knn versus svm in the collaborative filtering framework. pages 251–260, Chicago, IL, août 2005.
- [GI03] S. GARLATTI et S. IKSAL : A semantic web approach for adaptive hypermedia. *Proceedings of AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems*, pages 81–85, 2003.
- [GIK99] S. GARLATTI, S. IKSAL et P. KERVELLA : Adaptive on-line information system by means of a task model and spatial views.

- In Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, pages 59–66, Washington, DC, USA, 1999. P. Brusilovsky and P. D. Bra, Computer Science Report, Report No. 99-07.
- [GMV99] N. GUARINO, C. MASOLO et G. VETERE : OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems*, 14(3):70–80, 1999.
- [GNOT92] D. GOLDBERG, D. NICHOLS, B. M. OKI et D. TERRY : Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.
- [GPPG95] S. GIROUX, F. PACHET, G. PAQUETTE et J. GIRARD : Des systèmes conseillers épiphytes. *Revue d'intelligence artificielle*, 9(2):165–190, 1995.
- [GR04] N. GOLOVIN et E. RAHM : Reinforcement learning architecture for web recommendations. *In ITCC '04: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2*, pages 398–402, Washington, DC, USA, 2004. IEEE Computer Society.
- [HCOC02] Z. HUANG, W. CHUNG, T.H. ONG et H. CHEN : A graph-based recommender system for digital library. *In JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65–73, New York, NY, USA, 2002. ACM Press.
- [Her02] E. HERDER : Metrics for the adaptation of site structure. *In N. HENZE, éditeur: Proceedings LLA-02 Lehren - Lernen - Adaptivitat, ABIS Workshop*, pages 22–25, Hannover, Germany, octobre 2002.
- [Her03] E. HERDER : Revisitation patterns and disorientation. *In A. HOTH et G. STUMME, éditeurs: German Workshop on Adaptivity and User Modeling in Interactive Systems ABIS03*, pages 291–293, Karlsruhe, octobre 2003.
- [HFM04] J. HERAUD, L. FRANCE et A. MILLE : Pixed: an its that guides students with the help of learners' interaction log. *In International conference on intelligent tutoring systems (workshop analyzing student-tutor interaction logs to improve educational outcomes)*, pages 57–64, Maceio, 2004.
- [HJ04] E. HERDER et I. JUVINA : Discovery of individual user navigation styles. *In Workshop on Individual Differences at AH2004*, pages 40–49, août 2004.
- [HKR00] J. L. HERLOCKER, J. A. KONSTAN et J. RIEDL : Explaining collaborative filtering recommendations. *In CSCW '00: Pro-*

- ceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, New York, NY, USA, 2000. ACM Press.
- [HM02] J. HERAUD et A. MILLE : Pixed: towards the sharing and the re-use of experience to assist training. *In World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 777–782, Denver, Colorado, USA, 2002.
- [HN01] N. HENZE et W. NEJDL : Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education*, 12: 325–350, 2001.
- [HSRF95] W. HILL, L. STEAD, M. ROSENSTEIN et G. FURNAS : Recommending and evaluating choices in a virtual community of use. *In CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [HvD04] E. HERDER et E.M.A.G. van DIJK : Site structure and user navigation: Models, measures and methods. *In S.Y. CHEN et G.D. MAGOULAS, éditeurs : Adaptable and Adaptive Hypermedia Systems*, pages 19–33. Idea Group Publishing, 2004.
- [HWOM06] E. HERDER, H. WEINREICH, H. OBENDORF et M. MAYER : Much to know about history. *In Adaptive Hypermedia 2006*, pages 283–287. ACM Press, 6 2006.
- [ILA95] D. J. ITTNER, D. D. LEWIS et D. D. AHN : Text categorization of low quality images. *In 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 301–315, ISRI, University of Nevada, Las Vegas, 1995.
- [Jav] Java. <http://java.sun.com/>.
- [JFM97] T. JOACHIMS, D. FREITAG et T. M. MITCHELL : Webwatcher: A tour guide for the world wide web. *In IJCAI (1)*, pages 770–777, 1997.
- [JGr] Jgraph : Java graph visualization and layout. <http://www.jgraph.com/>.
- [JH05] I. JUVINA et E. HERDER : The impact of link suggestions on user navigation and user perception. *UM2005 User Modeling: Proceedings of the Tenth International Conference*, pages 483–492, 2005.
- [JO06] I. JUVINA et H. Van OOSTENDORP : Individual differences and behavioral metrics involved in modeling web navigation. *Univers. Access Inf. Soc.*, 4(3):258–269, 2006.

- [JT98] M. JACZYNSKI et B. TROUSSE : Broadway, a case-based browsing advisor for the web. *In ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 697–698, London, UK, 1998. Springer-Verlag.
- [JvO05] I. JUVINA et H. van OOSTENDORP : Bringing cognitive models into the domain of web accessibility. *Proceedings of HCI2005 Conference, Las Vegas, USA*, 2005.
- [KBP00] M. KITAJIMA, M. H. BLACKMON et P.G. POLSON : A comprehension-based model of web navigation and its application to web usability analysis. *S. McDonald and Y. Waern and G. Cockton (eds.), People and Computers XIV - Usability or Else! (Proceedings of HCI 2000)*, Springer, pages 357–373, 2000.
- [KGX⁺03] M. KILFOIL, A. GHORBANI, W. XING, Z. LEI, J. LU, J. ZHANG et X. XU : Toward an adaptive web: The state of the art and science. *In Proc. of the 1st Annual Conference on Communication Networks & Services Research (CNSR 2003)*, pages 119–130, Moncton, Canada, 2003.
- [KKP01] A. KOBASA, J. KOENEMANN et W. POHL : Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16:111–155, 2001.
- [LAR02] W. LIN, S. ALVAREZ et C. RUIZ : Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6:83–105, 2002.
- [LBW00] A. W. LAZONDER, H. J. A. BIEMANS et I. G. J. H. WOPEREIS : Differences between novice and experienced users in searching information on the world wide web. *J. Am. Soc. Inf. Sci.*, 51(6): 576–581, 2000.
- [LCPM05] J. LAFLAQUIÈRE, P. A. CHAMPIN, Y. PRIÉ et A. MILLE : Approche de modélisation de l'expérience d'utilisation de systèmes complexes pour l'assistance aux tâches de veille informatique-médiées. *ISKO'France 2005 Organisation des connaissances dans les systèmes d'informations orientés utilisation: Contexte de veille et d'intelligence économique*, page 17, 2005.
- [Lie95] H. LIEBERMAN : Letizia: An agent that assists web browsing. *In Chris S. MELLISH, éditeur: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

- [LLS02] H. LIU, H. LIEBERMAN et T. SELKER : Goose: A goal-oriented search engine with commonsense. *In AH '02: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 253–263, London, UK, 2002. Springer-Verlag.
- [LRS01] H. LIEBERMAN, E. ROSENZWEIG et P. SINGH : Aria: An agent for annotating and retrieving images. *Computer*, 34(7):57–62, 2001.
- [MARS02] S. E. MIDDLETON, H. ALANI, D. C. De ROURE et N. R. SHADBOLT : Exploiting synergy between ontologies and recommender systems. *Proceedings of the Eleventh International WWW conference, Semantic Web, Workshop, May 2002, Hawaii, USA*, 2002.
- [MCS00] B. MOBASHER, R. COOLEY et J. SRIVASTAVA : Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, 2000.
- [MGT⁺87] T. W. MALONE, K. R. GRANT, F. A. TURBAK, S. A. BROBST et M. D. COHEN : Intelligent information-sharing systems. *Commun. ACM*, 30(5):390–402, 1987.
- [Mic02] C. MICHEL : Caractérisation d’usages et personnalisation d’un portail pédagogique. état de l’art et expérimentation de différentes méthodes d’analyse du web usage mining. *7ème colloque de l’AIM « Affaire Électronique et Société de Savoir : Opportunités et Défis »*, 29 mai-1er juin 2002, 2002.
- [MLR03] M. MONTANER, B. LOPEZ et J. L. De La ROSA : A taxonomy of recommender agents on the internet. *Artif. Intell. Rev.*, 19(4):285–330, 2003.
- [MMN01] P. MELVILLE, R. J. MOONEY et R. NAGARAJAN : Content-boosted collaborative filtering. *Proceedings of the SIGIR-2001 Workshop on Recommender Systems*, 2001.
- [MPT99] F. MASSEGLIA, P. PONCELET et M. TEISSEIRE : Using data mining techniques on web access logs to dynamically improve hypertext structure. *SIGWEB Newsl.*, 8(3):13–19, 1999.
- [MR00] R. J. MOONEY et L. ROY : Content-based book recommending using learning for text categorization. *In Proceedings of DL-00, 5th ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US.
- [MRS01] S. E. MIDDLETON, D. C. De ROURE et N. R. SHADBOLT : Capturing knowledge of user preferences: ontologies in recommender systems. *In K-CAP '01: Proceedings of the 1st international*

- conference on Knowledge capture, pages 100–107, New York, NY, USA, 2001. ACM Press.
- [MRS02] S. E. MIDDLETON, D. C. De ROURE et N. R. SHADBOLT : Foxtrot recommender system: User profiling, ontologies and the world wide web. *In Proceedings International WWW Conference(11)*, 2002.
- [MTCGdGP03] A. A. MACEDO, K. N. TRUONG, J. A. CAMACHO-GUERRERO et M. da GRAÇA PIMENTEL : Automatically sharing web experiences through a hyperdocument recommender system. *In HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 48–56, New York, NY, USA, 2003. ACM Press.
- [Net] Netcraft company. <http://news.netcraft.com/>.
- [Nie05] J. NIELSEN : Usability for the masses. *Journal of usability studies, Issue 1, Vol. 1, November 2005*, pages 2–3, 2005.
- [Paz99] M.J. PAZZANI : A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408, 1999.
- [PB97] M.J. PAZZANI et D. BILLSUS : Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.
- [PB02] M.J. PAZZANI et D. BILLSUS : Adaptive web site agents. *Autonomous Agents and Multi-Agent Systems*, 5:205–218, June 2002.
- [PE98] M. PERKOWITZ et O. ETZIONI : Adaptive web sites: Automatically synthesizing web pages. *In AAAI/IAAI*, pages 727–732, 1998.
- [pro] protege. <http://protege.stanford.edu/>.
- [PT99] G. PAQUETTE et P. TCHOUNIKINE : Une approche méthodologique pour la construction de systèmes conseillers. *In Journées Ingénierie des Connaissances (IC'99)*, pages 1–12, 1999.
- [PT02] G. PAQUETTE et P. TCHOUNIKINE : Contribution à l'ingénierie des systèmes conseillers : une approche méthodologique fondée sur l'analyse du modèle de la tâche. *Revue Sciences et Techniques Educatives*, 9(3-4):409–435, 2002.
- [RAC⁺02] A. M. RASHID, I. ALBERT, D. COSLEY, S. K. LAM, S. M. MCNEE, J. A. KONSTAN et J. RIEDL : Getting to know you: learning new user preferences in recommender systems. *In IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134, New York, NY, USA, 2002. ACM Press.

- [RaM] Research and markets. <http://www.researchandmarkets.com/>.
- [Rau92] M. RAUTERBERG : A method of a quantitative measurement of cognitive complexity. *G.C. van der Veer, M.J. Tauber, S. Bagnara and A. Antalovits, Eds., Human-Computer Interaction: Tasks and Organisation*, pages 295–307, 1992.
- [RIS⁺94] P. RESNICK, N. IACOVOU, M. SUCHAK, P. BERGSTROM et J. RIEDL : GroupLens: an open architecture for collaborative filtering of netnews. *In CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM Press.
- [Roc71] J. J. ROCCHIO : Relevance feedback in information retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323, 1971.
- [RT04a] B. RICHARD et P. TCHOUNIKINE : Une approche centrée modèle pour la construction d'un système conseiller pour un site web. *In Actes de IC 2004*, pages 151–162, 2004.
- [RT04b] B. RICHARD et P. TCHOUNIKINE : Enhancing the adaptivity of an existing website with an epiphyte recommender system. *New Review of hypermedia, Adaptive Hypermedia in the Age of the Adaptive Web, Volume 10 Number 1*, pages 31–52, 2004.
- [RV97] P. RESNICK et H. R. VARIAN : Recommender systems. *Communications of the ACM*, 40(3), pages 56–58, 1997.
- [SCDT00] J. SRIVASTAVA, J. COOLEY, M. DESHPANDE et P. N. TAN : Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.
- [SKR99] J. Ben SCHAFER, J. A. KONSTAN et J. RIEDL : Recommender systems in e-commerce. *In ACM Conference on Electronic Commerce*, pages 158–166, 1999.
- [SKR04] J. Ben SCHAFER, J. A. KONSTAN et J. RIEDL : The view through metalens: Usage patterns for a meta-recommendation system. *IEE Proceedings Software*, 151(6):267–279, december 2004.
- [SKZ03] M. C. SCHRAEFEL, M. KARAM et S. ZHAO : mspace: interaction design for user-determined, adaptable domain exploration in hypermedia. *Proceedings of AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems*, pages 217–235, 2003.
- [TBB⁺04] P. TCHOUNIKINE, M. BAKER, N. BALACHEFF, M. BARON, A. Derycke A., D. GUIN, J.-F. NICAUD et P. RABARDEL : Platon-1 : quelques dimensions pour l'analyse des travaux de re-

- cherche en conception d'EIAH. *In Rapport d'Action Spécifique du CNRS*, 2004.
- [Tch02] P. TCHOUNIKINE : Quelques éléments sur la conception et l'ingénierie des EIAH. *In Actes du GDR I3*, pages 233–245, 2002.
- [TPDB98] A. TRICOT, C. PIERRE-DEMARCY et R. EL BOUSSARGHINI : Un panorama des recherches consacrées à l'étude de l'activité mentale de l'utilisateur d'un hypermédia. *Sciences et Techniques Éducatives*, 5 (4), pages 371–400, 1998.
- [TR04] A. TRICOT et J.F. ROUET : Activités de navigation dans les systèmes d'information. *J.M. Hoc and F. Darses (Eds.), Psychologie ergonomique : tendances actuelles*, Paris : PUF, pages 71–95, 2004.
- [Upe94] S. UPENDRA : Social information filtering for music recommendation. *S.M. Thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology*, 1994., 1994.
- [Vas96] J. VASSILEVA : A task-centered approach for user modeling in a hypermedia office documentation system. *User Modeling and User-Adapted Interaction*, 6(2-3):185–224, 1996.
- [VPF02] P. VIAPPIANI, P. PU et B. FALTINGS : Acquiring user preferences for personal agents. *In AAAI Fall Symposium*, pages 53–59, North Falmouth, MA, 2002. AAAI Press.
- [WOHM06] H. WEINREICH, H. OBENDORF, E. HERDER et M. MAYER : Off the beaten tracks: Exploring three aspects of web navigation. *In WWW Conference2006*, pages 133–142. ACM, ACM Press, 5 2006. Conference acceptance rate 11%, winner of the 'Best Student Paper Award'.
- [ZGH⁺05] T. ZHU, R. GREINER, G. HAUBL, B. PRICE et K. JEWELL : Behavior-based recommender systems for web content. *In IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 83–88, New York, NY, USA, 2005. ACM Press.

RÉSUMÉ :

Nos travaux s'inscrivent dans le cadre des recherches menées sur les systèmes conseillers. Les systèmes conseillers sont des systèmes dont l'objectif est de proposer une assistance aux utilisateurs de systèmes informatiques. Parmi les systèmes pour lesquels une telle aide peut être proposée, nous avons choisi de cibler les sites Web dits complexes, i.e., des sites qui présentent un nombre important de pages, une structure large et profonde et de nombreuses fonctionnalités. Ces caractéristiques en font des sites difficiles à appréhender et extrêmement riches de par les tâches qu'ils permettent de réaliser.

Nos travaux ont conduit à la proposition d'une approche fondée sur la construction de systèmes conseillers à partir d'une architecture épiphyte. Cette approche est originale de par (1) la nature de l'architecture proposée et (2) la nature de l'aide apportée.

L'architecture générique épiphyte proposée permet une séparation effective entre les données du serveur Web pour lequel une aide est proposée et les données propres au système conseiller. Cette séparation franche entre les systèmes autorise l'évolution indépendante de chacun des systèmes.

L'aide apportée aux utilisateurs est de type guidage et est fondée sur la tâche que l'utilisateur souhaite réaliser en abordant un site. Il ne s'agit donc pas de proposer une aide ponctuelle, c'est à dire, une aide associée uniquement à une page, mais plutôt une aide qui va s'étendre sur de nombreuses pages du site et lier ces pages pour que leur lecture et leur compréhension permettent la réalisation du but de l'utilisateur. Pour proposer une telle aide à l'utilisateur, le système conseiller analyse la navigation de l'utilisateur en fonction de modèles d'utilisation créés par des experts et propose les conseils associés aux modèles.

La création des modèles d'utilisation nous est apparue comme une tâche particulièrement délicate. Nous avons proposé deux approches informatiques pour assister cette tâche, la deuxième offrant une assistance plus poussée à l'expert.

Deux prototypes ont été construits. Ils nous ont permis d'illustrer et de mettre en oeuvre notre approche.

MOTS-CLÉS : système conseiller, modèle d'utilisation d'un site Web, Web.