**MASTER EN SCIENCES INFORMATIQUES**

# Etude de la qualité des informations de personnalisation web

Gilain, Olivier

*Award date:*
2011

*Awarding institution:*
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur
Faculté d'Informatique
Année académique 2010 - 2011

# Etude de la Qualité des Informations de Personnalisation Web

Olivier Gilain


Mémoire présenté en vue de l'obtention du grade de Master en Informatique

# Abstract

This master's thesis is carrying out a study on the quality of web information in the context of web personalization. User's knowledge are summarized in a structure called "user model". Based on the information contained in this user model, a web personalization system will adjust the display of web pages in content as well as form. However, such information is not always trustworthy. For this, we are thinking about the qualities specific to web information. This allows us to identify quality criteria. Then we learn how, with the information we have, we can firstly estimate them and secondly merge them to obtain a "confidence index". It is computed for each information present in the user model. This will allow a web personalized system to receive an indication about the reliability of information. Then, it will weigh the knowledge from which it will build personalized web pages.

**Keywords :** Information quality, user models, web personalization, personalized website, quality evaluation, confidence index, information trustness, freshness, correctness

# Résumé

Ce mémoire réalise une étude sur les qualités des informations web s'inscrivant dans le cadre de la personnalisation web. Les connaissances au sujet d'un utilisateur sont regroupées dans une structure que l'on appelle "modèle utilisateur". Sur base des informations contenues dans ce modèle, un système de personnalisation web va adapter l'affichage des pages web tant dans le contenu que dans la forme. Toutefois, ces informations ne sont pas toujours dignes de confiance. Pour cela, nous menons une réflexion sur les qualités inhérentes aux informations web. Cela nous permet de dégager des critères de qualité. Nous découvrons ensuite comment, au vu des informations que nous possédons, nous pouvons d'une part les estimer et d'autre part les agréger afin d'obtenir un indice dit de " confiance ". Il est calculé pour chaque information présente dans le modèle utilisateur et permettra à un système de personnalisation web de bénéficier d'une indication quant à la fiabilité d'une information. Celle-ci viendra alors pondérer les connaissances à partir desquelles il va construire les pages web personnalisées.

**Mots-clés :** Qualité des informations, modèles utilisateurs, personnalisation web, site web personnalisé, évaluation de la qualité, indice de confiance, confiance des informations, freshness, correctness

# Avant-propos

Ce mémoire est le fruit d'un investissement personnel important et a mobilisé une grande partie des compétences développées durant mon cursus universitaire aux Facultés Universitaires Notre-Dame de la Paix à Namur. Pour cela je tiens tout d'abord à remercier l'ensemble du corps professoral de cette institution ainsi que mes collègues étudiants pour l'apprentissage mutuel retiré de nos collaborations.

Je tiens également à remercier M. Luc Ponsard, administrateur de Defimedia s.a. et maître du stage ayant servi de base aux réflexions faites au cours de ce mémoire. M. Ponsard m'a donné l'accès à des ressources clés pour ce travail et m'a accompagné tout au long de ma formation chez Defimedia s.a.. A ce titre, je voudrais aussi adresser mes remerciements à M. Jean-Bernard Collet, collaborateur de l'entreprise, pour son encadrement instructif et constructif. Ma gratitude va aussi vers le personnel de Defimedia s.a. pour son enthousiasme, son aide et sa sympathie.

Bien sûr et surtout, je n'aurais pas pû réaliser ce mémoire sans l'aide inestimable de mon promoteur, M. Philippe Thiran. Je lui dédie sincèrement toute ma gratitude pour sa disponibilité, son accompagnement et ses nombreux conseils et relectures lors de mes recherches.

Je remercie également les membres de ma famille ainsi qu'Alice Fontaine pour le temps passé aux relectures de ce travail.

Enfin, je remercie mes lecteurs et le jury de la faculté d'informatique des FUNDP pour l'intérêt manifesté à l'égard de mon sujet et leur souhaite bonne lecture.

# Table des matières

# Chapitre 1

# Introduction

## 1.1 Contexte et sujet

« Information », voilà le mot-clé de la société actuelle. Tout ou presque est information. Depuis l'arrivée de l'informatique et des réseaux, cette philosophie informationnelle n'a fait que s'amplifier. Dans cette informatisation sans frontière, comment nous, êtres humains, pouvons-nous garder le contrôle ? Comment gérer cette surcharge informationnelle venant de tous horizons ? Peut-on mettre en place des mécanismes permettant de traiter ces données efficacement, d'évaluer leur pertinence et d'en tirer des conclusions par rapport à un contexte quelconque ? En somme, comment peut-on procéder pour évaluer la qualité des informations collectées ? Des études ont déjà été réalisées sur l'évaluation de la qualité des données dans différents domaines comme les systèmes coopératifs [5, 27, 29] ou les bases de données [46, 12, 49]. Néanmoins, bien que prônée et requise dans de nombreux domaines, la qualité n'a pas, jusqu'à ce jour, été définie stricto-sensu. Les scientifiques s'accordent cependant sur le fait qu'elle peut se décomposer en dimensions ou critères [40, 52, 44, 36] parmi lesquels on retrouve fréquemment la fiabilité, la complétude et l'exactitude des informations. Ces critères doivent être définis en fonction du contexte de recherche de qualité. Dans ce mémoire, la qualité des informations sera essentiellement étudiée dans le contexte du web. Plus spécifiquement encore, dans le cadre de la personnalisation web.

La personnalisation web s'inscrit dans le cadre de la contextualisation des données. En effet elle a pour objet de modeler l'affichage du contenu web en fonction des intérêts, préférences de l'internaute. Ceci va permettre de discriminer les utilisateurs en fonction de leurs besoins spécifiques. Son principe est en somme assez simple : les informations sociodémographiques ou comportementales recueillies

concernant l'utilisateur sont regroupées sous forme de profil. Le contenu est ensuite adapté sur base de celui-ci. Comme nous le verrons, sa réalisation est une autre paire de manches. Au final, son but ultime n'est autre que la satisfaction des visiteurs du site web dont il est question, bien qu'en pratique, la personnalisation web est couramment utilisée à des fins de marketing.

## 1.2 Enjeux

La personnalisation d'un site internet offre une multitude de possibilités. D'une manière générale, elle va permettre d'enrichir les informations proposées à l'internaute en sélectionnant le contenu qui lui est le plus adapté.

Concrètement, supposons que le système a déduit que le visiteur est fan d'automobiles. Pour remplir une section consacrée à la publicité, une bannière publicitaire offrant des conditions intéressantes sur des voitures sera préférée à celle proposant des vélos. Le système va donc générer la page avec cette première, proposant dés lors, un contenu personnalisé. Imaginons que ce site propose sur sa page d'accueil les dernières actualités pour différentes thématiques. Il serait alors également pertinent d'afficher principalement les articles concernant le monde de l'automobile, ou tout le moins, augmenter la visibilité de ceux-ci. Pareillement, sur un site d'e-commerce, proposer les produits les plus attrayants pour l'utilisateur - autrement dit le consommateur potentiel - représente un réel atout du point de vue marketing. Toujours dans l'optique business, certains systèmes de personnalisation vont même jusqu'à utiliser les données recueillies afin de générer des rapports, des statistiques et généralisations. Par exemple, ils permettent de dégager des tendances sur des questions du style « quel est l'impact de proposer tel contenu de telle couleur à tel endroit ». Ensuite, sur base de ces tendances, des études sont réalisées afin de promouvoir les produits ou services proposés. Voilà grosso-modo les grandes applications qui se font à l'heure actuelle sur les sites d'e-business proposant de la personnalisation web.

Après avoir fait un tour d'horizon des intérêts pour le marketing, voyons ce que l'utilisateur, premier concerné, peut en retirer. Quel est l'intérêt de naviguer sur un site proposant du contenu adapté ? La valeur ajoutée de ce système pour un internaute se résume en quatre aspects. D'abord, l'affichage de contenus pertinents. Ensuite, la recommandation de contenus. Après, des résultats aux requêtes

(formulées par l'internaute) affinés. Autrement dit, lorsqu'une recherche est effectuée, la liste de résultats proposée est raffinée grâce aux caractéristiques recensées contenues dans son profil. Enfin, une adaptation de l'interface prenant en compte les goûts et préférences de celui-ci permettant d'améliorer la navigation sur le plan visuel et interactionnel. Bref, tant pour le gestionnaire du site que pour le visiteur, les atouts du système de personnalisation sont nombreux et bien réels. Sur cette base, étudier la qualité des informations jouant un rôle tout au long de ce processus prend tout son sens. Les enjeux sont clairs, les informations sont nombreuses et de fiabilité variable. C'est dans ce cadre que l'étude réalisée dans ce mémoire va s'inscrire.

## 1.3 Problèmes et motivations

Au premier abord, mener une réflexion sur la qualité des informations qui entrent en jeu dans un système de personnalisation web peut sembler trivial. Néanmoins, comme cité précédemment, nous sommes face à une absence de consensus sur la notion de qualité. Bien que, en pratique, la plupart des scientifiques admettent sans difficultés que ce concept ne réside pas tant sur une définition unanime, mais effectivement sur un ensemble de dimensions, critères, facteurs ou attributs [40, 52, 44, 36]. Afin d'adopter une démarche précise et sans ambigüité, nous devrons donc, dés le départ, proposer une agrégation de critères sur lesquels, toutes notions de qualité (dans notre processus) reposera.

Contrairement à ce qu'on pourrait laisser croire, les principales difficultés de ce travail concernent d'autant plus le concept d' « informations » et tout ce qui tourne autour (comme sa nature, ses sources et ses processus d'acquisition) que de l'interprétation même de celui de qualité.

En effet, intéressons nous dans un premier temps à la nature d'une donnée afin de mieux cerner le problème. Les données (ou bien informations dans ce contexte) sont de plusieurs types. On recense par exemple les données personnelles, données financières, données contextuelles (ou d'environnement), centre d'intérêts, préférences, etc. On comprend bien que la façon de les traiter, de les utiliser, et a posteriori, d'évaluer leur qualité va différer selon leur « famille ». Par exemple, l'attribut « date de naissance » d'un individu n'est pas traité, utilisé et évalué de la même manière qu'un attribut « résolution d'écran ». De plus, leurs relations au temps n'est pas du tout la même. Alors que la valeur « date de naissance » est fixe, c'est-à-dire, toujours vérifiée quelque soit l'instant considéré, la résolution

d'écran quant à elle peut varier suite notamment à un changement d'ordinateur. De même l'adresse postale, information ayant une longévité moyennement longue, probablement de l'ordre de plusieurs années, perd malgré tout, au gré du temps de son potentiel informatif, de sa « pertinence ». En effet, entre la date d'obtention de la dite adresse et la date d'utilisation, l'internaute peut avoir déménagé sans l'avoir indiqué au système. Dés lors, attribuer une confiance [1] aveugle aux données détenues par le site n'est très certainement pas une bonne approche. La prise en compte de la dépréciation temporelle des caractéristiques des visiteurs en l'injectant dans le modèle de qualité sera donc également une difficulté supplémentaire.

A cela s'ajoute l'aspect provenance des informations. Sur le web, elles sont tantôt reçues, tantôt capturées et même parfois déduites. Typiquement, la réception, la capture et la déduction s'effectue respectivement comme suit. Un utilisateur peut compléter son profil via des formulaires web (réception). Le navigateur, quant à lui, peut fournir des données plus techniques telle que l'IP, la version du navigateur, le système d'exploitation, etc (capture). Il est également possible de déduire les préférences et intérêts du visiteur en analysant son interaction avec le site (déduction). De ce fait, on distingue clairement une multitude de sources de données. Chacune d'entre elles fournit des informations avec plus ou moins de pertinence. A priori, il serait logique de considérer les infos provenant de l'internaute-même comme fiables. Cependant, cela n'est pas toujours le cas. Une faute d'orthographe ou d'inattention peut facilement se glisser lorsque l'utilisateur complète le formulaire. De plus, en quoi peut-on accorder totale confiance à l'internaute ? Pareillement, la crédibilité accordée à un centre d'intérêt quelconque obtenu par l'analyse du comportement de l'internaute lors de sa navigation sur le site ne doit pas être maximale. En effet, cette dernière méthode n'est pas la plus « sûre ». Un comportement peut être l'objet de plusieurs déductions, ce qui peut provoquer une certaine confusion. En outre, il est naturel qu'un utilisateur clique sur du contenu qui ne l'intéresse pas, tout simplement par soucis d'exploration ou bien simplement par erreur. Il faudra donc veiller à ne pas tirer de conclusions hâtives et à outrance. La difficulté sera donc ici de porter une attention spécifique à chaque source informationnelle tout en gardant à l'esprit que leur fiabilité risque de ne pas être identique.

Comme nous venons de le voir, il existe des procédés d'analyse comportementale permettant d'extraire des informations au sujet de l'internaute. Celui-ci va effectuer une action (par exemple cliquer

1. Par "confiance" nous entendons "l'assurance que la donnée soit correcte".

sur un article), ensuite un algorithme va en déduire des caractéristiques afin de venir compléter son profil. Parfois cette technique joue sur le temps de consultation d'une page, d'un contenu, etc. C'est typiquement un processus interactif. L'expérience montre que ce genre de processus n'est pas des plus aisés.

Parallèlement à cela, on retrouve aussi un processus incrémental. De fait, le profil utilisateur ne s'édifie pas d'un seul trait, mais se bâtit progressivement, au rythme de la réception des informations. Un profil n'est jamais « achevé ». Il est en constante évolution. Une caractéristique valable il y a quelque mois ne l'est peut-être plus actuellement. Par ailleurs, de par l'aspect multi-sources, des contradictions entre les données peuvent apparaitre. Plus qu'un « simple » processus incrémental, traiter les infos relève clairement d'un processus d'apprentissage !

Réussir à concilier les différentes difficultés invoquées jusqu'ici apparait comme un réel challenge et constitue un travail de longue haleine. De ce fait, s'attaquer à l'étude de la qualité des informations dans le cadre d'un processus de personnalisation web représente donc un exercice périlleux, c'est pourquoi il fait l'objet d'un mémoire.

A l'heure actuelle, beaucoup de sites intègrent d'ores et déjà la personnalisation. Le premier auquel on pourrait penser est Google où les caractéristiques de l'utilisateur influent directement sur les résultats affichés lors d'une recherche. Citons également Amazon. Amazon est une entreprise américaine de commerce électronique. Sa spécialité est la vente de livres. Néanmoins, elle s'est diversifiée et propose toutes sortes de produits. Cette société utilise énormément la personnalisation afin de proposer des produits cohérents par rapport aux centres d'intérêts des visiteurs. Nous pourrions énumérer dans les grandes lignes les différents acteurs, mais cela ne présenterait pas de réel intérêt. Néanmoins, nous retiendrons que les systèmes de personnalisation sont déjà bien en place dans de nombreux sites. Cependant, l'étude de la qualité dans ce cadre reste un domaine inexploré.

Pourquoi ce sujet n'a-t-il jamais réellement été étudié ? La jeunesse et le manque de maturité du web et plus spécifiquement de la personnalisation constituent des éléments de réponse non-négligeables. Les possibilités de ce genre de système sont, comme nous l'avons vu, essentiellement liées au business. Or ce sujet mérite une étude approfondie au préalable et n'est donc pas directement exploitable. Pourtant, beaucoup de scientifiques s'efforcent d'étudier les données et leur qualité dans les grandes masses de données comme les datawarehouses. Il y aussi énormément de travaux réalisés dans le domaine de l'exploration de données (datamining) consistant à extraire un certain « savoir » à partir des agrégats

de données présents dans les bases de données. Cet attrait s'arrête là, et n'est pas envisagé dans ce cadre-ci.

Notons que les discussions concernant la vie privée, la confidentialité et la sécurité des données ne seront pas abordées. Ces questions sont en réalité liées à la définition d'un système de personnalisation, ce que nous n'allons pas présenter dans le présent document. Il n'est dés lors pas pertinent d'aborder ces thèmes.

## 1.4 Objectifs

Les objectifs de ce travail sont multiples :

1. Via un examen de la littérature scientifique, étudier la question de la modélisation utilisateur et en dégager un méta-modèle utilisateur.

2. Dresser un état de l'art concernant la personnalisation web et ses techniques.
3. Mener une étude spécifique au sujet de la qualité des données web collectées par le système de personnalisation web.
4. Sur base des résultats de l'étude du point 3, tenter de définir un procédé permettant d'évaluer la qualité des informations web.
5. Eventuellement, mettre en pratique les réflexions menées grâce à un prototype d'évaluation de la qualité des informations web.

## 1.5 Plan et méthodologie

Ce mémoire se découpe en différents chapitres, chacun correspondant à une étape de notre méthodologie de recherche concernant le sujet. En voici les aperçus :

Le chapitre 2 étudiera la question de la modélisation utilisateur. Nous contextualiserons et définirons d'abord le modèle utilisateur, après quoi nous en dégagerons une représentation type et étudierons le processus qui lui est lié. Nous aborderons également en fin de chapitre une seconde modélisation, la modélisation de "stéréotypes".

Le chapitre 3 proposera de poser différentes méthodes et techniques afin de définir un système de personnalisation web. Cela passera par un examen de la litérrature scientifique ainsi que par la définition d'un processus de personnalisation web. Les limites et inconvénients de la personnalisation web seront également abordés.

Le chapitre 4 va entamer la réflexion sur la qualité des informations. Nous verrons d'abord ce qui se fait dans le domaine de la qualité des informations dans la littérature scientifique. Ensuite, une étude sur les qualités des informations web collectées dans le cadre de la personnalisation web sera menée.

Le chapitre 5 portera sur la définition d'une méthode permettant l'évaluation d'un indice dit de "confiance", basée sur les aboutissements du chapitre précédent.

Le chapitre 6 proposera une application permettant la simulation de l'alimentation d'un modèle utilisateur par des données concernant un internaute, et, pour chacunes d'elles, le calcul de l'indice de confiance défini au chapitre précédent. Ce prototype viendra mettre en pratique les réflexions menées tout au long de ce travail. Des pistes de validation seront ensuite proposées.

Le chapitre 7 présentera les conclusions de ce mémoire et les perspectives qui y sont liées.

# Chapitre 2

# Le Modèle Utilisateur

Pour faire simple, un modèle utilisateur est un ensemble de connaissances au sujet d'un utilisateur pouvant être utiles pour un système. Nous reviendrons sur sa définition dans la section 2.2, prévue à cet effet. Ce chapitre traite de la modélisation utilisateur et a pour but de satisfaire trois objectifs. Le premier sera de répondre à un certain nombre de questions générales [39] telles que :

- Quel est l'intérêt de modéliser les utilisateurs d'un système ?
- Quelles sont les caractéristiques les plus significatives des utilisateurs d'un système ?
- Quels sont les types de comportement utilisateur qui se distinguent dans un système ?
- Quelles sont les techniques de modélisation qui peuvent être appliquées pour maintenir les modèles utilisateurs d'un système ?

Le second objectif sera de passer en revue les différents concepts et techniques en vigueur dans l'élaboration d'un modèle utilisateur. Parallèlement à cela, le dernier objectif sera de tenter de fixer des balises, des méthodes ainsi que des décisions quant à l'utilisation d'une représentation ou d'une autre. La représentation retenue servira de support à l'élaboration d'un indice de confiance (cf Chapitre 5). Pour atteindre cet objectif, nous prendrons pour fil conducteur, l'enrichissement du méta modèle utilisateur.

Ce chapitre est composé de cinq parties. Les deux premières vont, respectivement, tenter de contextualiser et de définir le modèle utilisateur. Dans la troisième section nous découvrirons une représentation couramment utilisée de celui-ci. Pour cela, nous prendrons pour fil conducteur, l'enrichissement du méta modèle utilisateur. La quatrième partie concernera l'étude d'un processus de modélisation

utilisateur. Enfin, nous découvrirons une forme particulière de modélisation utilisateur appelée modèle de stéréotype.

## 2.1 Contexte

L'idée de modèle utilisateur [1] a vu le jour dans les années 70 afin de répondre aux besoins de concevoir des systèmes aptes à s'harmoniser avec l'utilisateur [19]. Depuis son apparition, ce concept est en constante évolution. Une évolution marquée par le contexte socio-économique, l'avancée technologique et la réévaluation continue de ses ambitions. C'est avec l'initiative de Kobsa et Wahlster en 1986, et leur étude concernant la modélisation utilisateur dans le cadre des systèmes de dialogue en langage naturel [51], que les recherches sur ce sujet ont véritablement débuté. Les modèles utilisateurs feront par après l'objet de recherche dans de nombreux domaines [39] comme la représentation des connaissances, le traitement du langage naturel, l'intelligence artificielle, les raisonnements automatisés, l'apprentissage automatique, etc.

La plupart des problèmes de modélisation utilisateur peuvent se résumer en trois grandes questions [39] :

- Quelle(s) utilisation(s) peut-on faire des informations capturées dans le modèle utilisateur ?
- Que doit contenir le modèle utilisateur ?
- Comment un système obtient des informations pertinentes au sujet des utilisateurs ?

La première interrogation nous amène à considérer le contexte d'utilisation du modèle utilisateur. Autrement dit, dans quel cadre va-t-on se servir du modèle, qui va l'utiliser et comment ? Pourquoi a-t-on besoin de modéliser l'utilisateur ? Concrètement, imaginons le cas du médecin pratiquant la médecine générale. L'utilisation qu'il va faire d'un modèle utilisateur va être à priori totalement différente de celle d'un site web. Typiquement, son but va être une meilleure gestion des informations sur le patient, un meilleur suivi de celui-ci ainsi qu'une augmentation de la justesse des diagnostics. Les informations contenues dans un modèle utilisateur ne vont pas être identiques pour chaque cas d'utilisation. Ceci répond aussi à la seconde question posée. Enfin, la dernière question concerne les différents mécanismes

1. Nous utiliserons désormais l'appellation « modèle utilisateur » conforme à la littérature scientifique en lieu et place de « profil utilisateur »

permettant de nourrir le modèle avec des données pertinentes (en rapport avec le contexte). Dans ce chapitre nous ferons le tour d'horizon des diverses techniques sur ce sujet.

Suite à cette réflexion menée sur les grandes questions concernant les modèles utilisateurs, nous pouvons dégager une ébauche de démarche de définition de modèle utilisateur : (1) définition du contexte d'utilisation, (2) définition des informations pertinentes à retenir dans le modèle en fonction de (1), (3) définition des mécanismes permettant de gérer le modèle utilisateur (par exemple, l'alimentation du modèle par des données). Les étapes correspondent respectivement aux questions posées ci-dessus.

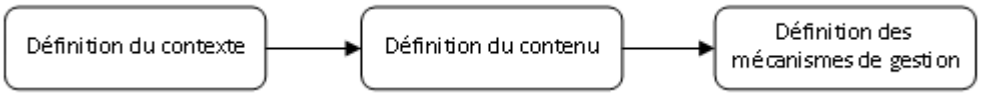



FIGURE 2.1: Processus de définition d'un modèle utilisateur

## 2.2 Définition

Le modèle utilisateur regroupe toutes les informations concernant l'utilisateur nécessaires pour adapter un service, un produit, un résultat (typiquement dans notre étude une page web) en fonction des caractéristiques de ce dernier. Il est à la base du système de personnalisation. En effet, la personnalisation d'une page pour un internaute implique incontestablement de posséder des informations sur celui-ci. Plusieurs auteurs scientifiques ont proposé leur définition du concept de modèle utilisateur. Nous retiendrons celle issue des travaux de Wahlster et Kobsa [51] :

> *Un modèle utilisateur est une source de connaissance qui contient des acquisitions sur tous les aspects de l'utilisateur qui peuvent être utiles pour le comportement du système.*

Tentons d'abord d'éplucher quelque peu cette définition afin de démystifier les différentes notions qui entrent en jeux. Dans notre cas, ledit « système » de cette définition fait référence au Système de Personnalisation Web (SPW), c'est-à-dire le système regroupant tous les composants servant à personnaliser une page web. L'utilisateur sera, quant à lui, un internaute naviguant sur un site internet. Enfin, les « acquisitions » seront de trois types : la réception, la capture et la déduction. Pour rappel, ces trois dernières font référence (respectivement) à la réception d'informations de la part de l'utilisateur (typiquement via un formulaire HTML), la capture de données techniques par le navigateur et la déduction d'informations par l'analyse comportementale de celui-ci.

## 2.3 Représentation courante

Dans cette section nous allons découvrir comment un modèle utilisateur est couramment représenté. Cette représentation est également une des plus élémentaires et est fréquemment appelée “représentation multidimensionnelle”.

### 2.3.1 Modélisation des dimensions

La figure 2.2 expose une manière rudimentaire de représenter le modèle utilisateur correspondant au profil de John Lage.

| | |
|---|---|
| Nom: | *Lage* |
| Prénom: | *John* |
| Age: | *30* |
| Genre: | *Homme* |
| Rue: | *Rue de Fer* |
| Numéro Rue: | *20* |
| Ville: | *Namur* |
| Code Postal: | *5000* |
| Pays: | *Belgique* |
| Hobbie: | *Photographie* |
| Genre de Film: | *Action* |

FIGURE 2.2: Modèle utilisateur simple de John Lage

L’ensemble des techniques de personnalisation existantes accorde beaucoup de crédit à l’approche par modèle utilisateur. Cependant, aucune catégorisation des données présentes dans le modèle utilisateur ne fait l’unanimité. Comme le cite D. Kostadinov [21], différentes applications ou recherches utilisant le concept de profil ou de modèle utilisateur définissent chacune leurs propres catégories de données. Par exemple, le standard concernant la sécurisation des profils [**?**] permet de définir des classes afin de faire la distinction entre les attributs démographiques, les attributs professionnels et les attributs de comportement. Dans les travaux de G. Amato et U. Straccia[1], une découpe en cinq catégories d’informations est prônée : données personnelles, données de livraison, données collectées, données de comportement et données de sécurité. [21] opte pour une représentation proche de cette dernière mais quelque peu divergente : données personnelles, domaine d’intérêt, données de qualité, données de livraison et données de sécurité. Les données personnelles comprennent les attributs qui font l’identité de l’utilisateur (par exemple son nom, son adresse, etc.) ainsi que les attributs démographiques (tels

que l'état civil, la profession, etc.). La dimension domaine d'intérêt inclut des attributs de préférences (comme le genre de film cité dans le profil de notre exemple ci-dessus), d'intérêts (comme les hobbies, également cité ci-dessus) ou de compétences (par exemple le niveau de qualification en photographie). Celle-ci se prête bien à ce que nous cherchons. En effet, dans la personnalisation web, la majorité des modèles utilisateur vont décrire les caractéristiques et centres d'intérêts de l'utilisateur. La topologie de connaissances au sujet du modèle utilisateur définie dans ce chapitre est légèrement [2] inspirée de cette dernière représentation. Par soucis de terminologie, nous appellerons « dimensions » ces classes ou catégories d'informations [21].

Un modèle utilisateur est donc, en général, structuré en différentes dimensions, qui sont elles-mêmes constituées d'attributs. Un attribut correspond à une caractéristique de l'utilisateur (par exemple son âge). Parfois, un ensemble d'attributs, sémantiquement liés, sont groupés logiquement afin de constituer ce que l'on appellera une « sous-dimension ». Ainsi, l'adresse est une sous-dimension car elle est composée des attributs « Numéro », « Rue », « Code Postal ». Le méta-modèle d'une telle représentation se profile de la sorte :

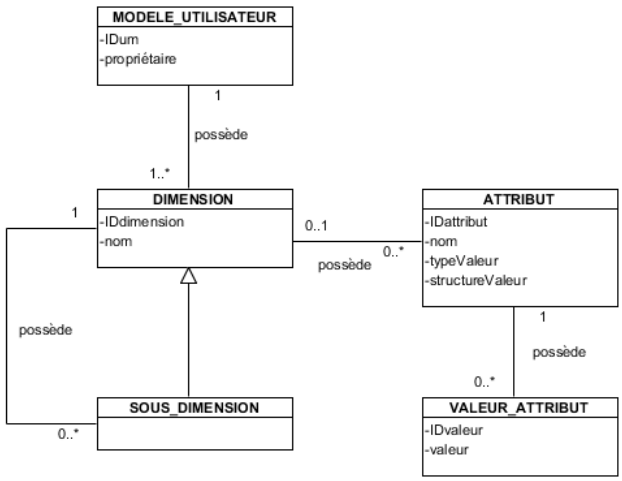



FIGURE 2.3: Méta-modèle du modèle utilisateur

2. La typologie proposée par [21] est trop large pour notre utilisation. Elle va donc est adaptée.

Un modèle utilisateur a un identifiant et un propriétaire. Comme nous l'avons vu, il possède une ou plusieurs dimensions. Celles-ci sont identifiées par un identifiant "IDdimension" et un nom. Elles peuvent elles-mêmes posséder une ou plusieurs dimensions. Il est à noter qu'une « sous-dimension » (dimension d'une dimension) ne peut posséder de dimension. Une dimension peut contenir des attributs. Un attribut est identifié par un identifiant "IDattribut" explicite. Il est aussi caractérisé par un nom, un type de valeur (typiquement entier, réel, chaine de caractères, etc.) et une structure de valeur (une valeur atomique, un ensemble de valeurs, etc.). Enfin, un attribut possède zéro, une, ou plusieurs valeur(s) [3] et est donc lié à l'entité « VALEUR_ATTRIBUT » qui renferme un identifiant et sa valeur.

Voici à présent le modèle utilisateur de John enrichi des dimensions « Données personnelles » et « Domaine d'intérêt » et de la sous-dimension « Adresse » :

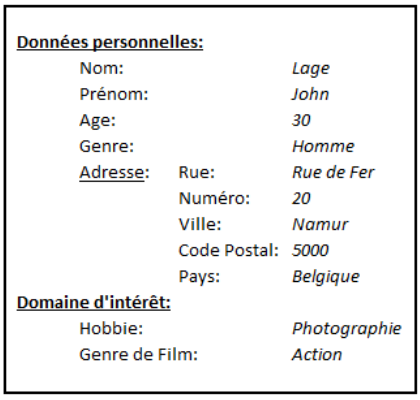



FIGURE 2.4: Modèle utilisateur de John Lage avec dimensions et sous-dimension

Le lecteur pourra vérifier que la figure 2.4 est bien l'instanciation du méta-modèle de la figure 2.3.

Ici un attribut est représenté en texte simple suivi de « : » et sa valeur correspondante est écrite en « Italique » à sa droite. Ainsi, « Lage » est la valeur de l'attribut « Nom ».

Notons que la définition des dimensions est libre. Le gestionnaire du modèle utilisateur peut en créer autant qu'il le souhaite. A titre illustratif, d'autres dimensions comme « Connaissance » et « Objectifs » (de l'internaute) peuvent également être intéressantes dans certains cas.

Le méta-modèle représenté à la figure 2.3 servira de fil conducteur tout au long de ce chapitre. Dans la suite nous allons découvrir différents éléments liés à la modélisation utilisateurs qui viendront

3. Nous verrons par la suite des cas concrets où un attribut nécessite de pouvoir être lié à plusieurs valeurs.

enrichir ce modèle. Ce méta-modèle résultant nous servira alors de base dans les chapitres qui suivront.

### 2.3.2 Modélisation du contexte

Un aspect intéressant à ne pas négliger dans le cadre de la modélisation utilisateur est la prise en compte du contexte.

> *Le contexte regroupe toutes les informations relatives à l'environnement dans lequel se fait l'interaction entre l'utilisateur et le système d'information [21].*

On relève trois types de contextes. Le *contexte temporel*, qui va renseigner sur le moment de l'interaction. Le *contexte spatial* qui indique le lieu dans lequel se situe l'utilisateur. Et le *contexte technique* qui va informer sur l'équipement tant matériel que logiciel que possède l'utilisateur.

A titre représentatif, adaptons le profil de « John Lage » pour y inclure un contexte spatial sur l'attribut « Genre de Film ». On va supposer qu'il désirerait regarder un film d'action quand il est à son domicile mais que lorsqu'il se trouve en vacances, il préfèrerait regarder un dessin animé avec son fils. Les informations concernant le contexte sont représentées en « souligné italique ».

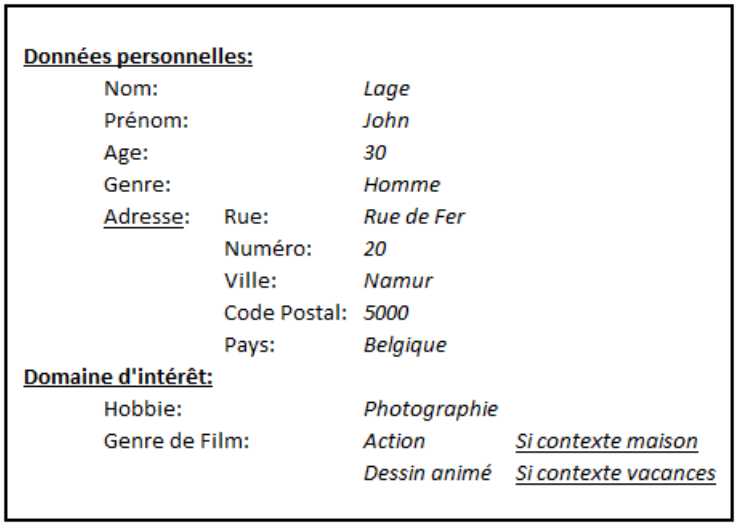



FIGURE 2.5: Modèle utilisateur de John Lage avec la modélisation du contexte

La figure 2.6 [21] dévoile le méta-modèle de contexte. On y retrouve les trois types de contexte décrits auparavant, baptisés « Dimension » de contexte. Les données de la dimension temporelle peuvent être relatées sous forme de date ou bien de moment (par exemple : matin, midi, soir). Celles de la dimension

spatiale sous forme localité ou de coordonnées. Le lieu où se situe l'utilisateur peut être statique (Fixe) ou dynamique (Mobile) lorsque celui-ci est en voiture par exemple. La dimension technique contient, quant à elle, à la fois des informations concernant le matériel et le logiciel, notamment, la taille de l'écran, la capacité de la mémoire et la vitesse du processeur pour le matériel, et le système d'exploitation et les logiciels bureautiques pour le logiciel.

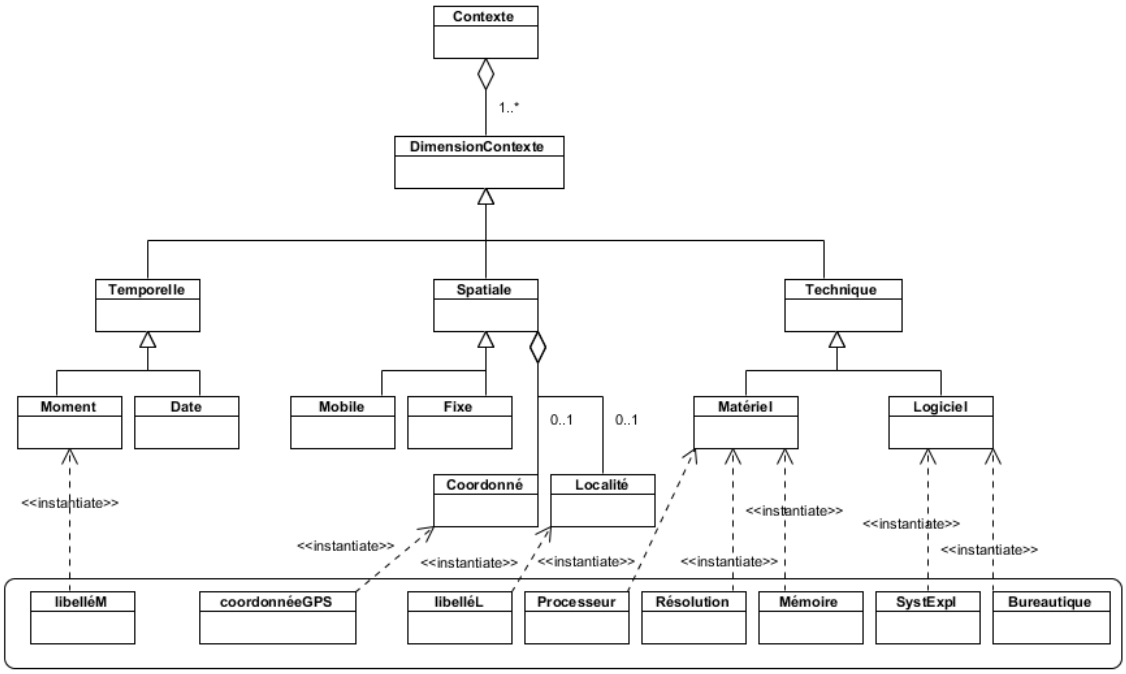



FIGURE 2.6: Méta-modèle de contexte [21]

### 2.3.3 Modélisation des préférences

Afin d'enrichir le modèle utilisateur, essayons d'intégrer des préférences. Préalablement, regardons ce qui se cache derrière ce terme inexpliqué. Une préférence est une expression permettant de hiérarchiser l'importance des informations dans un modèle utilisateur ou un contexte [21].

Reprenons notre modèle-exemple et ajoutons l'aspect préférence, représenté ici par un « score » que l'on peut associer aux valeurs d'un attribut. Ce score est une valeur prise dans l'intervalle [0,1] et est indiqué en caractère « gras » à droite de sa valeur correspondante.


FIGURE 2.7: Modèle utilisateur de John Lage avec la modélisation des préférences

D. Kostadinov et K. Djemai et K. Ghouali [21, 11]définissent de manière avancée le modèle de préférence. Ils distinguent notamment deux types de préférences (simple et composée). Une préférence simple peut être unitaire, binaire ou ensembliste selon le nombre d'objets (valeurs d'un attribut) qu'elle caractérise. Une préférence unitaire s'emploie pour un seul objet. Typiquement, un poids, une probabilité, un score (comme dans l'exemple cité ci-dessus). Une préférence binaire va définir une relation d'ordre entre deux objets (la relation d'incomparabilité [4] peut en être une). Enfin, une préférence ensembliste s'utilise sur un ensemble d'objet. Par exemple l'opérateur ensembliste « best », discriminant les éléments de l'ensemble en fonction de leur niveau de concordance avec un critère donné. Le second type de préférence sont les préférences dites «composées».

> *Une préférence composée est une expression de deux ou plusieurs préférences intermédiaires [21].*
>
> *Une préférence intermédiaire peut être une préférence simple ou une préférence composée.*

Plusieurs préférences peuvent être combinées ensembles de manière indépendante ou prioritaire (selon le fait qu'elles ont la même importance ou non) afin de former une préférence composée. La figure 2.8 présente le méta-modèle d'une telle représentation.

4. On parle d'incomparabilité entre deux éléments lorsqu'il n'est pas possible de choisir entre les deux.

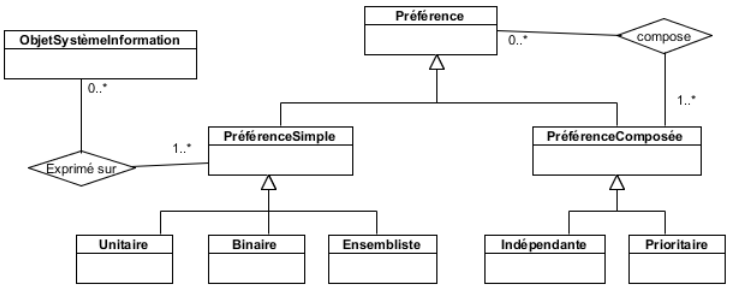
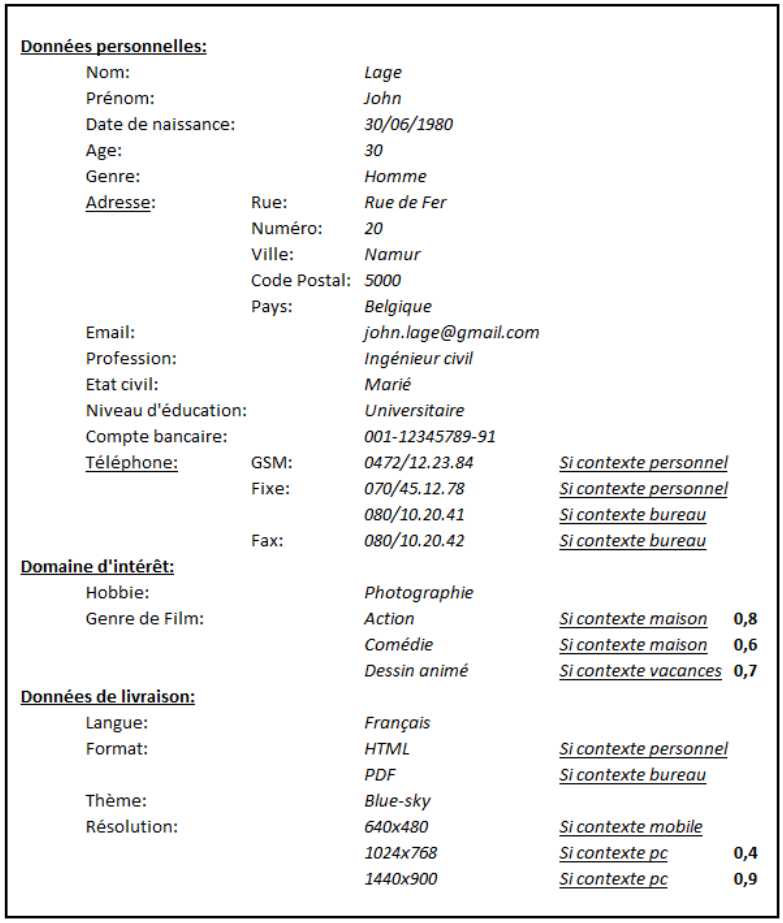



FIGURE 2.8: Méta-modèle de préférences [21]

La description des préférences la plus rencontrée est la préférence simple unitaire. Les autres, bien que puissantes, sont très peu exploitées. Ceci témoigne néanmoins des possibilités de la modélisation utilisateur. Nous verrons dans le chapitre 6 comment la modélisation des préférences viendra articuler l'intégration de la qualité dans le modèle utilisateur.

### 2.3.4 Exemple étendu d'une représentation d'un modèle utilisateur

Le modèle utilisateur étendu de John Lage (voir figure 2.9) comporte quelques nouveaux attributs ainsi qu'une dimension jusque maintenant inexploitée : la dimension « données de livraison ». Cette dimension va regrouper un certain nombre d'attributs liés aux possibilités d'affichages des sorties ou résultats. Ici, spécifiquement, les attributs « Langue », « Format », « Thème » et « Résolution » vont être utilisés afin de jouer ce rôle de présentation des résultats. Prenons l'attribut « Langue » pour exemple. Le modèle de John Lage nous indique qu'il voudrait voir les pages et contenus web en français (de préférence) étant donné la valeur « Français » de cet attribut. Il est important d'évoquer la relation de dépendance qui existe entre les attributs de cette dimension et la notion de contexte. En effet, dans le cas de la résolution d'écran d'une page web (attribut « Résolution »), celle-ci devra être de valeur différente en fonction que l'utilisateur la consulte depuis un ordinateur classique ou depuis un téléphone mobile (la résolution de ce dernier étant en principe inférieure à celle du pc).



| | | | | |
|---|---|---|---|---|
| **Données personnelles:** | | | | |
| Nom: | | *Lage* | | |
| Prénom: | | *John* | | |
| Date de naissance: | | *30/06/1980* | | |
| Age: | | *30* | | |
| Genre: | | *Homme* | | |
| Adresse: | Rue: | *Rue de Fer* | | |
| | Numéro: | *20* | | |
| | Ville: | *Namur* | | |
| | Code Postal: | *5000* | | |
| | Pays: | *Belgique* | | |
| Email: | | *john.lage@gmail.com* | | |
| Profession: | | *Ingénieur civil* | | |
| Etat civil: | | *Marié* | | |
| Niveau d'éducation: | | *Universitaire* | | |
| Compte bancaire: | | *001-12345789-91* | | |
| Téléphone: | GSM: | *0472/12.23.84* | *Si contexte personnel* | |
| | Fixe: | *070/45.12.78* | *Si contexte personnel* | |
| | | *080/10.20.41* | *Si contexte bureau* | |
| | Fax: | *080/10.20.42* | *Si contexte bureau* | |
| **Domaine d'intérêt:** | | | | |
| Hobbie: | | *Photographie* | | |
| Genre de Film: | | *Action* | *Si contexte maison* | **0,8** |
| | | *Comédie* | *Si contexte maison* | **0,6** |
| | | *Dessin animé* | *Si contexte vacances* | **0,7** |
| **Données de livraison:** | | | | |
| Langue: | | *Français* | | |
| Format: | | *HTML* | *Si contexte personnel* | |
| | | *PDF* | *Si contexte bureau* | |
| Thème: | | *Blue-sky* | | |
| Résolution: | | *640x480* | *Si contexte mobile* | |
| | | *1024x768* | *Si contexte pc* | **0,4** |
| | | *1440x900* | *Si contexte pc* | **0,9** |

FIGURE 2.9: Modèle utilisateur étendu de John Lage

Le méta-modèle (Figure 2.10) correspondant à cette représentation est quasiment identique à celui proposé plus haut. Toutefois, afin de tenir compte de la modélisation du contexte et des préférences, ce méta-modèle a été enrichi de deux champs « contexte » et « indice ». Le premier permet de spécifier le contexte dans lequel on se trouve, tandis que le second est destiné à recevoir un « score », une probabilité ou autre dans le but de renseigner sur les préférences de l'utilisateur.

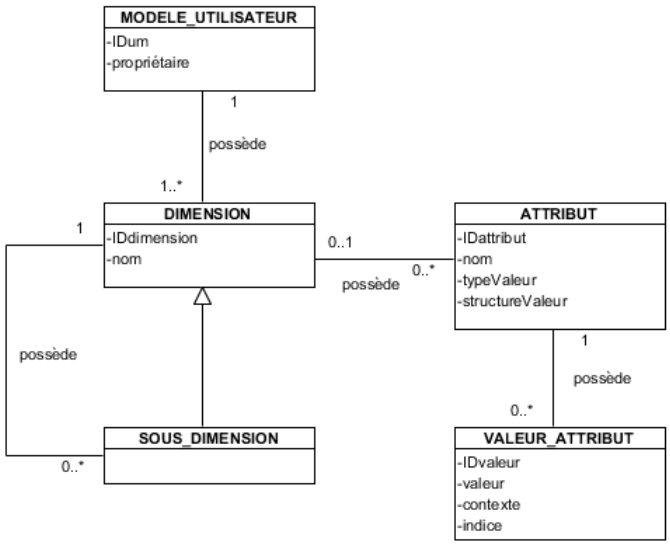



FIGURE 2.10: Méta-modèle de la modélisation utilisateur

Nous venons de découvrir la représentation dite multidimensionnelle. D'autres sont également citées dans la littérature. Nous faisons le choix de ne pas les découvrir car elles ne nous seront pas utiles pour la suite de ce mémoire. Notons tout de même que beaucoup d'auteurs définissent leur propre formalisme.

## 2.4 Processus de modélisation utilisateur

Maintenant que nous avons posé les bases de la modélisation utilisateur (la contextualisation, la définition et la représentation), nous allons en étudier les grandes étapes.

Le modèle utilisateur peut être l'objet d'un « cycle de vie » visant à définir différentes étapes de son existence. Comme le montre la figure 2.11, on y retrouvera logiquement en premier lieu l'étape consistant à construire le modèle, la « définition ». Celle-ci peut être qualifiée de phase préliminaire. Ensuite, il y a tout le travail d'alimentation du modèle utilisateur et bien sûr, son exploitation. Ces deux phases-ci sont non ordonnées et peuvent donc se dérouler en parallèle. En effet, il n'y a pas de période réservée, définie stricto-sensu pour nourrir le modèle utilisateur, ni même pour l'exploiter. Le système doit pouvoir, lorsqu'il reçoit une requête, l'analyser afin d'alimenter le profil de l'utilisateur, mais également, sortir des résultats destinés à, par exemple, adapter l'affichage de la page dans le cadre

de la personnalisation d'un site internet. C'est précisément ce que représente l'entrée « requête » et la sortie « réponse » de la figure 2.11. Cependant, exception à la règle, toute exploitation nécessite un contenu. Il est donc primordial que, la toute première fois, l'étape de remplissage du profil soit antécédente à celle d'utilisation.

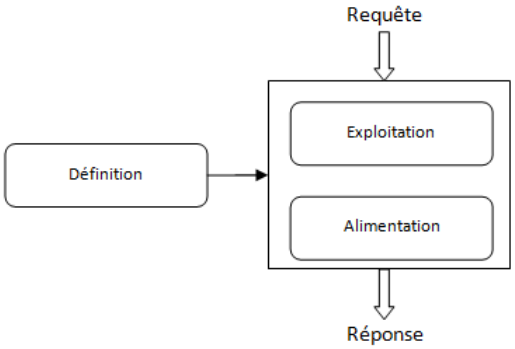



FIGURE 2.11: Schéma du processus du modèle utilisateur

En réalité, l'utilisation du terme « cycle de vie » est incorrecte. Un cycle de vie est par définition un processus qui comporte un début et une fin. C'est une période de temps qui se déroule entre un commencement (la définition du modèle utilisateur) et un aboutissement. Or, c'est justement cette dernière étape qui manque à l'appel dans notre cas. Effectivement, un modèle utilisateur n'est jamais « achevé » ou « abouti ». Il est en constante évolution au gré du flux d'informations qui l'alimente.

Un modèle utilisateur peut être supprimé mais cela ne constitue pas une étape en tant que tel car il n'est pas destiné à être détruit. Le but d'un modèle utilisateur est son exploitation et ce de manière la plus continue possible.

Dans les sections suivantes, nous allons suivre ce schéma de processus afin d'aborder dans le détail les différents mécanismes associés à la modélisation utilisateur.

### 2.4.1 Définition du modèle utilisateur

La définition du modèle utilisateur comprend trois activités (figure 2.12). D'abord, la définition du domaine dans lequel le modèle utilisateur va être utilisé. Ensuite, la détermination d'une « architecture » pour le modèle utilisateur. Enfin, une réflexion sur quelques aspects techniques propres à l'implé-

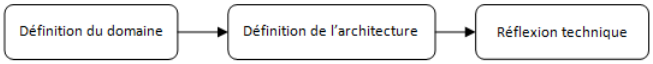



FIGURE 2.12: Trois activités de l'étape de définition du modèle utilisateur

mentation du modèle. Cette section se clôturera par une proposition de méthodologie de définition du modèle utilisateur.

#### 2.4.1.1 Domaine

Avant toute modélisation, il est primordial de bien comprendre le domaine d'utilisation du modèle utilisateur. Nous avons déjà abordé brièvement ce thème sous forme d'une question [5] à la section « Contexte » de ce chapitre. La question était : « Quelle(s) utilisation(s) peut-on faire des informations capturées dans le modèle utilisateur ? ». Sur base de cette première question, nous en avons déduit la seconde : « Que doit contenir le modèle utilisateur ? ». Le contenu d'un modèle utilisateur variera en fonction de son domaine. Cela va permettre de sélectionner des informations pertinentes à retenir. C'est pourquoi, bien cerner le cadre d'utilisation des données de l'utilisateur est essentiel. Sachant que nous pouvons modéliser un ou plusieurs contexte(s) à chaque attribut d'un modèle utilisateur, il serait pertinent d'ajouter une question à notre réflexion : « Dans quelles conditions l'internaute va utiliser l'application [6] ? ». Pour rappel, nous avons vu trois dimensions de contexte : la dimension spatiale, temporelle et technique. Cette réflexion va donc se baser sur trois « sous-réflexions » :

1. Où l'internaute consulte-t-il l'application ?
2. Quand l'internaute consulte-t-il l'application ?
3. Avec quoi/comment l'internaute consulte-il l'application ?

Répondre à ces trois sous-questions nous permettra par la suite d'établir une liste de contextes d'utilisation.

En résumé, la définition d'un modèle utilisateur passe donc par :

1. La définition du domaine
2. La recherche d'informations pertinentes à retenir
3. Une réflexion sur les conditions d'utilisation de l'application utilisatrice du modèle utilisateur

5. En l'occurrence la première des trois grandes questions qui se pose dans le cadre de la modélisation utilisateur.
6. Visiter le site internet dans notre cas.

Par exemple, un site de commerce électronique proposant des appareils photos à ses internautes a pour domaine la « vente d'appareils photographiques à des internautes » et va avoir besoin (au moins) du nom, du prénom, d'un téléphone, d'une adresse de livraison, d'une adresse de facturation et d'un numéro de compte bancaire. Un site proposant l'accès à une base de données de recettes de cocktails a pour domaine la « diffusion de recettes de cocktails à des internautes » et sera intéressé quant à lui par les ingrédients préférés du visiteur, ses recettes préférées, ses tendances de goûts (sucré, salé, amer, acide, fruité, crémeux, etc.), ses évaluations de recettes, ses allergies et ainsi de suite afin de proposer un contenu plus approprié.

#### 2.4.1.2 Architecture

La seconde étape de définition d'un modèle utilisateur est la création au sein même du modèle d'une structure (organisation) de données qui va permettre une meilleure lisibilité (du modèle utilisateur) mais aussi une division sémantique des données qui sera utile à l'exploitation des données. Une caractérisation des informations sera également effectuée, après quoi nous nous occuperons de la définition du contexte et des préférences. Nous appellerons cette étape « architecture ».

Il y a plusieurs possibilités. Les différentes manières de représenter un modèle utilisateur ont été évoquées ci-dessus dans les sections « Représentation courante » et « Autres représentations ». Prenons la représentation courante à titre illustratif. La structure d'une telle façon de faire se situe en deux points :

1. Découpe en attributs

   Il est nécessaire de transformer les informations pertinentes relevées dans l'activité de définition du domaine en attributs. Nous appellons cette tâche la "découpe en attributs". Ce découpage doit s'exercer méticuleusement. En fonction d'une découpe ou d'une autre, la modélisation pourrait diverger sémantiquement. Prenons l'exemple du site de vente d'appareil photo afin de bien cerner ce problème. Ce site web voudrait proposer un contenu adapté en fonction du niveau de l'utilisateur. Après réflexion sur le domaine, le gestionnaire du site a conclu qu'il avait besoin de stocker dans le modèle utilisateur le niveau de dextérité de l'internaute. Il distingue dans un premier temps trois niveaux possibles : expert, amateur et débutant. Nous allons voir deux modélisations différentes basées sur une découpe en attributs différente. La première, comme le montre la figure 2.13 comprend un attribut « Niveau » et peut être complétée des valeurs correspondant aux

divers niveaux cités ci-dessus. Chaque valeur possède un score variant entre 0 et 1 manifestant la certitude avec laquelle on pense que le niveau correspond bien à celui de l'utilisateur. Ici, la somme de chaque indice correspondant à chacune des valeurs n'est pas forcément égale à l'unité. La seconde (figure 2.14), découpe le contenu informationnel correspondant au niveau en photographie de l'utilisateur en trois attributs : « Niveau Photo. Expert », « Niveau Photo. Amateur » et « Niveau Photo. Débutant ». Chacun des ces attributs comprend un ensemble fermé[7] de valeurs. La valeur « Oui » associée à une probabilité décrit la probabilité que l'utilisateur **est effectivement** du niveau de l'attribut. Inversement, la valeur « NON » associée à une probabilité décrit la probabilité que l'utilisateur **n'est pas** du niveau de l'attribut. Ici, la somme des probabilités correspondant à la valeur « OUI » et « NON » doit impérativement être égale à l'unité. Ces deux découpes présentent donc bien deux sémantiques différentes, notamment au niveau de l'indice associé à une valeur. Remarquons aussi que la seconde découpe supporte très mal l'ajout de niveaux supplémentaires étant donné que cela nécessite une modification de la structure du modèle utilisateur. Modifier la structure peut accroître considérablement le temps de traitement ainsi que, dans certains cas, empêcher l'exploitation en temps réel du modèle [21]. Par contre, la première solution est plus évolutive car pour ajouter un nouveau niveau, il suffit d'ajouter une nouvelle valeur à l'attribut « Niveau » ainsi qu'un score. Etant donné la sémantique des indices et que leur somme ne doit pas impérativement être égale à un, un tel ajout n'implique pas de modifier les indices des autres valeurs. Cet exemple nous permet d'appuyer l'hypothèse qu'un changement structurel dans le modèle utilisateur peut impliquer un changement de sémantique. Il est donc fondamental de bien définir la sémantique de l'architecture pour que les données soient correctement exploitées et interprétées.

7. Ensemble dans lequel on ne peut ajouter ni retirer une valeur supplémentaire. Ses valeurs sont fixes et définies une fois pour toutes.

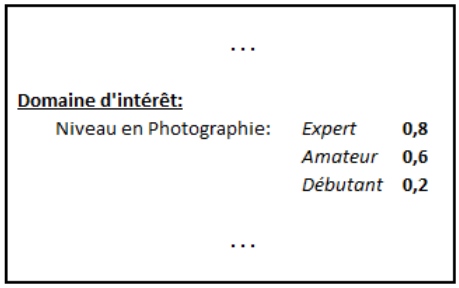



FIGURE 2.13: Exemple 1 de modélisation du niveau de dextérité en photographie


FIGURE 2.14: Exemple 2 de modélisation du niveau de dextérité en photographie

2. Catégorisation des attributs en dimension(s) et sous-dimension(s)

   Les attributs résultant de la découpe du premier point sont catégorisés en dimension(s) et sous-dimension(s). Nous avons vu qu'il est fréquent d'opter pour les dimensions « Données Personnelles », « Données Implicites » et « Domaine d'intérêts », chacunes tenant une sémantique différente. L'intérêt est double. Les informations sur l'utilisateur sont hiérarchisées, facilitant la lecture, mais aussi la compréhension. Et les traitements effectués (typiquement l'alimentation et l'exploitation) sur ces dimensions peuvent différer en fonction de leur sémantique. Par exemple, on sait que les attributs de la dimension « Données implicites » sont remplis automatiquement par le système via les informations du navigateur ou bien par les informations contenues dans l'en-tête « http » d'une requête. Cela n'est pas le cas pour les données personnelles : la majorité d'entre elles sont complétées par l'utilisateur via des formulaires web. Quant à la dimension « Domaine d'intérêts » les données seront principalement issues des scripts ayant pour fonction d'analyser et interpréter

les interactions homme-machine. En ce qui concerne l'agrégation d'attributs en sous-dimension, le regroupement se fait principalement entre attributs possédant une sémantique liée. Le cas classique est la sous-dimension « Adresse » qui comprend les attributs « Rue », « Numéro », « Ville » et « Code postal ». Il n'y a pas de règles pour le regroupement, mais seulement du bon sens.

Jusqu'ici, nous avons une structuration hiérarchique des informations concernant l'utilisateur. Nous allons désormais tenter de caractériser ces informations. Il existe différents types d'informations. Entre autres, il peut y avoir des chaines de caractères, des nombres, et même des dates. Un attribut va donc être caractérisé par un type de valeur. Aussi, certains attributs sont liés à une unique valeur, d'autres à un ensemble de valeurs. Un attribut est donc également déterminé par une structure. En somme, en fonction des caractéristiques de chaque attribut, la façon d'exploiter (ou même de nourrir) le modèle peut être différente.

Il ne reste plus qu'à définir les contextes et la façon de représenter les préférences. Pour les contextes, sur base de la réflexion sur les conditions d'utilisations menée auparavant dans l'étape d'analyse du domaine, une liste de contextes peut être déduite aisément. Au sujet des préférences, pour chaque attribut, il est nécessaire de définir une sémantique claire et précise afin d'éviter les problèmes d'ambigüité sémantique comme commenté ci-dessus.

En synthèse, pour définir l'architecture d'un modèle utilisateur il faut :

1. Effectuer la découpe en attributs
2. Catégoriser les attributs
3. Caractériser les attributs
4. Définir les contextes d'utilisation
5. Définir les préférences

#### 2.4.1.3 Aspects techniques

Après la réflexion et la modélisation, vient la réalisation. C'est dans cette étape que tous les aspects techniques liés à l'implémentation du modèle vont être envisagés. L'exemple le plus significatif est sans doute le choix concernant la façon de stocker le modèle utilisateur. Allons-nous utiliser une base de données relationnelle ou employer le stockage en XML ? Quelle sera la structure de la base de données

ou celle d'un document XML ? En principe, une fois cette étape réalisée, le modèle utilisateur doit pouvoir être nourri et exploité.

#### 2.4.1.4 Proposition de méthodologie

Pour correctement définir un modèle utilisateur il faut :

1. Effectuer une analyse du domaine du modèle utilisateur
    a) La définition du domaine
    b) La recherche d'informations pertinentes à retenir
2. Déterminer l'architecture du modèle utilisateur
    a) Effectuer la découpe en attributs
    b) Catégoriser les attributs
    c) Caractériser les attributs
    d) Définir les contextes d'utilisation
    e) Définir les préférences
3. Aspects techniques : réflexion, choix et implémentation

Le lecteur pourra adapter cette manière de procéder en fonction de ses propres besoins.

### 2.4.2 Alimentation du modèle utilisateur

Maintenant que le modèle utilisateur est défini, la deuxième étape de notre processus de modélisation utilisateur peut débuter : l'alimentation du modèle. Notons que dans le cadre de la personnalisation cette phase est appelée « acquisition de données ». « Alimentation » étant spécifique à la modélisation utilisateur. En effet, on *nourrit* un modèle utilisateur avec des données alors qu'un système de personnalisation *acquiert* des données. Ceci est dû au fait qu'aucune « intelligence » n'est présente dans le modèle utilisateur. C'est un concept statique (contrairement à un SPW). Dans cette section il sera question de mener une étude sur les types et les sources de données. Après quoi nous nous pencherons sur l'alimentation effective et la mise à jour d'un modèle utilisateur.

#### 2.4.2.1 Types de données

Un modèle utilisateur est en quelque sorte une agrégation d'informations de tous types concernant un utilisateur. Il n'y a pas vraiment de limites, toute information pouvant être stockée peut en faire partie. Notamment les données saisies par l'utilisateur [15], les données destinées à décrire l'organisation et la structure des résultats [25], les données concernant les contenus [31], les données d'utilisation [33], etc.

Il existe différents types de données. Il y a les données dites explicites, et les données dites implicites ou tacites [39].

Le premier type est le plus commun. Les connaissances explicites sont celles qui sont transmissibles dans un langage formel et structuré. Typiquement, le nom, le prénom, l'adresse, le numéro de téléphone, la date de naissance, etc. Ces informations sont, en général, aisément capturables, enregistrables et traitables.

Le second type est plus difficile à articuler. La formalisation, le traitement et la communication des connaissances tacites (ou implicites) est une tâche plus ardue de par leur nature. Les informations tacites sont des connaissances personnelles, non-tangibles, qui peuvent être distinguées en deux classes. D'une part, les aspects cognitifs, à savoir les représentations mentales que les Hommes se font d'un objet, d'un concept ou autre. D'autre part, les aspects techniques tels que le savoir-faire, les aptitudes, les capacités, les compétences, les intérêts ou encore l'expérience. Selon Nonaka and Takeuchi [35], les connaissances tacites sont les plus importantes... pour les entreprises. En effet, l'expérience, le savoir-faire, le savoir-vivre, le relationnel sont autant de connaissances difficilement exprimables qui agissent sur différents facteurs essentiels (notamment la productivité). Est-ce vrai également pour les connaissances tacites d'un internaute ? Quoi qu'il en soit, un des défis pour le gestionnaire de modèles utilisateurs sera de tenter d'intégrer tant bien que mal [8] ce type de connaissances sur l'utilisateur et cela commencera par une bonne modélisation du modèle (cf. section 2.2). Au plus le modèle utilisateur sera rempli de telles informations, au plus la personnalisation sera efficace de par la bonne compréhension des besoins et intérêts de chaque internaute.

8. Il est à l'heure actuelle impossible de savoir tout de l'utilisateur, il en saura toujours plus que ce qu'il ne pourra expliciter. La partie « source de données » de cette section décrira un procédé basé sur l'analyse interactionnelle qui tentera de déduire ce genre d'informations. Néanmoins, de manière très partielle et rudimentaire.

#### 2.4.2.2 Sources de données

Une distinction que l'on retrouve très couramment dans la littérature scientifique [39, 11, 47, 45, 42, 41, 18] est la différenciation entre acquisition explicite de données et acquisition implicite. A ces deux types d'acquisitions de données sont associés différentes sources de données. Une source de données est l'élément, l'objet ou l'acteur qui a envoyé les données dont il est question, ou permis cet envoi. Dans ce chapitre, nous utiliserons le concept de source de données comme « méthode qui a permis d'obtenir les informations ». Cela prendra son sens dans la suite de cette section. Nous allons à présent passer en revue les différentes techniques d'acquisitions ainsi que quelques sources de données classiques qui sont couramment d'application.

**Acquisition explicite de données** On parle d'acquisition explicite de données lorsque les données sont fournies par l'utilisateur même. Celui-ci a clairement l'intention de les fournir et est conscient des données qu'il partage. On dit qu'il alimente explicitement son profil.

L'obtention explicite de données se fait principalement via des formulaires Web. Un formulaire Web est un ensemble de composant(s) HTML permettant à l'utilisateur de saisir de l'information. La figure 2.15 montre un exemple de formulaire HTML invitant l'internaute à compléter son profil. Une fois les informations entrées, celui-ci n'a plus qu'à cliquer sur le bouton « Envoyer » afin d'envoyer ce formulaire à un serveur dont la fonction sera d'analyser, de traiter et d'enregistrer les données reçues. Cette manière de faire constitue la façon la plus simple pour compléter le modèle utilisateur d'un internaute. Néanmoins, c'est à l'utilisateur d'effectuer le travail de saisie, ceci peut parfois le rebuter. De plus, les données étant entrées manuellement, des fautes de frappes peuvent se glisser. Et plus encore, l'internaute peut volontairement introduire de fausses informations ! Dans le chapitre sur la qualité, nous étudierons en détail ces phénomènes et tenterons d'y proposer des solutions.

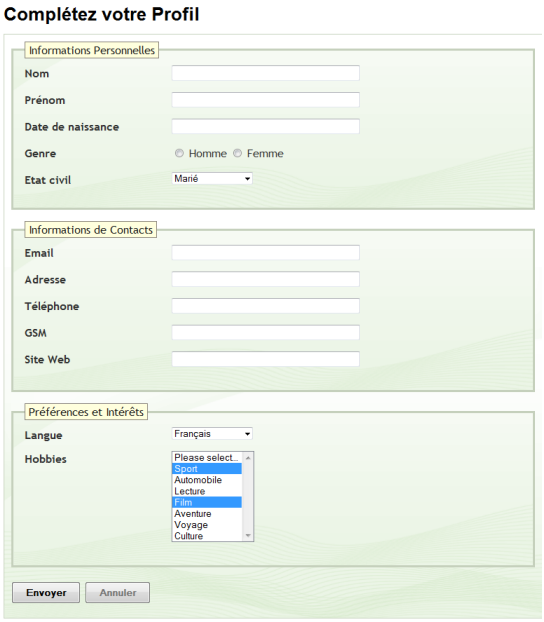



FIGURE 2.15: Formulaire HTML de complétion de profil

Au jour d'aujourd'hui, les formulaires se voient améliorés par toutes sortes de technologies. Notamment, le Javascript, langage de programmation exécuté coté client (c'est-à-dire chez l'internaute par le navigateur web). Ce langage permet, entre autres, d'améliorer l'interaction avec l'utilisateur, et, dans le cas de nos formulaires, d'augmenter leur ergonomie rendant la tâche de complétion beaucoup plus attrayante pour l'internaute. AJAX, pour « Asynchronous Javascript And XML » est une nouvelle façon d'élaborer des applications Web et va permettre de réaliser des sites dynamiques. AJAX résulte de la composition de diverses technologies dont le JavaScript et le langage de balisage générique XML (Extensible Markup Language). Le dynamisme pour les formulaires est intéressant dans la mesure où une certaine « validation en temps réel » va être réalisable. Cette validation se conçoit par différents mécanismes tels que l'auto-complétion, la vérification syntaxique des données saisies ou même la vérification sémantique. L' « auto-complétion », anglicisme de « complètement automatique », est une fonctionnalité qui, lorsque l'utilisateur débute la saisie d'un mot, propose un certain nombre de mots que celui-ci pourrait vouloir entrer. Tout se passe en temps réel et permet donc à l'internaute de se voir proposer une quantité d'information filtrée sur base des premières lettres qu'il a déjà introduites. Cela permet dans certains cas d'éviter d'éventuelles fautes de frappes et de consulter le domaine des

valeurs attendues possibles pour un préfixe donné. Ce mécanisme est plutôt une aide à la saisie qu'une réelle technique de validation.

La validation sémantique quant à elle, permet de vérifier si les informations que l'utilisateur a fournies ont un sens. En règle générale, il n'y a aucune intelligence artificielle derrière cette fonctionnalité, mais simplement une routine vérifiant que la valeur entrée appartient bien à un ensemble de valeurs existantes. Par exemple, nous voulons que l'internaute indique quelle est sa langue maternelle. Pour valider sémantiquement sa saisie, il suffira de la comparer avec une base de connaissances concernant les langues parlées dans le monde. Dans le cas, où l'entrée n'a pas de correspondance avec cette base de connaissance, il y a de très fortes chances qu'une erreur ait été commise par l'utilisateur. Dans ce cas, le formulaire n'est pas validé. Notons que tout ne peut pas être validé sémantiquement. Certains attributs comme le prénom d'une personne sont difficilement vérifiables auprès d'une base de connaissance mais peuvent néanmoins faire l'objet d'une validation croisée. La validation croisée est un procédé par lequel on va pouvoir valider ou non des données sur base d'autres données que l'on possède déjà. Prenons pour exemple le cas d'un employé devant s'inscrire sur un site partenaire de son entreprise. Si nous savons que l'utilisateur se nomme « Lage », qu'il fait partie de l'entreprise en question et qu'il n'y a qu'un seul employé dénommé « Lage », par inférence on peut en déduire que l'entrée « Jon » est erronée. Le site pourrait aller plus loin en lui indiquant qu'il a fait une faute d'orthographe et lui proposer « John » comme prénom.

Les différentes techniques de validation de formulaire proposées ci-dessus aident à augmenter la qualité des informations que l'utilisateur fournit mais ne doivent en aucun cas servir de légitimité à l'attribution d'une confiance exagérée en ce contenu informationnel.

**Acquisition implicite de données** On parle d'acquisition implicite de données lorsque les données sont déduites de la navigation. L'utilisateur ne désire pas[9] explicitement les fournir au système et ne sait à priori pas quelles informations vont être acquises. On dit que le système alimente implicitement son profil.

Il y a plusieurs manières de récolter des connaissances sur l'internaute de façon implicite. Nous en distinguons cinq : (1) analyse des variables et propriétés du navigateur web (2), analyse des cookies (3), analyse des paramètres HTML (4), analyse interactionnelle (5) et analyse de processus

9. Cela ne signifie pas qu'il ne veut pas que le site obtienne ces informations.

1. Analyse des variables et propriétés du navigateur web : Les navigateurs web possèdent quelques informations très intéressantes concernant l'utilisateur (principalement son environnement technique). La figure 2.16 en présente les principales.

| Variables | Valeur |
|---|---|
| Nom d'application : | Netscape |
| Nom de code d'application : | Mozilla |
| Version : | 5.0 (Windows; fr) |
| Langage : | FR |
| Système d'exploitation : | Win32 |
| Java activé : | Oui |
| Anti-aliasing des Polices : | Non |
| Résolution courante : | 1280 x 1024 |
| Résolution maximale : | 1280 x 1024 |
| Profondeur de couleurs : | 24 bits |
| Couleurs : | 16777216 |
| Pages vues : | 5 |
| IP : | xxx.xxx.xxx.xxx |
| Hôte : | host |
| MIME Type : | video/quicktime |
| | audio/wav |
| | ... |
| Plugins : | npqtplugin.dll |
| | npqtplugin2.dll |
| | npqtplugin3.dll |
| | ... |

FIGURE 2.16: Informations intéressantes fournies par le navigateur internet

La plupart des langages web permettent d'y accéder facilement. Typiquement, ces données pourront servir à adapter la présentation en fonction du navigateur, du système d'exploitation, de la résolution, de la langue, etc.

2. Analyse des cookies : Un cookie est un amas d'informations laissé par un serveur web chez un client HTTP. Il permet de garder chez le client quelques informations jugées utiles par l'application côté serveur. Un cookie contient les éléments suivants [30] :
   - un nom (name) : valeur alphanumérique permettant d'identifier de manière unique le cookie.
   - un contenu (value) : une valeur de contenu.
   - un domaine (domain) : représente le nom de domaine du serveur qui a créé le cookie.
   - un chemin (path) : URLs pour lesquelles le cookie est valable.
   - une date (expire) : date d'expiration du cookie.
   - un drapeau de sécurité (security) : indique si le cookie agit dans le cadre d'une transaction sécurisée (SSL, HTTPS) entre client et serveur ou non.
   - un commentaire (comment) : documentation au sujet de l'utilisation du cookie.

   Ils peuvent donc servir à identifier un utilisateur ou à stocker des informations sur celui-ci. Il

serait dés lors intéressant de les consulter dans le but d'obtenir des informations qui vont venir compléter le profil de l'internaute.

3. Analyse des paramètres HTML : En HTML, il y a deux manières de transmettre des paramètres. Il y a la méthode « GET », où les paramètres transitent par l'URL et la méthode « POST » où les paramètres ne passent pas via l'URL. Il peut dans certains cas être intéressant d'analyser ses paramètres afin d'en obtenir des informations utiles pour compléter le modèle utilisateur.

4. Analyse interactionnelle : L'analyse interactionnelle ou comportementale est une technique visant à inférer un certain nombre d'informations au sujet de l'utilisateur sur base de ses comportements de navigation. Un panel (incomplet) d'indicateurs [11] pour la prédiction et la déduction de données concernant l'internaute est présenté dans l'annexe A. Les informations résultantes de l'analyse de ces indicateurs seront pour la plupart des préférences ou centre d'intérêts plutôt que des données « brutes » telles que l'adresse ou encore le numéro de téléphone, qui sont elles obtenues de manière explicite (cf. Acquisition explicite de données). Ces indicateurs pourront s'obtenir (entre autres) via le langage JavaScript, qui, exécuté coté client, va pouvoir disposer d'une multitude de données interactionnelles (notamment les événements liés à la souris).

   Notons que l'analyse comportementale peut combiner plusieurs de ses techniques de déduction afin d'obtenir d'autres informations, parfois plus riches. Prenons le cas où l'utilisateur clique sur un lien, ensuite, s'empresse de quitter la page qu'il vient de charger. En combinant les déductions de l'analyse des clics avec celles de l'analyse du temps passé sur les pages, on pourrait conclure que l'internaute ne trouve pas la page intéressante.

   Liana Razmerita [39] a introduit une classification des modèles utilisateurs contenant des données provenant de l'analyse interactionnelle. Elle distingue deux types de modèles :

   a) Le modèle peu profond (« shallow model ») : basé sur une interaction utilisateur-système à court terme. Ce modèle prend uniquement en compte les données interactionnelles de cette session de navigation (ne se souvient pas des déductions acquises lors de sessions antérieures). A chaque fois que l'internaute visite le site, le système de déduction repart de zéro, n'ayant dés lors aucune connaissance préalable de celui-ci point de vue interactionnelle. En quelque sorte, ces données peuvent être qualifiées de « volatiles ».

   b) Le modèle profond (« deep model ») : basé sur une interaction utilisateur-système à long

terme. Ce modèle prend en compte les données interactionnelles de cette session de navigation et des précédentes (se souvient des déductions acquises lors de sessions antérieures). Au fur et à mesure des visites, le système accumule des informations interactionnelles sur l'internaute.

L. Razmerita distingue également deux techniques d'analyse des informations comportementales de l'utilisateur (déduction) :

a) Les techniques de modélisation on-line (« on-line modeling techniques ») : le processus de déduction, d'extraction de caractéristiques au sujet de l'utilisateur se fait en temps réel.

b) Les techniques de modélisation off-line (« off-line modeling techniques ») : le processus de déduction, d'extraction de caractéristiques au sujet de l'utilisateur se fait en différé. Pour cela le système doit enregistrer toutes les traces d'interactions dans des fichiers « logs ». Ces fichiers sont analysés à un moment choisi par l'administrateur système (par exemple chaque jour à minuit). Il en ressort un certain nombre de caractéristiques qui vont venir alimenter le profil d'un utilisateur. Evidemment, cette technique ne fonctionne pas avec la modélisation peu profonde discutée ci-dessus.

Les possibilités de l'analyse comportementale sont énormes, cependant elles sont parfois difficiles à mettre en place. Combiner plusieurs techniques n'est pas aisément réalisable. De plus, la plupart des déductions se font sur base d'un contenu, d'une page, d'une action sur lesquels on doit connaitre la sémantique. En effet, lorsque l'internaute clique sur un lien pointant vers une page, le système en déduit que celui-ci est intéressé par le contenu de cette page. Encore faut-il connaitre ce contenu, le « comprendre » afin de pouvoir associer à cet internaute, un thème, une préférence ou un intérêt qui le caractériserait ! Outre cela, certaines pages possèdent plusieurs contenus rendant la tâche d'autant plus ardue.

5. Analyse de processus : On parle d'analyse de processus comme technique d'acquisition implicite de connaissances lorsque l'on peut déduire du processus même ou de ses résultats, des informations concernant l'utilisateur. Celles-ci ne sont jamais inconnues mais souvent supposées. Un processus, une fois accompli, a réussi ou a échoué. Supposons que le processus était l'achat d'un livre via un site web, que le paiement ait été reçu et que, de ce fait, le livre ait été expédié et livré. Il y a de fortes chances que le numéro de compte bancaire et l'adresse de livraison soient exacts.

L'intérêt de cette technique réside dans la validation des données que le processus va pouvoir apporter. Un autre exemple connu de ce type de processus est l'envoi d'un email (ou d'un SMS). Nous reviendrons sur ce sujet au chapitre 5 sur la définition d'un indice de confiance.

**Acquisition Hybride de données** Certaines données peuvent provenir à la fois de sources explicites et implicites. La valeur d'un attribut peut être complétée à partir de différentes sources. Pour l'illustrer, reprenons l'exemple du site de vente d'appareil photo. Le site peut proposer à l'internaute de s'inscrire afin de bénéficier d'offres promotionnelles. Lors de l'inscription, il est demandé explicitement à l'utilisateur (via un formulaire HTML) son niveau en photographie (professionnel, amateur ou débutant). Celui-ci stipule qu'il fait partie des « pros » de la photo. Par après, en analysant son comportement de navigation, le système déduit qu'il serait plus pertinent de le classer en amateur. L'attribut « Niveau en Photographie » a donc été nourri par deux sources, d'abord par l'utilisateur via le formulaire d'inscription, ensuite par le système de déduction. Un des problèmes de l'acquisition multi-sources est la concordance des résultats. Dans l'exemple ci-dessus, un attribut est alimenté par deux sources munies de valeurs différentes. Il y a conflit de valeurs entre les sources. Quels sont les mécanismes qui peuvent être mis en place pour parer à ce problème tout en gardant cet aspect multi-sources ? Aussi, si les valeurs avaient été identiques, peut-on considérer cette valeur comme correcte ? Ces questions seront étudiées en long et en large lors du chapitre 5 sur la définition d'un indice de confiance pour les informations web.

**Récapitulatif** La figure suivante fait la synthèse des points discutés précédemment concernant les sources de données. Il y a deux manières d'alimenter le modèle utilisateur. La première se fait par acquisition explicite de données, c'est-à-dire via des formulaires HTML que l'utilisateur remplit et envoie au serveur. La seconde, l'acquisition implicite, alimente le modèle utilisateur de façon discrète, sans que l'utilisateur l'ait explicitement demandé. Cette manière de nourrir le profil de l'internaute peut s'effectuer grâce à des techniques telles que l'analyse des variables du navigateur web, des cookies, des paramètres HTML, des processus et de son comportement. Cette dernière, peut être réalisée en temps réel, ou en différé par l'analyse de fichiers « logs ».

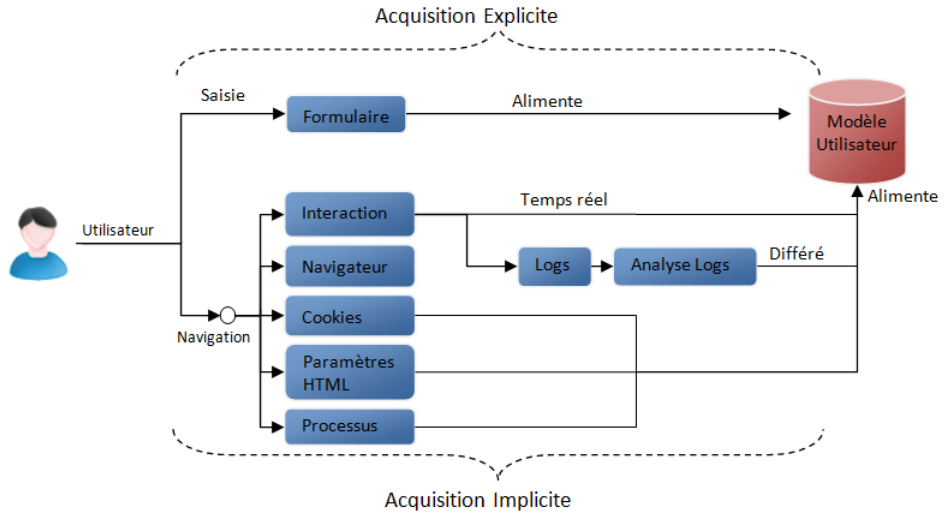



FIGURE 2.17: Récapitulatif des sources de données

Nous venons de découvrir un nouvel élément important pour la modélisation des utilisateurs. Nous pouvons dés lors mettre à jour notre méta-modèle utilisateur (figure 2.10) en incorporant l'entité « source » qui possède un identifiant et un nom. Une valeur peut provenir de 1 à N sources. En effet, toute valeur est fournie par une source de données, mais peut également être confirmée par d'autres sources de données. Une valeur peut donc posséder plusieurs sources de données. Une source peut être associée à 0 ou N attributs.

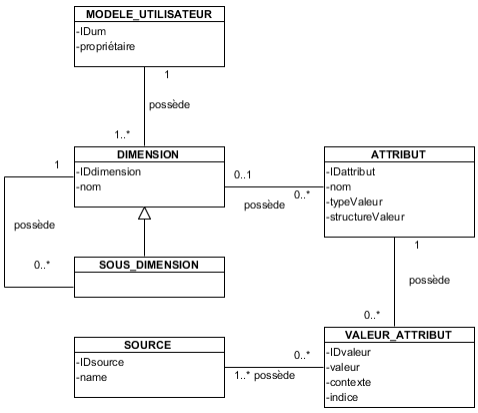



FIGURE 2.18: Méta-modéle utilisateur avec prise en compte des sources de données

#### 2.4.2.3 Alimentation des données

Alimenter le modèle utilisateur consiste à introduire une ou plusieurs valeur(s) pour des attributs précédemment définis lors de l'étape de définition. On parle d'instanciation du modèle utilisateur [21, 11]. L'instanciation peut être partielle (certains attributs définis sont instanciés) ou complète (tous les attributs définis sont instanciés). Comme nous l'avons vu, une valeur peut-être liée à un contexte. Si tel est le cas, celui-ci doit être indiqué dans l'enregistrement de la valeur. De même, une donnée sur l'internaute est toujours issue d'une unique source de données. On associera, dés lors, cette valeur à sa source. Enfin, dans le but de modéliser les préférences, nous avons vu qu'il était intéressant de calculer un indice associé à une valeur manifestant une certaine sémantique liée au concept de préférence et choisie par le gestionnaire du modèle utilisateur. Un indice de préférence sera donc calculé et associé à la valeur.

Les valeurs insérées doivent impérativement être syntaxiquement et sémantiquement correctes. Dans le cas contraire, l'insertion et l'utilisation risquent d'être compromises. Pour vérifier la syntaxe d'une valeur, il suffit de contrôler si son type effectif correspond au type attendu par l'attribut auquel elle est liée. Quant à la sémantique, elle ne peut être prouvée correcte ou non. C'est au gestionnaire du modèle utilisateur de mettre en place des mécanismes permettant d'assurer au mieux son exactitude.

#### 2.4.2.4 Mise à jour des données

Un modèle utilisateur n'est jamais « achevé », il est en constante évolution. Les données contenues en son sein peuvent s'affiner au fur et à mesure de la navigation de l'internaute [11]. A priori, la connaissance que le système possède d'un utilisateur ne peut que croître. Bien que, à long terme, une mauvaise gestion et modélisation des modèles utilisateurs, une sensibilité trop élevée vis-à-vis de l'acquisition de données et l'accumulation d'informations, peuvent avoir pour effet de bord la dégradation des connaissances sur les utilisateurs. La mise à jour des données doit donc se faire de façon méticuleuse et avisée.

La mise à jour peut se faire partiellement (les parties du modèle concernées par de nouvelles informations sont actualisées), ou complètement (le modèle est alors ré-instancié entièrement) [21].

Concrètement, la mise à jour se déroule comme suit. Pour un attribut, si une valeur est déjà présente dans le modèle d'un utilisateur mais est issue d'une source différente, on lui associe la nouvelle

source de données. Ainsi, la valeur de cet attribut possède désormais deux sources de données. Dans le cas contraire, la valeur associée à sa source est simplement ajoutée comme instance de l'attribut correspondant. En ce qui concerne les préférences, à chaque ajout de valeur, l'indice devra être recalculé.

#### 2.4.2.5 Archivage du modèle utilisateur

L'archivage du modèle utilisateur va permettre de garder en mémoire l'ensemble des instances du modèle utilisateur qui ont existé. Le principe est d'associer la date de création de l' « archive » avec l'ensemble des valeurs du modèle utilisateur [21, 11]. Un modèle utilisateur peut posséder plusieurs archives. Le but de cette approche est de pouvoir visualiser l'historique des modifications, suivre l'évolution des préférences d'un internaute ou encore analyser son comportement. Cela permet également de revenir à une précédente version en invalidant une mise à jour. Parce que cela nécessite de gérer un ensemble de version du profil, cette technique est souvent appelée « versionning ».

### 2.4.3 Exploitation du modèle utilisateur

La troisième et dernière étape du processus de modélisation utilisateur est l'utilisation du modèle utilisateur. Un modèle utilisateur peut être utilisé par exemple pour la personnalisation de système (web ou non) mais aussi pour la reformulation de requête [10] , le partage d'informations, l'agrégation d'informations, la génération de statistiques, etc. Dans le cadre de ce mémoire, nous voulons modéliser l'utilisateur afin de pouvoir personnaliser des services web. L'exploitation du modèle se fera donc par le système de personnalisation web. Nous verrons au chapitre suivant les différentes techniques de personnalisation web. Pour exploiter les données contenues dans un modèle, il est dans un premier temps nécessaire de les lire. Lire des données correspond à les sélectionner dans le but d'une utilisation prochaine. La lecture peut se faire partiellement ou complètement. Une lecture partielle est ciblée, le modèle n'est pas lu dans son entièreté contrairement à la lecture complète. Typiquement, cela correspond à la sélection d'un ou plusieurs attribut(s), mais pas la totalité. Après cette étape de lecture, les données peuvent être utilisées. Notons que l'exploitation peut se faire par le système, les gestionnaires ou bien directement par les utilisateurs.

10. Technique visant à améliorer les requêtes envoyées à un système/composant en incluant des informations pertinentes issues par exemple d'un modèle utilisateur.

## 2.5 Le modèle de stéréotype

Jusqu'ici nous avons étudié la forme classique de modélisation utilisateur : le modèle utilisateur. Il en existe une seconde forme que nous appellerons « modèle de stéréotype ». Cette dernière section va, de la même manière que pour le modèle utilisateur, tâcher de définir le modèle de stéréotype, de présenter ses intérêts, étudier sa représentation et son processus.

### 2.5.1 Définition

Il existe un second type de modèle utilisateur : le modèle de stéréotype. Celui-ci est tiré du concept défini par les sciences humaines du « stéréotype ». Un stéréotype est une généralité concernant un type d'individu, un groupe ou une classe sociale. En 1987, Fischer définissait le stéréotype comme étant une « *catégorie descriptive simplifiée par laquelle nous cherchons à situer autrui ou des groupes d'individus* ». E. Rich, dans ses travaux de modélisation utilisateur fin des années 70 [42, 41], fut le premier à incorporer le concept social de « stéréotype » dans la modélisation utilisateur. Il met en garde contre les associations négatives qui sont souvent faites au terme « stéréotype ». « *Il est important ici de restreindre son utilisation purement à une énumération descriptive d'un ensemble de traits qui apparaissent souvent ensemble. De ce point de vue, un stéréotype est simplement un moyen de capturer certaines structures qui existent dans le monde qui nous entoure* ». Selon lui, un stéréotype représente donc une collection de traits partagée par plusieurs utilisateurs. Par exemple, pour un étudiant, nous pouvons dégager les stéréotypes « Etudiant Master », « Doctorant » et « Ingénieur » [3, 17].

### 2.5.2 Intérêts

Les intérêts d'une telle modélisation sont multiples. D'abord, cela permet d'adopter un autre point de vue informationnel. Les données ne sont plus présentées individuellement mais de manière agrégée. D'autres types de raisonnements sont possibles et des statistiques sur la population peuvent être générées plus aisément. Ensuite, posséder une vue globale des utilisateurs va permettre de faciliter les échanges et partages d'informations au sujet de ceux-ci. En effet, il est plus intéressant et optimal d'échanger des informations du type « les hommes âgés entre 20 et 30 ans préfèrent regarder des films d'action » plutôt que « la personne 1 qui est un homme de 26 ans est intéressée par les films d'action », « la personne 2 qui est un homme de 23 ans est intéressée par les films d'action », et ainsi de suite.

Un autre avantage est la possibilité de raisonner sur un ensemble connu et bien déterminé d'attributs. Enfin, catégoriser un modèle utilisateur individuel dans un ou plusieurs modèles de stéréotypes peut dans certains cas l'enrichir grâce à des informations qui seraient présentes dans le stéréotype mais pas dans le modèle individuel. Néanmoins, rien ne garanti que ces informations sont également correctes pour cet utilisateur. Notons également que si le raisonnement (l'exploitation du modèle) est uniquement effectué sur base de ces stéréotypes, la connaissance que l'on aura de l'utilisateur durant le raisonnement sera sans doute moindre. En effet, l'agrégation des données en stéréotypes va avoir pour effet un renforcement des « pattern », caractéristiques communes à la population étudiée par un affaiblissement des particularités spécifiques de chacun des individus. Le modèle résultant sera donc moins riche en informations.

Bref, la modélisation de stéréotype apporte d'autres possibilités d'exploitations et permet d'adopter un point de vue de plus haut niveau par la généralisation des informations des profils utilisateurs.

### 2.5.3 Représentation

La représentation d'un modèle de stéréotype est souvent similaire au modèle utilisateur. Un stéréotype est une liste de couple <Attribut-Valeur(s)> éventuellement catégorisée en dimensions et sous-dimensions. Une valeur est à chaque fois associée à une probabilité indiquant le degré de certitude que l'attribut soit de cette valeur pour un utilisateur [47, 42, 41]. Un attribut peut posséder plusieurs valeurs. Typiquement, une valeur peut être un nombre, une date, une chaine de caractères ou un élément d'une liste. Une valeur peut également être un intervalle, ce qui n'était pas permis pour les modèles utilisateur. Dans certains cas, le stéréotype est divisé en deux parties. Une partie descriptive et une partie prédictive [47].

La figure 2.19 représente le stéréotype « Photographe Professionnel ». Précisons que nous n'utilisons pas ici le terme « Professionnel » afin d'exprimer qu'une personne en fait son métier. L'objectif est simplement de décrire le niveau de dextérité en photographie d'un individu. Un photographe de niveau professionnel n'est rarement plus jeune que 20 ans ou plus vieux que 65 ans, mais possède souvent (dans 55% des cas) entre 41 et 65 ans ou de 20 à 40 ans (35% des cas). Il y a 80% de chance qu'il soit de sexe masculin. Il y a beaucoup de chance qu'il soit manager ou travaille dans l'art ou la nature (40% et 45%). Avec cette dernière hypothèse, nous voyons clairement que l'objectif de la modélisation par stéréotype est moins d'être le plus représentatif possible de la réalité, que de modéliser au mieux le domaine avec

lequel le système devra traiter. D'où cette appellation « stéréotype ». Ce stéréotype est muni d'une seconde partie relatant les prédictions ou centres d'intérêts supposés de ces individus. Trivialement, celle-ci exprime le fait qu'un photographe professionnel possède, à 80%, un niveau professionnel en photographie. Notons que la somme de chaque indice correspondant à chacune des valeurs du niveau en photographie n'est pas forcément égale à l'unité. Ceci n'est pas le cas pour les attributs « Age », « Genre » et «Job » où l'on considère que tout le domaine des valeurs est représenté avec pour conséquence que la somme de chaque indice doit être égal à un (pour toutes les valeurs d'un attribut). Nous avons déjà discuté de ces choix de modélisation précédemment.

| | | | |
|---|---|---|---|
| **Nom:** | Photographe professionnel | | |
| **Profil:** | | | |
| | Age | *<20* | **0,05** |
| | | *20-40* | **0,35** |
| | | *41-65* | **0,55** |
| | | *>65* | **0,05** |
| | Genre | *Homme* | **0,8** |
| | | *Femme* | **0,2** |
| | Job | *Manager* | **0,4** |
| | | *Art & Nature* | **0,45** |
| | | *Autre* | **0,15** |
| **Prédictions/intérêts** | | | |
| | Niveau en Photographie: | *Professionnel* | **0,8** |
| | | *Amateur* | **0,6** |
| | | *Débutant* | **0,2** |

FIGURE 2.19: Stéréotype d'un photographe professionnel

Un stéréotype peut, dans certains cas, posséder la même syntaxe qu'un modèle utilisateur, cependant, sa sémantique est différente. Sémantiquement, il représente un ensemble de caractéristiques **communes que partagent les** utilisateurs, alors que le modèle utilisateur représente un ensemble de caractéristiques **individuelles concernant un** utilisateur.

La modélisation de stéréotypes ressemble fortement à la modélisation des utilisateurs. Un modèle de stéréotype est en quelque sorte un « cas particulier » d'un modèle utilisateur. Pour preuve, le méta-modèle de stéréotype (figure 2.20) est identique (ou presque) à celui décrit plus tôt dans ce chapitre concernant un utilisateur. Signalons au passage qu'il est également permis pour un stéréotype de modéliser les contextes d'utilisation.

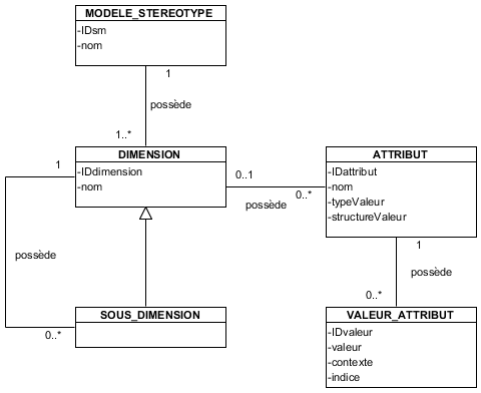



FIGURE 2.20: Le méta-modèle de stéréotypes est quasi-identique au méta-modèle utilisateur

### 2.5.4 Processus de modélisation d'un stéréotype

Le modèle de stéréotypes peut également être l'objet d'un processus visant à définir différentes étapes de son existence. La figure 2.21 décrit les différentes étapes. La première étape est logiquement la définition du stéréotype. La seconde est son exploitation. Parfois, son exploitation peut suggérer une réévaluation (grâce à différents mécanismes d'analyse de performances). Un stéréotype peut tout comme un modèle utilisateur, être en constante évolution au gré du flux d'informations qui l'alimente. Cependant, contrairement aux modèles utilisateurs, la redéfinition du stéréotype (réévaluation) a son importance, dû notamment au fait qu'on ne peut connaitre précisément les caractéristiques de la population utilisatrice du système.

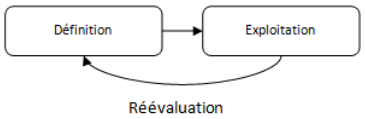



FIGURE 2.21: Schéma du processus de modélisation d'un stéréotype

Un stéréotype peut aussi être supprimé mais cela ne constitue pas une étape en tant que telle car il n'est pas destiné à être détruit. Son but est son exploitation.

Dans les sections suivantes, nous allons suivre ce schéma de processus afin d'aborder dans le détail les différents mécanismes associés à la modélisation des stéréotypes.

#### 2.5.4.1 Définition des stéréotypes

Similairement à la définition du modèle utilisateur, définir un stéréotype comprend trois activités. D'abord, la définition du domaine dans lequel il va être utilisé. Ensuite, la détermination d'une « architecture ». Enfin, une réflexion sur quelques aspects techniques propres à l'implémentation du modèle.

**Domaine** Définir un stéréotype n'est pas une tâche simple. Les stéréotypes peuvent être mal choisis, mal construits ou mal ciblés. C'est pourquoi une étude du domaine est essentielle afin de dégager de bons stéréotypes. Nous comprendrons dans le chapitre suivant que travailler avec de bons modèles de stéréotype augmente considérablement l'efficacité ainsi que la pertinence du système.

Un stéréotype étant par nature différent d'un modèle utilisateur, l'analyse du domaine le sera également. Le raisonnement est en effet tout autre, il faut non plus raisonner au niveau individuel mais au niveau « agrégé » c'est-à-dire sur un ensemble d'individus, une sous-population de la population entière.

Dans le cas d'un site internet, l'idée est de découvrir qui (quels groupes d'individus) visitera le site. En général, répondre à cette question n'apportera pas beaucoup d'éléments permettant de déterminer les stéréotypes. Pour un site vendant des appareils photos, les utilisateurs sont des amateurs de photographie. Trouver de bons stéréotypes nécessite de découvrir les caractéristiques pertinentes qui subdivisent la population globale des internautes en sous-populations. Le problème réside tant dans la découverte de caractéristiques marginalisantes que dans leur pertinence.

Dans l'annexe B nous proposons deux méthodologies aidant à la découverte de telles caractéristiques. Elles vont être illustrées par notre exemple concernant le site d'appareils photographiques.

Ces méthodologies ont pour but d'aider les gestionnaires à construire leurs stéréotypes de manière progressive. Cependant, elles ne prétendent en aucun cas remplacer une bonne analyse marketing et/ou statistique. L'ensemble de stéréotypes dégagés peut également être une bonne base comme « input » pour des outils de réévaluation de stéréotypes (voir section réévaluation des stéréotypes).

**Architecture** La seconde étape dans la création d'un stéréotype est la définition d'une structure, une organisation des données qui va permettre une meilleure lisibilité mais aussi une division sémantique qui sera utile à l'exploitation des données. Les principes discutés dans la partie concernant l'architecture

des modèles utilisateurs sont également d'actualité pour le modèle de stéréotype. Pour plus de détails, s'y reporter.

Deux éléments sont cependant à préciser. Premièrement, la décomposition en dimensions en comprend classiquement deux. Une dite « descriptive » et une dite « prédictive ». Nous en avons déjà discuté ci-dessus. Deuxièmement, un formalisme syntaxique permettant de représenter les intervalles devra être défini car jusqu'ici cette structure de données n'était pas prise en compte.

**Aspects techniques** Les principes discutés dans la partie concernant les aspects techniques des modèles utilisateurs sont également d'actualité pour le modèle de stéréotype. Pour plus de détails, s'y reporter.

#### 2.5.4.2 Exploitation des stéréotypes

Dans l'introduction sur les stéréotypes nous avons discuté des intérêts d'une telle approche. L'utilisation peut être multiple. Nous verrons dans le chapitre suivant que leur principale utilisation sera de classer les profils individuels dans ces stéréotypes. Les critères de classement seront les attributs du modèle utilisateur (de la partie descriptive). Nous appellerons l'algorithme exécutant ce classement « le matching ». Le but poursuivi étant d'enrichir les informations que l'on possède sur un utilisateur avec des informations dites de « prédictions » ou d' « intérêts ». Celles-ci concernent souvent directement le « business » du système. De plus, cela permet de raisonner sur un même ensemble d'attributs bien déterminés. Cet ensemble peut parfois être associé à une sémantique (via le nom de stéréotype par exemple).

#### 2.5.4.3 Réévaluation des stéréotypes

Par réévaluation des stéréotypes nous entendons les processus permettant d'adapter les stéréotypes en fonction des nouvelles informations obtenues. C'est un réel système d'apprentissage.

La réévaluation des stéréotypes peut se faire à deux niveaux. Tout d'abord, au niveau de la découpe en différents stéréotypes. Il arrive fréquemment que celle-ci soit mal effectuée parce qu'il est très ardu de dégager avec précision la population d'internaute, le public cible, leurs caractéristiques pertinentes et les besoins business. Un algorithme de réévaluation de stéréotype pourrait aider à ajuster ces découpes. Ensuite, la seconde réévaluation de stéréotype peut se faire au sein même d'un stéréotype,

où les attributs, leurs valeurs et leurs probabilités peuvent être adaptés. L'étude de tels algorithmes, complexes, ne fait pas partie du cadre de ce mémoire.

# Chapitre 3

# Système de personnalisation web

Après avoir posé les bases de la modélisation utilisateurs, nous allons, dans ce chapitre, d'abord découvrir ce qui se fait en termes de personnalisation web dans la littérature scientifique, mais également sur le web. Puis, nous présenterons un processus de personnalisation web "type" et discuterons des limites et inconvénients de la personnalisation web pour clôturer le chapitre.

## 3.1 Etat de l'art

Avec l'avènement des télécommunications et plus particulièrement d'internet, le rapport entre l'Homme et l'information s'est vu changé, et cela principalement par son accès et sa quantité. En effet, l'accès à l'information est de plus en plus direct. Nous pouvons observer que le chemin et le temps nécessaire pour obtenir une information ne cessent de diminuer. Force est également de constater que nombreux sont les sites web qui mettent à disposition une grande quantité d'informations. Le temps de consultation peut donc être relativement long et décourager l'internaute. Cela traduit le besoin qu'à l'Homme d'aller toujours plus vite. C'est dans ce cadre que la personnalisation web prend son sens. Elle va permettre un accès plus rapide et plus pertinent à l'information.

Nous distinguons deux niveaux de personnalisation : la personnalisation de contenu et la personnalisation d'interface (présentation). La personnalisation de contenu consiste à sélectionner parmi l'ensemble de contenus web (texte, liens ou multimédia), ceux qui sont les plus pertinents pour l'internaute. La personnalisation d'interface consiste à agencer la présentation, l'affichage en fonction des préférences et centres d'intérêts de l'internaute.

A. Pretschner et S. Gauch [38]dans leur étude des systèmes de personnalisation ont proposé deux catégories de système de personnalisation :

- « accès personnalisé aux contenus (portails web personnalisés) »
- « filtrage et estimation (journaux électroniques, les systèmes de recherche d'informations et services de recommandation d'éléments incluant la navigation) »

Selon S. Gauch, un système de personnalisation a pour objectif de « fournir à l'utilisateur une interface d'accès aux informations adaptée à ses besoins et ses centres d'intérêts » en générant des pages dont le contenu varie en fonction des préférences et des centres d'intérêts de l'internaute.

Un tel système est utilisé sur le site MyYahoo [1] . Comme les captures d'écran suivantes le montrent, MyYahoo propose à l'internaute de configurer ses données et ses intérêts manuellement.

1. http ://my.yahoo.com/

MyYahoo :

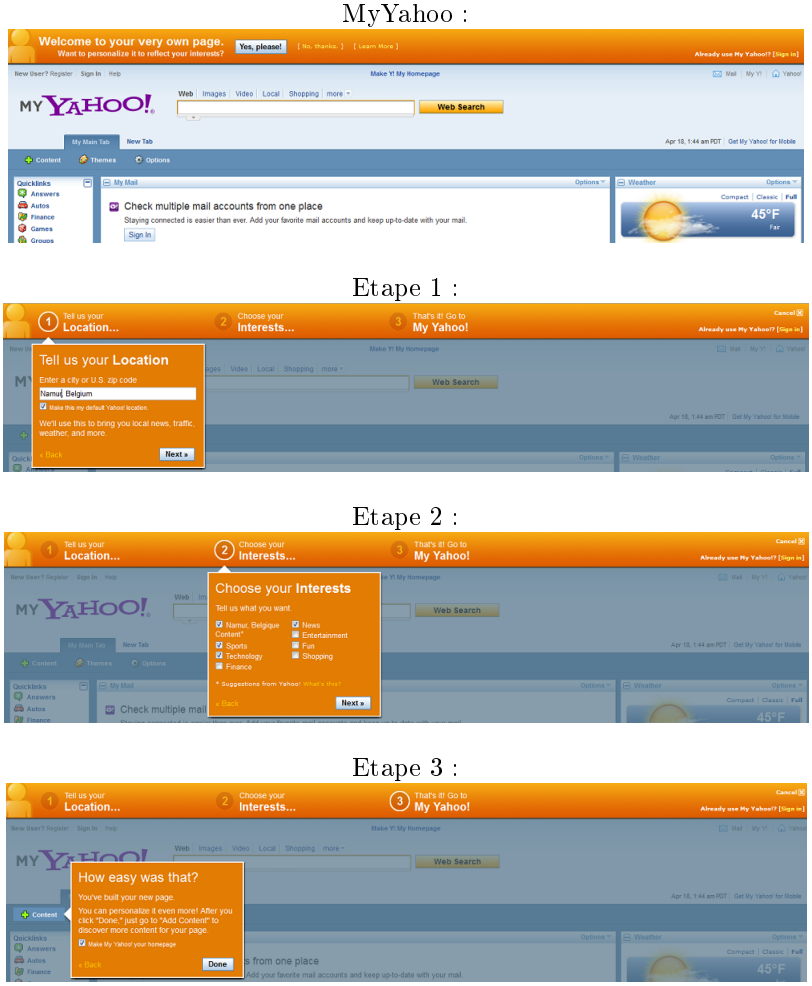



FIGURE 3.1: MyYahoo permet la personnalisation

Cette technique de personnalisation n'est que très peu décrite dans la littérature scientifique.

A. Pretschner et S. Gauch [37] ont fait une autre classification des systèmes de personnalisation. Selon eux, lorsqu'un internaute effectue une recherche, les mots-clés employés par ce dernier sont ambigus, empêchant dès lors le système de comprendre complètement la recherche. Pour palier à cela, ils proposent trois solutions : le ré-ordonnancement des résultats, le filtrage des résultats et l'extension des requêtes avec les données de l'internaute (par exemple ses centres d'intérêts).

### 3.1.1 Techniques de personnalisation courantes

Selon D. Kostadinov [20], le filtrage de résultats, le ré-ordonnancement des résultats et la recommandation d'éléments sont les techniques les plus couramment utilisées dans les systèmes de personnalisation. D. Kostadinov [20] présente brièvement ces trois méthodes.

La technique de filtrage de résultats va, dans un premier temps, exécuter la requête sans prendre en compte la personnalisation. Une fois la liste de résultats obtenue, les éléments les moins pertinents pour l'internaute vont être retirés. Un tel système est relativement simple à mettre en œuvre étant donné qu'aucun traitement de personnalisation n'est nécessaire en amont (au fournisseur d'informations). Cependant, le volume d'informations traité est plus important étant donné que l'on obtient tous les résultats de la requête initiale, y compris ceux qui ne sont pas pertinents. De plus, en général, le filtrage de résultats se fait coté client, entrainant un transfert important de données entre le serveur et le client. Un autre inconvénient est le risque de suppression définitive de résultats pertinents par le filtrage. Il n'y a pas de notion de poids. Un résultat fait partie ou non des résultats pertinents.

La technique de ré-ordonnancement des résultats va modifier l'ordre d'affichage des résultats en fonction de leur pertinence vis-à-vis de l'internaute. Cela se fait grâce à une fonction permettant de calculer le nouveau rang d'un élément de résultat. Encore une fois, le traitement de personnalisation ne se fait pas en amont, mais lorsque la liste complète des résultats a été obtenue. Le ré-ordonnancement possède les mêmes avantages et inconvénients que le filtrage, à l'exception de la non-exclusion de résultats. Tous les résultats sont présents, même les moins pertinents. Ceux-ci auront cependant tendance à se positionner à la fin de l'affichage.

La technique de recommandation va proposer des résultats en fonction des intérêts de l'internaute ou en se servant de l'expérience des autres internautes. Comme illustré ci-dessous, le site Amazon utilise la recommandation pour augmenter ses ventes et satisfaire ses clients.


FIGURE 3.2: Amazon propose à ses clients une réduction sur un pack contenant deux produits fréquemment achetés ensemble.


FIGURE 3.3: Amazon indique à ses clients ce qu'achètent les autres clients après avoir consulté cet article.

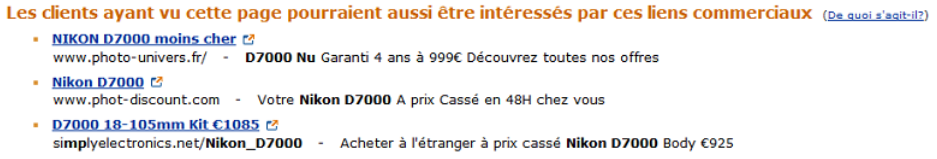



FIGURE 3.4: Amazon recommande un certain nombre de liens commerciaux à ses clients.

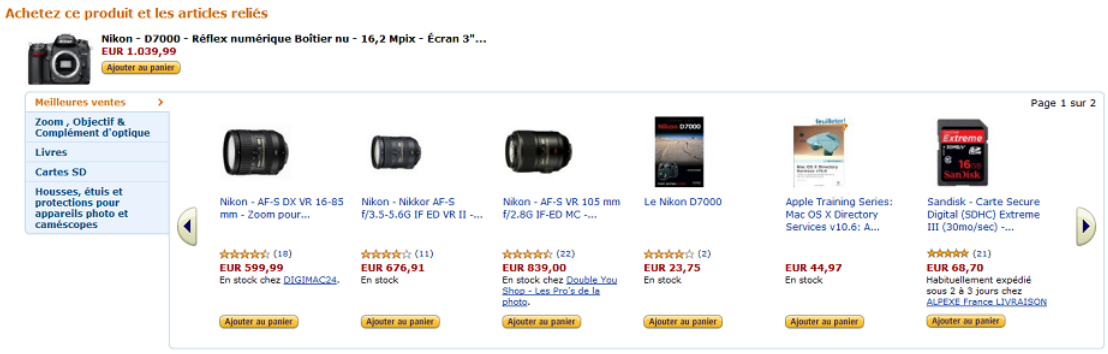



FIGURE 3.5: Amazon propose un certain nombre d'articles reliés.


FIGURE 3.6: Amazon recommande un certain nombre d'articles ayant également été achetés par les autres clients en plus du Nikon D7000 (article de la page en consultation).

La technique de recommandation est la plus utilisée.

D. Kostadinov [21] parle aussi de technique de reformulation de requêtes. Il y a deux approches de reformulation : la réécriture et l'enrichissement de requêtes. Réécrire une requête consiste à recréer la requête en prenant en compte les données, préférences de l'internaute. Enrichir une requête consiste à incorporer des éléments de préférences ou centres d'intérêts dans la requête sans la réécrire.

Remarquons que ces approches sont applicables pour les deux niveaux de personnalisation que sont la personnalisation de contenu et la personnalisation de l'interface graphique (de la présentation).

L. Ardissono a réalisé plusieurs études de cas de système de personnalisation. Notamment un système de personnalisation de news utilisant la modélisation d'utilisateurs et de stéréotypes [22]. Le contenu à présenter est sélectionné grâce a un « matching [2] » entre le modèle de l'internaute et un stéréotype. Le stéréotype comporte une partie « Predictions on interests » dans laquelle nous retrouvons différents domaines (économie, politique, sport, culture, technologie, etc.) avec une probabilité d'inté-

2. Algorithme permettant de classer un modèle utilisateur dans un ou plusieurs stéréotypes. Nous en reparlerons un peu plus loin dans ce chapitre.

rêts (figure 3.7). Par exemple, pour le domaine politique, un « Professional Financial reader » aura 70 pourcent de chance d'être fortement intéressé, 30 pourcent d'être moyennement intéressé et aucune chance d'être faiblement intéressé et nullement intéressé. Ce système permet également d'afficher plus ou moins de détails d'une news en fonction de l'internaute.

**PROFESSIONAL FINANCIAL READER**:
**profile**:
age: <20: 0; 20-25: 0.1; 26-35: 0.2; 36-45: 0.3; 46-65: 0.3; >65: 0.1
gender: M: 0.8; F: 0.2
job: manager: 0.57; self-trader: 0.3; self-employed: 0.05; ...; student: 0.01
job field: financial, banking, insurance: 0.8; politics, law, civil services: 0.14; ...
reason of connection: work: 0.9; personal: 0.1
hobbies: theatre: a lot: 0.1; some: 0.3; a little: 0.4; not at all: 0.2;
hobbies: following sports: a lot: 0.4; some: 0.3; a little: 0.2; not at all: 0.1;
...
**predictions on interests**:
economy: high: 1; medium: 0; low: 0; null: 0
politics: high: 0.7; medium: 0.3; low: 0; null: 0
sport: high: 0.2; medium: 0.4; low: 0.3; null: 0.1
culture: high: 0; medium: 0.2, low: 0.5; null: 0.3
technology: high: 0; medium: 0.3; low: 0.6; null: 0.1

FIGURE 3.7: Exemple de stéréotype "Professional Financial Reader" défini par L. Ardissono.

L. Ardissono a également travaillé sur un système de recommandation de programme TV intitulé « Personal Program Guide » (PPG)[23] et un système d'e-commerce [2], basés, ici aussi, sur la modélisation utilisateur et la modélisation de stéréotypes. Le but d'un tel système est d'aider l'utilisateur dans sa sélection de programme TV.

### 3.1.2 Sélection du contenu personnalisé

La plupart des systèmes de personnalisation présents sur le marché fonctionnent comme représenté à la figure 3.8.

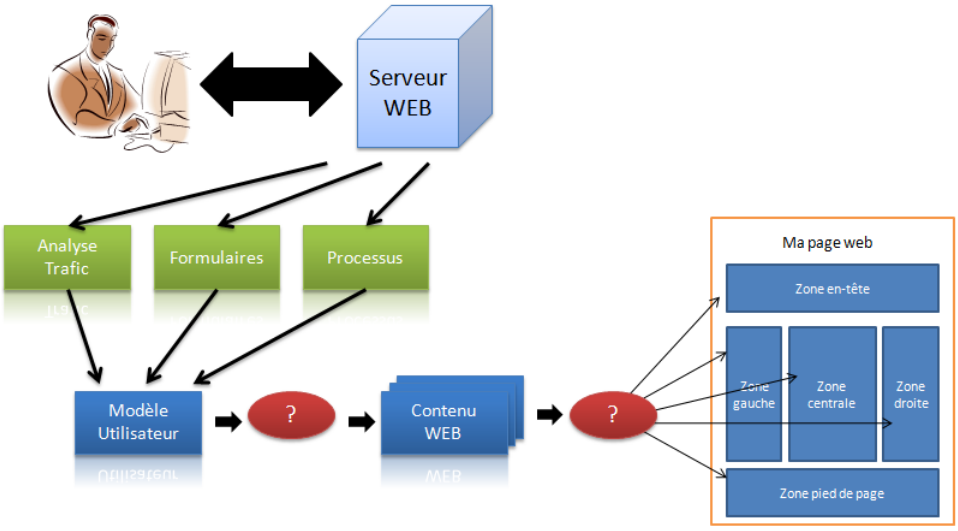



FIGURE 3.8: Schéma de fonctionnement d'un système de personnalisation.

Un internaute, lors de sa navigation, interagit avec un serveur web. Ce serveur web va obtenir un certain nombres d'informations au sujet de l'internaute via les différents mécanismes d'acquisition de données présentés au chapitre précédent (l'analyse de trafic [3] , les formulaires et les processus). Les données obtenues vont permettre d'alimenter le modèle utilisateur de l'internaute. Le système de personnalisation va ensuite devoir sélectionner du contenu à partir des informations stockées dans le modèle utilisateur. Ce contenu va également devoir être affecté à différentes zones réservées à l'affichage d'éléments personnalisés.

Il y a deux manières de déterminer quel sera le contenu le plus approprié pour l'utilisateur. La première, la plus triviale, se basera sur un certain nombre de règles prédéfinies. La seconde, quant à elle, utilisera la modélisation de stéréotypes. Voyons plus en détails comment fonctionnent ces deux techniques.

#### 3.1.2.1 Sélection orientée règles

Dans cette approche, l'algorithme de sélection du contenu se basera sur un ensemble de règles prédéfinies. Typiquement, supposons que le site possède trois contenus multimédia : une image d'un

3. Comprend l'analyse de variables, de paramètres, de cookies ainsi que l'analyse de l'interaction.

appareil photo professionnel, amateur et débutant. Supposons également que nous possédons le niveau en photographie de l'internaute dans son modèle utilisateur. Et enfin supposons un ensemble de règles :

– SI *le niveau de l'internaute est professionnel* ALORS *sélectionner l'image de l'appareil photo pro*

– SI *le niveau de l'internaute est amateur* ALORS *sélectionner l'image de l'appareil photo amateur*

– SI *le niveau de l'internaute est débutant* ALORS *sélectionner l'image de l'appareil photo débutant*

Une des trois images doit être sélectionnée pour être affichée sur la page web. Un système de personnalisation web orienté règles va examiner la liste de règles et appliquer celle qui correspond le mieux. En appliquant la première règle, l'image de l'appareil photo pro va être sélectionnée.

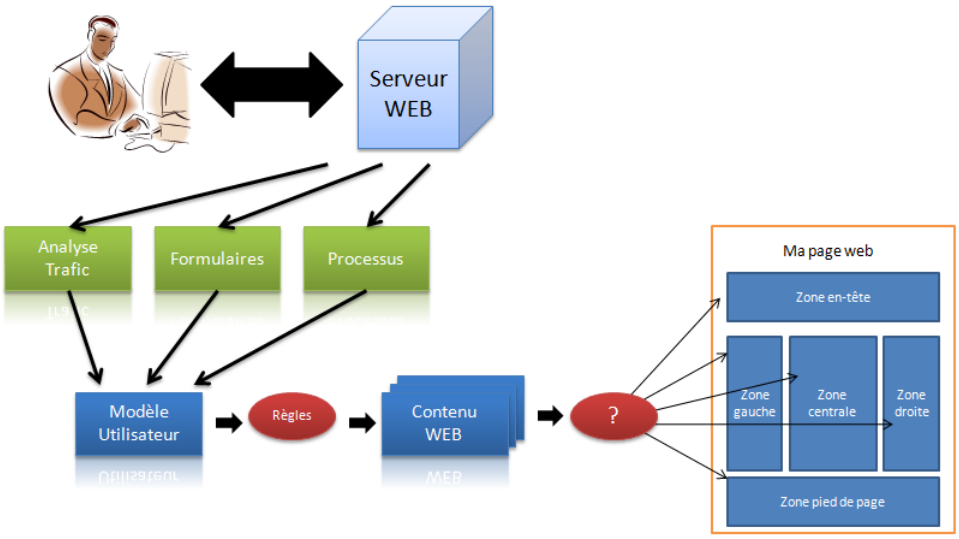



FIGURE 3.9: Schéma de fonctionnement d'un système de personnalisation orienté règles.

#### 3.1.2.2 Sélection orientée stéréotypes

Dans cette approche, l'algorithme de sélection du contenu se basera sur la modélisation de stéréotypes. Nous l'appelons « matching ». Le matching consiste à faire correspondre le modèle utilisateur avec un ou plusieurs stéréotype(s). Typiquement, si le système comporte trois stéréotypes « Photographe Pro », « Photographe Amateur » et « Photographe Débutant » et que dans le modèle utilisateur de l'internaute figure son niveau en photographie, en l'occurrence « pro », alors l'algorithme de matching va « classer », stéréotyper l'internaute dans la catégorie (stéréotype) « Photographe Pro ». Le système va également « matcher » les contenus web avec ces stéréotypes. Ainsi, par le biais du stéréo-

type, il existe une relation entre le modèle utilisateur et les contenus web. Notons que cette seconde opération de matching peut être faite au préalable afin de réduire le temps de personnalisation de la page. C'est bien ce qu'illustre la figure 3.10. Il y a deux niveaux de matching : le matching « Modèle Utilisateur – Stéréotype(s) » et le matching « Contenu Web – Stéréotype(s) ».

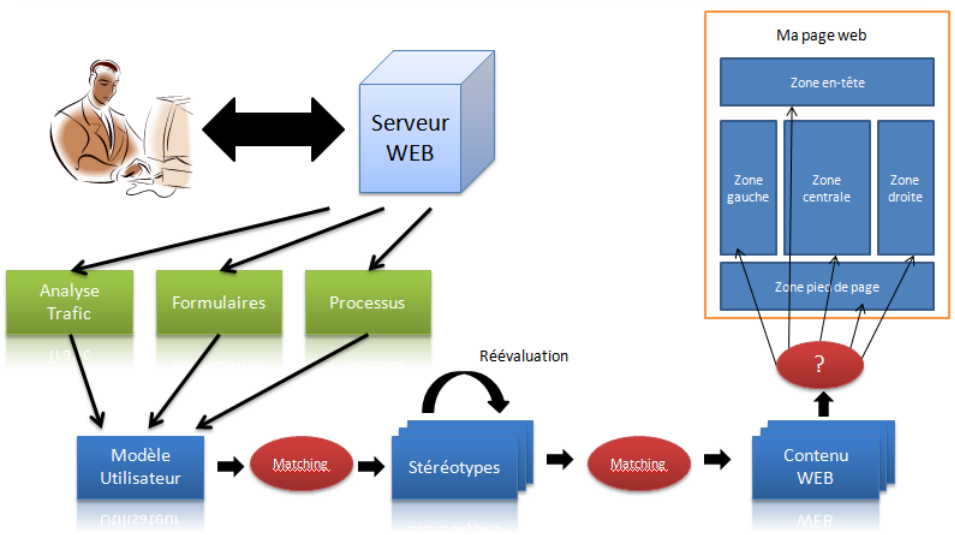



FIGURE 3.10: Schéma de fonctionnement d'un système de personnalisation orienté stéréotypes.

Il existe différents algorithmes de matching. Certains sont basés sur des méthodes bayesiennes, d'autres sur des techniques de datamining, d'autres encore sur des principes généraux de statistiques et de probabilités. L'étude de tels algorithmes sort du cadre de ce mémoire.

#### 3.1.2.3 Comparaison des deux approches

**Paramétrisation** Les deux solutions permettent une certaine paramétrisation du mécanisme de catégorisation. D'une part les algorithmes de matching permettent parfois d'agir sur divers paramètres, variables afin de modifier leur comportement. D'autre part, l'approche orientée « règles » permet par essence une paramétrisation accrue dans la mesure où les conditions et les actions d'une règle sont définies par le gestionnaire du site. L'ajustement du matching aux besoins est cependant plus limité que la création de règles. En effet, une règle peut facilement être créée par des acteurs gestionnaires du site.

Typiquement, les personnes s'occupant du marketing auront une plus grande aisance à définir des règles matérialisant leur stratégie qu'en jouant sur des variables d'un algorithme de matching. Le pouvoir de catégorisation des profils réside, dans ce cas, dans les mains de ces personnes. Ce sont donc ces personnes qui guident, déterminent la personnalisation.

Cette paramétrisation accrue de l'approche par règles entraine néanmoins un inconvénient majeur. En réalité, les besoins et les attentes des internautes et du site, ainsi que leurs caractéristiques ne sont pas toujours facilement perceptibles par un être humain. D'autant plus lorsque le nombre d'internautes, de données sur les internautes et de produits est important. Dans ce cas, les règles dégagées par le(s) gestionnaire(s) du site peuvent manquer d'efficacité, d'optimalité et entrainer une personnalisation peu conforme aux caractéristiques de l'internaute. Un algorithme de matching pourrait dès lors être plus performant lorsque le nombre de données à prendre en compte n'est pas gérable par l'homme.

**Automatisation** Certains algorithmes de matching sont directement opérationnels, d'autres ont besoin d'un temps d'apprentissage (manuel ou automatique). Après cet apprentissage éventuel, le matching se déroule de façon automatique. Bien sûr, il peut être nécessaire de définir de nouveaux stéréotypes. De même, les stéréotypes peuvent nécessiter un réajustement. Des algorithmes de réajustement et de suggestion de stéréotypes peuvent être utilisés permettant une certaines évolution automatique du système.

Quant aux règles, elles doivent être définies au début. Après quoi le système peut fonctionner de manière autonome en appliquant, à chaque fois, les règles définies. Les règles étant souvent plus spécifiques que les stéréotypes, cette approche parait moins automatisable. Cependant, similairement à l'approche par matching, des algorithmes de réajustement et de suggestion de règles peuvent être utilisés afin de permettre l'évolution du système de façon quasi-automatique.

**Complexité** La complexité de réalisation d'un algorithme de matching varie de l'un à l'autre. Un algorithme appliquant des règles peut paraitre simple à réaliser. Toutefois, les cas particuliers, où deux règles sont applicables ou aucunes, relèvent considérablement la difficulté de sa réalisation.

L'approche par règle comporte une étape de moins que celle par matching. Un système utilisant les règles comme sur la figure 3.9 n'a, a priori, pas besoin de travailler avec des stéréotypes. Cela diminue le temps de personnalisation mais également l'espace mémoire occupé.

## 3.2 Processus de personnalisation

Cette partie a pour but de décrire le processus « type » du système de personnalisation que nous voulons définir.

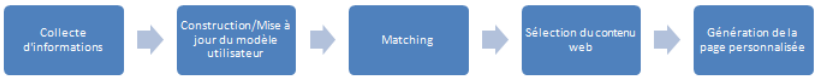



FIGURE 3.11: Processus de personnalisation web

1. Collecte d'informations sur l'internaute : pour en savoir plus sur l'internaute, le système va collecter le maximum d'informations au sujet de l'internaute.
2. Construction du modèle utilisateur : grâce aux informations collectées à l'étape précédente, le modèle utilisateur peut être construit.
3. Matching (trouver un stéréotype correspondant) : le modèle utilisateur construit à l'étape précédente va être stéréotypé grâce à un ensemble de stéréotypes préalablement définis (voir activités parallèles « Définition d'un stéréotype »). Le(s) stéréotype(s) qui correspond(ent) le mieux au modèle utilisateur de l'internaute va(vont) être sélectionné(s).
4. Sélection du contenu web à afficher sur base du stéréotype : sur base du(des) stéréotype(s) sélectionné(s), le contenu web à afficher sur la page va être choisi.
5. Génération de la page personnalisée : la page web va être construite et les zones destinées à la personnalisation vont être complétées par des contenus web liés au(x) stéréotype(s) de l'internaute.

**Activités parallèles**

- Réévaluation de stéréotypes : activité optionnelle consistant à ajuster les stéréotypes automatiquement, sur base des données collectées.
- Définition de la structure d'un modèle utilisateur : activité réalisée par le gestionnaire du site web consistant à définir la structure d'un modèle utilisateur que tous les modèles utilisateurs devront respecter. Par exemple, sans entrer dans les détails, le modèle utilisateur contient trois dimensions, la première, « données personnelles » comporte l'attribut nom, prénom et âge, la

seconde « données implicites » comprend l'attribut résolution d'écran, et la troisième « domaines d'intérêts » possède l'attribut niveau en photographie.

- Définition de la structure d'un stéréotype : activité réalisée par le gestionnaire du site web consistant à définir la structure d'un stéréotype que tous les stéréotypes devront respecter.
- Création d'un stéréotype : activité réalisée par le gestionnaire du site web consistant à définir un stéréotype, c'est-à-dire une catégorisation de profil.
- Lier les contenus web aux stéréotypes : activité consistant à associer des contenus web à des stéréotypes. Ainsi, possédant le stéréotype d'un modèle utilisateur, nous supposons que ces contenus associés intéressent l'internaute.

## 3.3 Limites et inconvénients de la personnalisation web

La personnalisation web, telle que définie jusqu'ici, ne possède pas que des avantages. Il peut survenir un certain nombre d'effets collatéraux pouvant être nuisibles à l'internaute. En voici un bref aperçu.

### 3.3.1 Temps de chargement

L'influence qu'un système de personnalisation web a sur le temps de chargement des pages est fortement liée à la technique de personnalisation web utilisée et à son architecture.

Un système basique de filtrage de résultats aura tendance à diminuer le temps de chargement puisque les résultats sont moins nombreux. Ceci à la condition que le temps d'exécution du filtrage ne vienne pas contrebalancer le gain de temps lié à la réduction du nombre de résultats.

Un système de personnalisation web implémentant le processus de personnalisation type présenté à la figure 3.11 pourrait allonger le temps de chargement étant donné qu'un simple site web sans personnalisation n'effectue pas les quatre premières étapes de cette figure.

En général, les systèmes de personnalisation ont tendance à augmenter légèrement le temps de chargement des pages étant donné qu'ils effectuent un certain nombre de traitements supplémentaires. Chaque système étant différent, cette tendance peut varier de l'un à l'autre.

### 3.3.2 Lourdeur supplémentaire liée à la saisie de données

Comme nous l'avons vu, un système de personnalisation se nourrit d'informations concernant l'internaute. Plus il en possède, plus il peut exercer une personnalisation affinée. Les concepteurs de sites web, incluant la personnalisation, pourraient donc vouloir mettre en place un maximum de mécanismes (tels que les formulaires HTML) permettant d'obtenir ces informations, et ceci au détriment de la navigabilité. En effet, l'internaute (dans sa navigation) se verrait dérangé par différents dispositifs de saisie de données afin de fournir des informations à son sujet.

Pour ne pas annihiler ses bénéfices, un système de personnalisation se doit d'être discret. Le concepteur du site web doit en être conscient.

### 3.3.3 L'apprentissage

Le profil d'un internaute se dresse rarement d'un seul jet, mais par l'accumulation d'informations concernant cet internaute. C'est un processus d'apprentissage non-limité dans le temps. Plus un utilisateur visite un site, plus les modèles utilisateurs derrière seront garnis et précis. A l'inverse, moins un internaute visite un site, moins les modèles utilisateurs sont complétés et moins la personnalisation est efficace. Par conséquent, lors des premieres intéractions de l'internaute sur un site web, les éléments personnalisés peuvent ne pas convenir.

### 3.3.4 Visibilité du contenu web

Le contenu web peut être présenté de différentes manières en fonction du système de personnalisation mis en place. Un tel système met en avant certains contenus par rapport à d'autres en jouant sur leur visibilité. Par visibilité nous entendons la facilité d'accès. Ainsi un contenu à visibilité médiocre pourrait être un contenu pour lequel son accès n'est pas aisé (par exemple, il faut parcourir une longue suite de liens pour l'atteindre). Voici les trois manières principales de présenter un contenu personnalisé dans un système de personnalisation.

#### 3.3.4.1 Visibilité accrue du contenu personnalisé et normale du contenu non-personnalisé

Cette technique[4] permet de mettre en avant le contenu personnalisé sans changer le niveau de visibilité du contenu non-personnalisé. Cela permet ainsi à l'internaute de pouvoir consulter très facilement du contenu que la personnalisation ne lui proposerait pas.

#### 3.3.4.2 Visibilité normale du contenu personnalisé et décrue du contenu non-personnalisé

Cette technique permet de mettre en avant le contenu personnalisé en diminuant le niveau de visibilité du contenu non-personnalisé. L'internaute pourrait présenter des diffcultés à trouver un contenu non-personnalisé.

#### 3.3.4.3 Visibilité du contenu personnalisé et invisibilité du contenu non-personnalisé

Cette technique permet de mettre en avant le contenu personnalisé par le non-affichage de tout élément non-personnalisé. Nous avons vu une telle manière de procéder précédemment appelée "filtrage". L'internaute ne dispose ici d'aucun moyen pour consulter le contenu non-personnalisé.

Nous constatons que moins la personnalisation accorde de visibilité aux contenus non-personnalisés, plus la personnalisation constitue un inconvénient pour l'internaute par le fait qu'il a accès (facilement) à un ensemble plus réduit de contenus.

### 3.3.5 Erreurs de personnalisation

Il arrive que le contenu personnalisé que propose le site web à un internaute ne corresponde pas aux attentes de l'internaute en termes de personnalisation. C'est ce que nous nommons les erreurs de personnalisation. Nous identifions trois causes d'erreurs. Les erreurs liées au système de personnalisation, les erreurs liées à une utilisation (navigation) multi-profils et les erreurs liées au modèle utilisateur.

#### 3.3.5.1 Erreurs liées au système de personnalisation

Le système de personnalisation, dans son fonctionnement, peut comporter des erreurs, voire des simplifications abusives (comme le matching trop réducteur d'un internaute dans un stéréotype). Cela

4. Technique utilisée par Amazon

peut engendrer une personnalisation erronée.

#### 3.3.5.2 Erreurs liées à une utilisation multi-profils

Parfois, il arrive qu'une personne navigue à la place d'une autre. Par exemple, cela se produit lorsqu'un site internet pratique la personnalisation sans étape de log-in, et que, sur une même machine, plusieurs internautes naviguent sur ce site. La construction du modèle utilisateur de l'internaute est alors biaisée. En effet, ces différents internautes ont des profils différents et le système n'a aucun moyen de distinguer le changement d'internaute. Toutes les informations collectées vont donc alimenter le même modèle utilisateur et la personnalisation sera dès lors plus que probablement erronée.

Un autre cas de figure similaire peut se produire. Imaginons qu'un site internet vend des produits de tous genres. Un internaute désireux de faire plaisir à un proche recherche sur ce site un cadeau susceptible d'intéresser ce proche, mais ne possède personnellement aucun intérêt envers ce présent. Pour se faire, il navigue sur le site dans les rubriques correspondant aux centres d'intérêts du futur bénéficiaire du cadeau. Le système de personnalisation enregistre alors ces centres d'intérets (qui ne sont pas les siens) et les associe au compte de l'internaute. La personnalisation est alors biaisée car le profil de l'utilisateur est erroné.

Idéalement, pour éviter ce genre de problèmes, l'internaute devrait avoir la possibilité de désactiver la personnalisation temporairement afin de ne pas collecter de données erronées.

#### 3.3.5.3 Erreurs liées au modèle utilisateur

La personnalisation d'un site internet peut également être erronée lorsqu'elle se base sur un modèle utilisateur d'un internaute ne correspondant pas à son profil. Cela peut se produire lorsque l'internaute introduit par inadvertance des fautes de frappes dans un formulaire de saisie, lorsqu'il fournit volontairement des informations incorrectes ou lorsque, pendant sa navigation, il exerce un comportement traduisant un intéret quelconque qui s'avère erroné. A cela s'ajoute le fait que certaines connaissances à propos de l'utilisateur n'ont plus lieu d'être et sont donc périmées.

## 3.4 Vers la qualité des informations web

Les limites et inconvénients présentés à la section précédente nous forcent à remettre en question l'utilisation de la personnalisation web. Au final, est-ce un réel avantage pour l'utilisateur de bénéficier de services de navigation personnalisés ou un désavantage de par la lourdeur qu'elle ajoute ? Tout dépend du système de personnalisation mis en place. Certains resteront discrets et efficaces, alors que d'autres se verront lourds et bancals. Au vu de nos précédentes constatations, il est clair qu'une majorité des échecs de personnalisation sont liés aux données concernant l'internaute, contenues dans le modèle utilisateur. Dans ce mémoire, nous essayons de proposer une solution à ce problème afin de permettre l'amélioration de la pertinence des résultats. Ceci nous amène à nous poser la question de la qualité des informations web, sujet du chapitre suivant.

# Chapitre 4

# La qualité des informations web

Maintenant que nous avons démystifié les systèmes de personnalisation, l'intérêt est d'améliorer la fiabilité de ces systèmes dans la justesse des résultats qu'ils vont fournir quant aux préférences et prédictions pour l'utilisateur concerné. Pour cela, dans ce chapitre nous allons mener une étude sur les qualités des informations qui sont collectées par de tels systèmes (il est donc bien question de la qualité des informations web). Cette étude a pour but de dégager un certain nombre de critères de qualité pertinents, qui seront exploités dans le chapitre suivant afin de permettre la conception d'une formule destinée à évaluer la confiance en une information.

Comme le propose R. Harrathi [14], il est important de faire la distinction entre qualités que l'internaute désirerait obtenir lors de sa navigation (par exemple : popularité, accessibilité, confidentialité, compréhensibilité, simplicité, temps de réponse, etc.) telles que l'étudie R. Harrathi et M. Bouzeghoub dans [13, 14, 26] et qualités que le système de personnalisation prendrait en compte afin de fournir un contenu plus adapté au profil de l'internaute. C'est cette seconde optique qui constitue notre cadre de recherche. On ne parle donc pas de la qualité du service ou même du contenu, mais bien de **la qualité des informations collectées sur l'utilisateur**.

Le chapitre est divisé en trois parties. La première étudiera la qualité des informations dans la littérature scientifique en présentant quelques grandes idées et résultats d'études. La seconde aura la tâche d'étudier les qualités ainsi que les problèmes de qualité présents dans le système de personnalisation que nous avons défini au chapitre précédent. Sur base des questions soulevées dans cette section, la troisième partie aura pour but de dégager un ensemble de qualités jugées pertinentes pour le SPW.

## 4.1 Qualité des informations dans la littérature scientifique

Comme l'indique R.Y. Wang [9], M. Jarke et Y. Vassiliou [16] et L. Berti [4], bien qu'étudiée depuis longtemps, la qualité des données a émergé il y a peu comme champs de recherche à part entière. Le problème majeur qui occupe les scientifiques de ce nouveau domaine d'étude est la définition stricto-sensu de « qualité ». Chacun tente de définir ce concept à sa manière. Ceci a pour conséquence de n'avoir aucun consensus sur la notion de qualité. Néanmoins, en pratique, la plupart des scientifiques admettent sans difficulté que ce concept ne réside pas tant sur une définition unanime, mais effectivement sur un ensemble de dimensions ou critères [40, 52, 44, 36].

R.Y. Wang et D.M. Strong [44, 52, 43] se sont penchés sur l'étude des qualités des données. Pour cela, ils ont, dans un premier temps, effectués une enquête visant à obtenir une liste exhaustive de 179 critères (attributs) de qualité de données potentielles. La figure 4.1 reprend cette liste.

| | | | |
|---|---|---|---|
| Ability to be Joined With | Ability to Download | Ability to Identify Errors | Ability to Upload |
| Acceptability | Access by Competition | Accessibility | Accuracy |
| Adaptability | Adequate Detail | Adequate Volume | Aestheticism |
| Age | Aggregatability | Alterability | Amount of Data |
| Auditable | Authority | Availability | Believability |
| Breadth of Data | Brevity | Certified Data | Clarity |
| Clarity of Origin | Clear Data Responsibility | Compactness | Compatibility |
| Competitive Edge | Completeness | Comprehensiveness | Compressibility |
| Concise | Conciseness | Confidentiality | Conformity |
| Consistency | Content | Context | Continuity |
| Convenience | Correctness | Corruption | Cost |
| Cost of Accuracy | Cost of Collection | Creativity | Critical |
| Current | Customizability | Data Hierarchy | Data Improves Efficiency |
| Data Overload | Definability | Dependability | Depth of Data |
| Detail | Detailed Source | Dispersed | Distinguishable Updated Files |
| Dynamic | Ease of Access | Ease of Comparison | Ease of Correlation |
| Ease of Data Exchange | Ease of Maintenance | Ease of Retrieval | Ease of Understanding |
| Ease of Update | Ease of Use | Easy to Change | Easy to Question |
| Efficiency | Endurance | Enlightening | Ergonomic |
| Error-Free | Expandability | Expense | Extendibility |
| Extensibility | Extent | Finalization | Flawlessness |
| Flexibility | Form of Presentation | Format | Integrity |
| Friendliness | Generality | Habit | Historical Compatibility |
| Importance | Inconsistencies | Integration | Integrity |
| Interactive | Interesting | Level of Abstraction | Level of Standardization |
| Localized | Logically Connected | Manageability | Manipulable |
| Measurable | Medium | Meets Requirements | Minimality |
| Modularity | Narrowly Defined | No lost information | Normality |
| Novelty | Objectivity | Optimality | Orderliness |
| Origin | Parsimony | Partitionability | Past Experience |
| Pedigree | Personalized | Pertinent | Portability |
| Preciseness | Precision | Proprietary Nature | Purpose |
| Quantity | Rationality | Redundancy | Regularity of Format |
| Relevance | Reliability | Repetitive | Reproducibility |
| Reputation | Resolution of Graphics | Responsibility | Retrievability |
| Revealing | Reviewability | Rigidity | Robustness |
| Scope of Info | Secrecy | Security | Self-Correcting |
| Semantic Interpretation | Semantics | Size | Source |
| Specificity | Speed | Stability | Storage |
| *Synchronization* | *Time-independence* | *Timeliness* | *Traceable* |
| Translatable | Transportability | Unambiguity | Unbiased |
| Understandable | Uniqueness | Unorganized | Up-to-Date |
| Usable | Usefulness | User Friendly | Valid |
| Value | Variability | Variety | Verifiable |
| Volatility | Well-Documented | Well-Presented | |

FIGURE 4.1: Liste de 179 critères de qualité de données résultant d'une étude de R.Y. Wang et D.M. Strong

Ensuite, ils ont tenté de classer ces attributs de qualité par importance afin de dégager un ensemble de dimensions qui serviront à définir un framework conceptuel de qualité de données. Wang et Strong expliquent que leur motivation provenait notamment d'un réel problème de qualité des données dans l'industrie. Les bases de données des entreprises ne sont en effet pas « *error-free* ». Ils constatent que cela a un impact social et économique non-négligeable. C'est pourquoi, ils ont voulu développer un framework « *qui capture les aspects des qualités des données qui sont importants pour les données des consommateurs* ». Etant orienté « *qualité des données perçue par les utilisateurs* », il n'est pas

pertinent de décrire ce framework dans ce mémoire. Comme cité au début de ce chapitre, nous désirons évaluer la confiance que l'on a des données que l'on possède au sujet des utilisateurs et non évaluer ce que ces derniers attendent d'une donnée ou d'un service quelconque. Cependant, la première partie de leur étude est intéressante dans la mesure où elle a permis de dégager une liste exhaustive des qualités d'une donnée.

Beaucoup d'autres études ont été menées dans ce domaine. Notamment celles de B. Pernici et M. Scannapieco [36] visant à définir un framework de gestion de la qualité (« *quality management framework* ») permettant notamment l'évaluation de cette qualité ; Missi [32] étudiant l'impact de la qualité des données et l'intégration de données ; M. Bouzeghoub et V. Peralta [6] définissant un framework d'analyse de la « fraicheur » (« freshness ») des données ; E. Stoops [48] concernant la mise à jour des données dans la cadre d'un système de personnalisation de contenu.

## 4.2 Qualité des informations au sein d'un système de personnalisation

Comme nous l'avons vu, le système de personnalisation défini jusqu'à présent se contente de récolter un maximum d'informations de tout genre au sujet de l'internaute afin de personnaliser les pages web qu'il consulte lors de sa navigation en fonction de ses préférences, habitudes, centre d'intérêts. Ce système utilise aveuglément toutes ces données sans prendre en compte leur pertinence, leur confiance. Il raisonne donc sur une sorte d'incertitude liée aux données. L'intérêt de cette section est d'identifier à quels moments et dans quelles situations de ce processus de personnalisation, la qualité des informations fait défaut. Cela nous permettra, dans la section suivante de ce chapitre, de mieux cerner les critères de qualité clés à considérer, et, dans les deux chapitres suivants, d'entrevoir comment un système permettant d'évaluer ces critères pourrait être élaboré et intégré au sein de ce système de personnalisation.

Il y a un certain nombre de problèmes liés à la qualité des données. Les informations collectées et utilisées par le système sont, comme nous l'avons vu, précaire. Nous identifions trois formes de précarité des informations.

### 4.2.1 Précarité d'une information liée au temps

Une information au sujet d'un utilisateur peut ne pas être indépendante du temps. Pour la plupart, elles ne sont fidèles à la réalité que durant un certain laps de temps. Pour preuve, lorsqu'un internaute indique au système son numéro de téléphone, on peut raisonnablement accepter que cette information soit « à jour ». Pourtant, trente ans plus tard, il y a de forte chance que ce numéro soit obsolète. Cette information n'est désormais plus à jour. A partir de quel moment une donnée n'est-elle plus considérée comme « à jour » ? En fait, d'une manière générale et naïve, nous n'avons pas de moyen de déterminer véritablement si à un instant donné, une information est à jour sans demander de confirmation de la part de cet utilisateur. Dès lors, dès l'instant où l'information est actualisée, nous perdons en « certitude » que cette donnée soit actuelle. Autrement dit, plus l'information est vieille (moins à jour), plus nous sommes dans l'incertitude qu'elle soit correcte. D'où l'idée de « précarité d'une information liée au temps ».

Une information possède donc une vitesse de « vieillissement ». Nous appellerons ce «vieillissement» **dépréciation temporelle**, c'est-à-dire, la perte de valeur (de certitude) de l'information liée à l'écoulement du temps. La difficulté est que chaque donnée est susceptible de posséder sa propre vitesse de dépréciation temporelle. Par exemple la résolution d'écran change en principe plus régulièrement que l'adresse postale.

Il existe aussi des données non-variables dans le temps (typiquement, la date de naissance ou le numéro de registre national). Ces données sont dites « fixes ».

Notons les travaux d'E. Stoops [48] et de M. Bouzeghoub [6] relatif au problème de « fraicheur » des données dans les systèmes d'information.

### 4.2.2 Précarité d'une information liée à la méthode d'acquisition

Nous avons vu trois manières d'obtenir des informations au sujet d'un internaute : la réception, la capture et la déduction. Voyons en quoi elles peuvent introduire un biais dans l'information par rapport à la conformité de celle-ci à la réalité.

#### La réception

La réception se produit lorsqu'un internaute envoie des informations au système via un formulaire web . Le biais pourrait venir du fait que le formulaire ne transmet pas (ou transmet mal) ce que l'utilisateur a introduit. D'autres facteurs influencent également la fiabilité d'un formulaire. Est-il muni de mécanismes de vérification syntaxique et sémantique, de mécanismes de vérification de complétude ou de mécanismes d'aide à la saisie ? Sa conception favorise-t-elle la bonne saisie des informations (notamment, a-t-on tendance à faire beaucoup ou peu de fautes de frappe avec ce formulaire ?) ?

#### La capture

La capture se produit lorsque le système intercepte les informations présentes dans les variables du navigateur internet. Le navigateur fournissant ces données est-il fiable/reconnu/sécurisé ? Dans le cas contraire, rien ne garanti que les informations obtenues par un tel procédé soient correctes.

#### La déduction

La déduction se produit lorsque le système déduit implicitement des informations de la navigation de l'internaute (sans qu'il les fournisse explicitement). Ce procédé tente donc de déduire des données sur base d'hypothèses faites par le système. Ces hypothèses peuvent être erronées, ce qui dans ce cas produira des conclusions fausses. Par exemple, faisons l'hypothèse qu'un internaute soit intéressé par un contenu sur lequel il clique. Lors de sa navigation, cet internaute clique sur tous les liens, un par un, afin d'explorer le site. Dans ce cas, cette hypothèse s'avère incorrecte car l'internaute pourrait cliquer sur un contenu qui représente peu d'intérêt à ses yeux. Ou bien, il pourrait cliquer sur un lien par inadvertance, rendant également cette hypothèse inexacte. Remarquons aussi qu'un comportement détecté peut parfois faire l'objet de plusieurs interprétations, déductions, avec encore une fois pour conséquence des résultats erronés.

### 4.2.3 Précarité d'une information liée à l'utilisateur

A priori, nous aurions tendance à considérer l'utilisateur comme fiable. En effet, qui est plus au courant de son profil que lui-même ? Pourtant, il influe sur la confiance que l'on a en une information.

Certains internautes peuvent être considérés comme plus fiables que d'autres. Il faut distinguer deux cas :

- L'utilisateur introduit une valeur erronée de manière involontaire.

  Typiquement, cela se produit lorsqu'en introduisant des données dans un formulaire, l'utilisateur commet une faute de frappe. Comme nous l'avons vu au chapitre 2, il existe des mécanismes (auto-complétion, vérification syntaxique, vérification sémantique, etc.) permettant de résoudre partiellement ce problème.
- L'utilisateur introduit une valeur erronée de manière volontaire.

  Il est imaginable que dans certaines situations l'utilisateur ne veuille pas fournir des données qu'il considère comme sensibles. Afin d'éviter de fournir de telles informations, il est envisageable qu'il introduise intentionnellement des informations erronées.

Il serait intéressant d'introduire au sein du système de personnalisation un module permettant de prendre en compte la fiabilité de l'utilisateur automatiquement sur base de ces deux aspects. Le système travaillerait avec un indice de fiabilité d'un utilisateur résultant de l'apprentissage de ses actions précédentes. Le sujet de l'apprentissage étant trop complexe pour être abordé brièvement et intégré au sein du système d'évaluation de la qualité des informations web (que nous allons aborder dans le chapitre suivant), il sera laissé à d'éventuels travaux ultérieurs visant à améliorer l'indice de confiance.

Au chapitre suivant nous définirons le concept de « source de données » englobant à la fois l'aspect « méthode d'acquisition » de l'information et l'aspect comportemental de l'utilisateur.

Les trois types de précarité d'une information qui viennent d'être discuté démontrent clairement le défaut de certitude, de confiance auquel fait face une donnée concernant un utilisateur du système de personnalisation défini au chapitre précédent. Le chapitre suivant tentera de proposer une solution à ce problème de manque de qualité en calculant un indice de confiance pour chaque donnée contenue dans le modèle utilisateur.

## 4.3 Sélection des qualités pertinentes pour le sysème de personnalisation

Cette section va tenter d'identifier les critères de qualité qui joueront un rôle dans l'évaluation de la confiance d'une information. Il faut garder à l'esprit que d'une part le système ne possède qu'un nombre restreint d'informations et d'autre part il ne permet pas de tout faire. Ceci à pour conséquence

qu'un certain nombre de critères de qualité sont non-évaluables et seront donc abandonnés.

### 4.3.1 Première sélection de critères de qualité d'une information

Comme nous l'avons vu, une multitude de critères de qualité pour une information sont envisageables. Pour preuve, R. Y. Wang et D. M. Strong [52, 44, 43] en ont identifiés 179. Afin d'aller droit au but, nous allons directement sélectionner quelques critères de qualité clés et en discuter :

- A jour : l'information est-elle actuelle ?
- Correcte, exacte, valide : l'information est-elle conforme à la réalité ?
- Complète : l'information est-elle intégrale ?
- Concise : l'information est-elle sans fioriture ?
- Non-ambigüe : l'information est-elle interprétable d'une seule manière ?
- Vérifiable : l'information peut-elle être vérifiée ?
- Compréhensible : l'information est-elle compréhensible ?
- Utile, pas de bruit, pertinente : l'information correspond-t-elle bien et de manière unique à ce qu'on attendait ?

### 4.3.2 Critères de qualité non-évaluables

Parmi cette première sélection de critères, il est impossible (à l'heure actuelle) pour le système défini au chapitre précédent d'évaluer un bon nombre d'entre-deux.

Le système est incapable de vérifier, évaluer, estimer ces critères :

- Complétude : rien ne peut garantir que l'utilisateur entre toutes les informations
- Concision : à l'heure actuelle, il est difficile pour un système de se prononcer sur le degré de concision d'une information car pour cela, il devrait pouvoir analyser sa sémantique.
- Non-ambigüité : à l'heure actuelle, il est difficile pour un système de se prononcer sur le degré d'ambigüité d'une information car pour cela, il devrait pouvoir analyser sa sémantique.
- Vérifiabilité : un système peut, à l'heure actuelle, vérifier une information mais pas se prononcer sur le degré de vérifiabilité d'une information.
- Compréhensibilité : à l'heure actuelle, il est difficile pour un système de se prononcer sur le degré de compréhensibilité d'une information car pour cela, il devrait pouvoir analyser sa séman-

tique. De plus, ce critère importe peu car il ne sera pas demandé au système de comprendre les informations qu'il traite.

– Utilité : Evaluer l'utilité dépend de l'utilisation que l'on compte faire d'une information. Cela dépend donc du site ou du gestionnaire du site. De plus, une information peut ne pas être utile dans un premier temps, et l'être par après.

Le but du système de qualité que nous voulons concevoir sera d'améliorer la pertinence de la catégorisation d'un utilisateur dans un stéréotype. Dès lors, le fait qu'une information soit concise, vérifiable, compréhensible et utile importe peu. L'essentiel est donc de savoir si elle est correcte ou non. Il est cependant vrai qu'une information qui n'est pas complète ou ambigüe peut influencer ce critère.

### 4.3.3 Seconde sélection de critères de qualité d'une information

En élaguant les critères non-évaluables et/ou non indispensables de la première sélection nous en obtenons une seconde. Celle-ci est composée de deux critères : « A jour » et « Correct ». Nous nommerons désormais ces critères respectivement « Freshness » et « Correctness ». Ils seront à la base du calcul de l'indice de confiance présenté au chapitre suivant.

# Chapitre 5

# Définition d'un indice de confiance

Lors du précédent chapitre, nous avons mené une brève étude concernant les qualités des informations web. Nous en avons conclu que seules deux qualités sont pertinentes et intéressantes à prendre en compte : la correctness et la freshness. Ce chapitre a pour objectif de tenter de définir un indice de confiance pour les informations web qui sont traitées. Il s'inscrit donc dans la continuité des chapitres précédents. Rappelons que par confiance nous entendons "l'assurance que la donnée soit correcte".

La méthodologie employée comportera deux grandes étapes. La première sera la définition d'une sous-formule pour chacune des deux qualités prônées. Celles-ci auront le rôle d'effectuer la transformation des informations reçues (input) en indice (output) dont la valeur devra refléter au mieux la sémantique de leur qualité associée. La seconde étape tentera d'agréger les deux indices issus de la première afin d'obtenir un indice de confiance global en une information. Parallèlement à cela, lorsque nous sommes en présence de plusieurs valeurs contradictoires (comme nous les verrons à la section 5.2.1.1 « Découverte des éléments clés »), nous discuterons des critères et conditions de choix permettant au système de décider laquelle (parmi elles) sera considérée. Tout au long de ce chapitre, nous adopterons une démarche progressive et notamment, nous avancerons par expérimentation tel que présenté à la figure 5.1. A chaque fois une hypothétique solution sera supposée, nous la vérifierons grâce à une table précédemment définie décrivant les effets désirés. Dans le cas où les tests ne sont pas concluants, la solution devra être peaufinée. Cette démarche à pour but de cerner pourquoi tel ou tel choix fut établi, plutôt que d'exposer directement la formule finale et de tenter de la comprendre superficiellement.

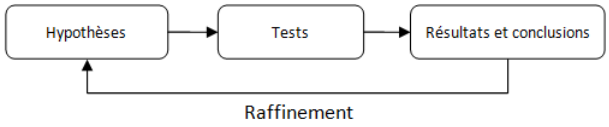



FIGURE 5.1: Démarche d'expérimentation employée dans ce chapitre

Rappelons que le système de personnalisation que nous avons étudié sera le cadre dans lequel l'évaluation des qualités informationnelles va s'établir. Nous avons spécifié un module que nous avons appelé « Quality Assessment » (figure 5.2). Celui-ci va recevoir en entrée un certain contenu informatif et devra en ressortir un indice de confiance (IC) compris dans l'intervalle [0,1]. Plus cet indice s'approchera de l'unité, plus la confiance en la donnée sera solide. A l'inverse, un indice avoisinant le 0 indiquera une confiance médiocre. Nous pouvons nous demander quel contenu est sujet à une évaluation de la confiance ? Quelles sont les inputs de ce module ? Cette question était, jusqu'à présent, volontairement laissée en suspend. Pour répondre à cette interrogation nous devrons dans en premier temps déterminer quels sont les éléments du modèle utilisateur intéressant pour une étude de la qualité. C'est ce qui nous occupera dans la première partie de ce chapitre. Ensuite, il est nécessaire de savoir quelles autres informations peuvent être utiles pour calculer les qualités. Ceci se fera au cas par cas, pour chaque qualité.

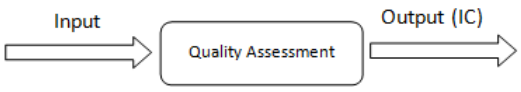



FIGURE 5.2: Schéma du module "Quality Assessment"

## 5.1 Informations à évaluer (inputs)

Quelles informations, présentes dans le modèle utilisateur, peuvent être évaluées ? Les dimensions et sous dimensions sont inhérentes au modèle, ce sont des informations structurelles non spécifiques à chaque utilisateur. Il ne serait pas pertinent de les inclure dans le calcul de qualité. De même, la liste des attributs est supposée prédéfinie par le gestionnaire du site. Or, nous voulons examiner la qualité des données acquises par le système de personnalisation, destinées à nourrir le modèle utilisateur, non pas la définition du modèle. De fait, l'intérêt réel réside au niveau des valeurs des attributs. Pour

chaque attribut, le module devra être capable d'apprécier la confiance de sa ou ses valeur(s). Pour cela, il recevra en entrée, une partie de l'instance du modèle d'un utilisateur pour lequel il est nécessaire d'évaluer la qualité. Typiquement cette partie sera les valeurs d'un attribut.

Nous le verrons, le méta-modèle utilisateur défini au chapitre 2 (figure 5.3) n'intègre pas encore assez d'informations permettant l'évaluation de la qualité. Quelques retouches légères viendront parachever ce modèle au fur et à mesure de notre réflexion.

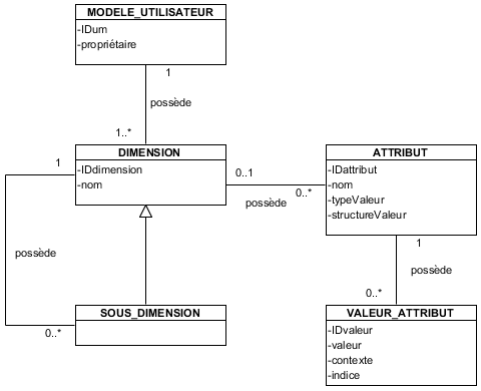



FIGURE 5.3: Méta-modèle utilisateur

## 5.2 Indice de confiance (output)

L'indice de confiance, nombre compris entre 0 et 1, doit indiquer le niveau de crédibilité, de confiance que l'on a d'une valeur d'un attribut. Par exemple, un attribut « Adresse », possède les valeurs « 20, rue de Fer Namur » si le contexte est « Maison » et « 50, rue de l'Ange Namur » si le contexte est « Travail ». Considérons uniquement la valeur liée au contexte "Maison". Nous avons conclu au précédent chapitre qu'il était possible et pertinent (dans le cadre de la personnalisation web) d'évaluer ces données sur base de deux aspects (qualités) : la correctness (à quel point une information est-elle exacte ?) et la freshness (à quel point une information est-elle à jour ?). L'objectif est de traiter séparément l'aspect « correct » d'une donnée indépendamment du temps (c'est-à-dire de considérer la donnée au temps $t_0$ d'acquisition). Ensuite, une fois cette première évaluation opérée, nous viendrons intégrer la prise en compte de l'aspect temporel grâce au calcul de la freshness. Ce module de qualité va donc effectuer une évaluation d'une information (par exemple l'adresse de l'internaute) pour ces deux qualités. Il en ressortira deux nombres appartenant à l'intervalle [0,1]. En les agrégeant, nous obtiendrons ce que nous

cherchons, c'est-à-dire, l'indice de confiance pour la valeur « 20, rue de Fer Namur » de l'attribut « Adresse ».

Nous allons à présent attaquer la définition des formules de Correctness, de Freshness et de l'indice de confiance. Commençons par cette première.

### 5.2.1 Correctness

#### 5.2.1.1 Découverte des éléments clés

**Fiabilité d'une source** Quelles informations pourraient être nécessaires pour déterminer l'exactitude d'une donnée ? Raisonnons sur base d'un exemple, l'attribut « nom » d'un utilisateur. Supposons qu'une source de données, en l'occurrence l'internaute via un formulaire d'inscription HTML, indique que cet attribut prend la valeur « Dupont ». Quelles sont les éléments dont nous disposons, qui pourraient nous aider à penser que l'internaute s'appelle bel et bien « Dupont » ? Naïvement, avec le peu d'information que nous possédons, nous avons le sentiment que la donnée entrée tend à être juste. Pourtant, il n'y a rien de fondé. Comment expliquer cela ? Analysons comment nous, être humain, fonctionnons pour évaluer la pertinence d'une information. Cela nous permettra de faire reproduire ce cheminement par un système automatique d'évaluation de la pertinence d'une donnée. Pour ce faire dressons les connaissances que nous pouvons tirer de cette supposition (SITUATION 1) :

SITUATION 1

- Le nom de l'internaute est « Dupont »
- C'est l'internaute, lui-même, qui l'a indiqué
- Via un formulaire d'inscription HTML

Et considérons un autre exemple, totalement opposé :

SITUATION 2

- Le nom de l'internaute est « Ghklrutdmlc »
- Donnée entrée par « AnonymousHacker » (donc pas par l'internaute lui-même)
- Via le champ « nom » d'un formulaire d'identification (sans inscription préalable) d'un chat non réputé

S'il était demandé de choisir la situation dans laquelle nous avons le plus confiance en le nom de l'internaute, qu'en ressortirait-il ? A l'unanimité, la situation 1 ! Pourquoi ?

Premièrement, « Dupont » (situation 1) est un nom courant, déjà vu et pas trop marginal, il parait donc plausible. Le nom « Ghklrutdmlc » de la deuxième situation parait beaucoup moins pertinent ! Malheureusement, déjà, le système montre ses limites. En effet, vérifier qu'une information est courante n'est pas évident à réaliser s'il n'existe pas au préalable une base de connaissances. De plus, comment un système pourrait s'assurer qu'une donnée n'est pas « trop » marginale ? Des algorithmes opérant un calcul de « distance » syntaxique existent [34, 10, 24]. Néanmoins, qu'en est-il de la distance sémantique ? Et comment agréger les deux ? Ceci sort du cadre de ce mémoire et pourrait éventuellement faire l'objet d'améliorations ultérieures.

Deuxièmement, le fait que cette donnée fut entrée par l'internaute même (situation 1) lui accorde une certaine confiance. Effectivement, si le nom avait été saisi par une tierce personne inconnue, comme dans la seconde situation, la pertinence aurait été moindre. De plus, la personne ayant entré le nom dans la situation 2 n'est en général pas synonyme de confiance (« Anonymous » + « Hacker »). Encore un problème lié à la sémantique, mais en rapport aux noms de sources cette fois.

Enfin, le moyen utilisé pour saisir les données contribue également à se forger une idée sur la pertinence. Pour preuve, un formulaire d'inscription de type HTML est utilisé dans la situation 1, alors que, pour la situation 2, il s'agit d'un formulaire d'identification sur un chat. Un formulaire d'inscription parait, a priori, plus fiable qu'un autre (pour un même site). En effet, l'enregistrement est une étape considérée comme importante dans le cycle de vie d'un compte sur un site internet. C'est moins le cas pour un formulaire de sondage ou pour une enquête par exemple. Bien sûr, la fiabilité de l'inscription varie d'un site à l'autre (site gouvernemental VS site de jeux flash par exemple). Concernant la façon d'introduire les données, la situation 2 révèle clairement une pertinence inférieure à la première. Non seulement par le sérieux du formulaire, mais aussi par le sens accordé au champ « nom » du formulaire d'un chat. Pour beaucoup dans un chat, lorsque l'on demande d'introduire un nom, les gens comprennent (ou veulent comprendre) « pseudonyme ». Ceci a pour incidence de faire dégringoler le peu de confiance que l'on avait dans la source de données.

De cette petite étude de cas, nous pouvons déduire qu'un facteur clé pour l'estimation de la correctness est la confiance que l'on peut avoir en une source de données (que nous appellerons « fiabilité »). Cette fiabilité doit prendre en compte à la fois la manière d'obtenir l'information (par exemple via un formulaire HTML) ainsi que l'individu ou le système qui fourni cette information. Dès lors, un nouveau problème se pose : comment évaluer la fiabilité d'une source ? La fiabilité d'une source s'exprimera par

une valeur comprise entre 0 et 1, où 1 correspondra à une fiabilité maximale de la source et 0 à une fiabilité minimale. Cette valeur sera issue de l'agrégation (par une moyenne pondérée) de deux « sous » fiabilités : la fiabilité de la méthode d'acquisition et la fiabilité d'un utilisateur.

L'évaluation de la fiabilité d'une méthode d'acquisition sera laissée au gestionnaire du système de personnalisation. En effet, la multiplicité des sources de données et l'infinité de systèmes de saisie pouvant voir le jour, rend cette tâche difficilement réalisable par un système. Il faudra néanmoins veiller à ce que le gestionnaire ne « surestime » pas cette fiabilité. Par exemple, la saisie de données par un formulaire n'est pas un moyen de totale fiabilité. En effet, des fautes de frappes peuvent facilement se glisser. Cette évaluation (par le gestionnaire) doit donc se faire de façon méticuleuse et avisée.

L'évaluation de la fiabilité d'un utilisateur quant à elle, est un problème qui peut se résoudre en utilisant des techniques d'apprentissages [7, 28]. Ce problème est trop complexe pour être abordé dans le cadre de ce mémoire.

Il existe d'autres facteurs jouant un rôle prépondérant dans la détermination de l'exactitude d'une donnée. Nous allons tenter de les découvrir ci-dessous.

**Multiplicités des occurrences** Tout d'abord, il serait logique d'observer un accroissement de la correctness lorsqu'un internaute introduit à plusieurs reprises les mêmes données via un même formulaire A. Pour reprendre l'exemple du « nom », cela correspond au cas où l'utilisateur remplit un certain nombre de fois le même formulaire en indiquant qu'il s'appelle « Dupont ». A force, il y a de plus fortes chances que ce nom soit exact, puisque la valeur a été confirmée. Nous appellerons ce phénomène le renforcement de la confiance liée à la « multiplicité des occurrences » d'une source. Cette multiplicité des occurrences se définit comme étant la répétition de saisies identiques issues d'*une même* méthode d'acquisition de données. En un mot, une source peut posséder plusieurs occurrences. Dans la formule de la correctness nous devrons donc intégrer un paramètre que l'on appellera simplement « nombre d'occurrences ». Celui-ci prendra ses valeurs dans l'ensemble des entiers naturels.

**Multiplicités des sources** De la même manière, si deux sources différentes viennent alimenter le modèle utilisateur avec une même valeur, nous aurions tendance à lui accorder plus de crédit que s'il y en avait qu'une. En effet, la « croyance » en la valeur introduite par la première source est renforcée par le fait que la seconde possède la même valeur. Cette seconde source permet une certaine validation (de

la valeur). Dans ce cas, on parle de renforcement de la confiance liée à la « multiplicité des sources ». La multiplicité des sources se définit comme étant la répétition de saisies identiques issues de méthodes d'acquisitions de données *différentes*. Cela se produit lorsqu'un internaute complète son nom (« Dupont ») dans un formulaire A, et qu'ultérieurement, via un formulaire B, ce même nom soit de nouveau introduit. Face à ce fait, il parait également plausible de majorer la correctness. Nous veillerons à prendre en compte ce paramètre lors de la définition de la formule de correctness. Il prendra également ses valeurs dans l'ensemble des entiers naturels.

**Divergence de valeurs** La multiplicité des sources de données et des occurrences sont deux éléments fondamentaux dans la définition de l'indice de correctness car ils permettent le renforcement de la certitude que l'on a en une valeur. Toutefois, ce n'est pas l'unique raison. En réalité, ces paramètres rendent possible également un « affaiblissement » du crédit que l'on accorde à une valeur. Nous avons considéré jusqu'ici le cas où une même valeur est entrée pour plusieurs sources de données ou occurrences. En pratique, il arrive fréquemment que des valeurs divergentes voire contradictoires soient acquises par le système. Comment gérer cela ? Par exemple, dans un formulaire A, l'internaute indique qu'il se nomme « Dupont », alors que dans un autre (formulaire B), son nom est « Dupond ». Si le système ne peut pas rester dans l'indétermination et donc qu'il soit absolument nécessaire qu'une et une seule valeur devienne effective, laquelle va-t-il choisir ? Dans ce chapitre, nous allons à la fois nous occuper de l'évaluation de la confiance d'une donnée par la combinaison d'un indice de correctness et de freshness, et discuter du choix de la valeur que le système va considérer. Sachant que toute valeur est issue d'une source de donnée, quels sont les critères généraux de choix ? Raisonnons cas par cas.

1. Si nous sommes face à deux sources différentes de fiabilité non égale, alors la valeur choisie sera celle issue de la source de plus haute fiabilité.
2. Si nous sommes face à deux sources différentes de fiabilité égale, alors il est difficile de déterminer laquelle est la plus pertinente à prendre en compte. Néanmoins, il est en général plus intéressant de choisir la valeur dernièrement ajoutée. En effet, étant plus acualisée que l'autre valeur, il y a plus de chance qu'elle soit juste.
3. Si nous sommes face à une seule source possédant deux occurrences de valeurs différentes, alors on considérera chacune des occurrences comme étant une source à part entière de même fiabilité. Dans ce cas cela correspond à la deuxième situation.

**Synthèse** Jusqu'ici nous avons vu qu'une valeur d'un attribut provenait immanquablement d'une source associée à une fiabilité. Cette valeur peut être confirmée ou infirmée au fur et à mesure de l'acquisition des données sur l'utilisateur. La source de données, quant à elle, est un concept englobant à la fois la méthode ou le moyen d'acquérir les données et la fiabilité de l'utilisateur introduisant ces données. Comme discuté dans la partie 2.4.2 (« Alimentation du modèle utilisateur », chapitre 2), la méthode d'acquisition de données peut être un formulaire HTML, un système d'analyse interactionnelle, un résultat de processus, etc. Une source peut posséder plusieurs occurrences, c'est-à-dire, plusieurs acquisitions de données à partir d'elle. Lorsqu'au sein des occurrences d'une source, il y a des divergences de valeurs, la source devra être éclatée en autant de (sous-)sources que de valeurs distinctes. Bien sûr, une valeur d'un attribut peut être complétée par différentes sources de données, possédant, ou non, les mêmes valeurs. Dans le cas ou les valeurs inter-sources seraient hétérogènes, il y a une contradiction entre autant de valeurs. Ce conflit doit être résolu par le choix d'une valeur hypothétiquement effective. Cette valeur sera soit celle issue de la source de plus haute fiabilité lorsque celles-ci ne sont pas égales, ou celle dernièrement acquise en cas de fiabilité identique.

La figure 5.4 reprend un tableau avec les différents cas lorsque deux valeurs sont acquises et que le système doit choisir la valeur qui sera considérée, « active ». Lorsque le signe « $=$ » (« $\neq$ ») est présent dans la colonne valeur, les deux valeurs sont identiques (distinctes). Il peut y avoir une ou plusieurs source(s). Lorsqu'il n'y en a qu'une, cela implique qu'il y a deux occurrences de cette source (car nous avons supposé qu'il y a deux valeurs acquises). A l'inverse, pour les même raisons, quand plusieurs sources sont présentes (donc ici 2), il y a une seule occurrence de chaque source. Lorsque l'on trouve le signe « $=$ » (« $\neq$ ») dans la colonne « Fiabilité », cela signifie que la fiabilité de chaque occurrence de chaque source est égale (différente). Quant à la colonne « Choix », elle indique quelle valeur le système devrait choisir comme active. Remarquons que les lignes 2 et 6 sont grisées. En fait, la combinaison de ces paramètres n'est pas possible, on dit que la ligne est inconsistante. En effet, toutes les occurrences d'une source possèdent par définition la même fiabilité. Or, ces lignes supposent de multiples occurrences en sein d'une source avec des fiabilités différentes, ce qui n'est pas permis.

| | Valeur | Source | Fiabilité | Choix |
|---|---|---|---|---|
| 1 | = | unique | = | Seule valeur disponible |
| 2 | = | unique | ≠ | inconsistant |
| 3 | = | plusieurs | = | Seule valeur disponible |
| 4 | = | plusieurs | ≠ | Seule valeur disponible |
| 5 | ≠ | unique | = | Eclater en deux sources de confiance égale ensuite voir cas 7 |
| 6 | ≠ | unique | ≠ | inconsistant |
| 7 | ≠ | plusieurs | = | Valeur dernièrement acquise |
| 8 | ≠ | plusieurs | ≠ | Valeur issue de la source de meilleur confiance |

FIGURE 5.4: Choix de la valeur active dans les différents cas lorsque deux valeurs sont acquises

Notons que pour arriver à ces conclusions, nous avons dû utiliser un certain nombre d'informations telles que les sources de données, les fiabilités de chaque source, les valeurs issues de chaque source, le nombre d'occurrences d'une source et la date d'ajout de chaque valeur. Pour fonctionner, le module calculant la qualité, et plus particulièrement celui concernant le calcul de la correctness, demandera de fournir en entrée toutes ces données.

Ces données doivent figurer dans le schéma du méta-modèle utilisateur. La figure suivante intègre ces nouvelles informations nécessaires à l'évaluation de l'indice de confiance. Sachant qu'une valeur est impérativement fournie par une source de données, la table « SOURCE » est créée. Elle possède un identifiant, un nom et une fiabilité. L'association « possède » entre les tables « SOURCE » et « VALEUR_ATTRIBUT » est également créée afin de modéliser cette relation et dispose d'un identifiant.

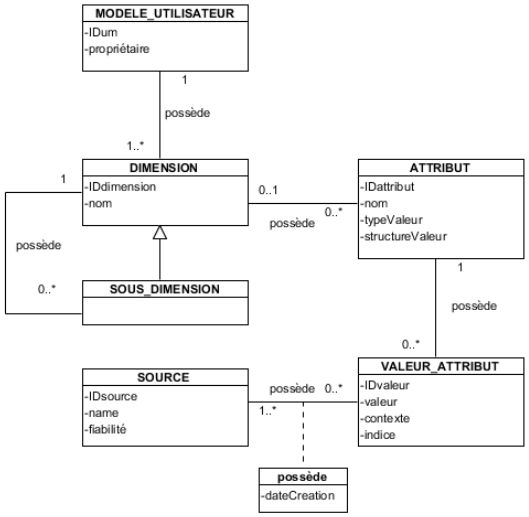



FIGURE 5.5: Méta-modèle utilisateur avec prise en compte des sources de données

#### 5.2.1.2 Effets désirés

Après avoir découvert et posé quelques éléments clés, nous allons continuer notre démarche visant à définir une formule de correctness en étudiant l'évolutivité des valeurs d'un attribut et leur correctness. Pour cela, il est intéressant de se poser la question suivante : qu'en est-il de l'évolution des valeurs actives et de la correctness à mesure que de nouvelles informations viennent alimenter le modèle utilisateur ? Examinons cette interrogation en tentant d'établir une table d'effets désirés d'une formule-type de correctness. Sur base de cela, nous pourrons, dans la section suivante, nous pencher sur cette formule.

Pour examiner les effets désirés de chaque situation possible, nous allons utiliser la technique de « table de décision ». La table a pour objectif de modéliser un ou plusieurs choix en fonction d'un certain nombre de « conditions » (que nous appellerons « paramètres »). Cet outil logique permet une représentation et un raisonnement plus aisé qu'avec une suite de « SI . . . ALORS » imbriqués. Il fait en quelque sorte figure de spécification pour la correctness et la valeur active. La table servira également à effectuer les tests de vérification et validation. La figure 5.6 représente la table des effets désirés de la correctness et de la valeur active. Voici le détail de ses colonnes (paramètres ou conditions) :

- Valeur : indique si les valeurs obtenues par les différentes sources et occurrences de sources sont égales à (« Oui ») ou non (« Non »).
- Source : indique le nombre de sources de données venant alimenter l'attribut considéré. Il peut y

en avoir une, plusieurs ou une infinité (0 étant un cas particulier). Pour faciliter la compréhension nous noterons « 2 » lorsque nous aurons n sources ($1 < n < \infty$). Il est en effet plus simple de raisonner sur deux sources plutôt que trois.

- Occurrence : indique le nombre d'occurrences d'une source de données. Similairement à la colonne « Source », il peut y avoir 1, 2 ou une infinité d'occurrence(s).
- Fiabilité : indique si les valeurs de fiabilité des différentes sources et occurrences sont égales (« $=$ ») ou différentes (« $\neq$ »). Autrement dit, si par exemple une donnée a été fournie par deux sources, est-ce qu'elles possèdent la même valeur de fiabilité ou non ?

Chaque colonne dispose d'un certain nombre de valeurs possibles. La première, 2, la deuxième et la troisième, 3 chacune, et la dernière 2. Nous avons donc $2 \times 3 \times 3 \times 2 = 36$ lignes possibles. Chaque ligne représente un effet désiré ou un test possible. Ainsi, la ligne 1 indique que pour une source, une occurrence, la correctness doit être égale à la fiabilité et la valeur active sera la seule disponible. Dans ce cas, étant donné qu'il n'y a qu'une source et qu'une occurrence, les valeurs et les fiabilités sont toutes égales. Remarquons que parmi cet ensemble de lignes, certaines n'ont pas de sens, elles sont « inconsistantes ». Ce sont les lignes grisées dans la figure 5.6. Prenons la ligne 2, avec une source et une occurrence, il est impossible d'avoir plusieurs valeurs de fiabilité étant donné qu'il n'y a qu'une valeur provenant d'une seule source et qu'une source ne peut posséder qu'une unique valeur de fiabilité. Cette ligne peut donc être éliminée. Pour la même raison, les lignes 4 et 6 peuvent également être supprimées. Notons que nous supposons que pour une source de données, chacune de ses occurrences possède la même fiabilité. Si ce n'est pas le cas, cette source doit être éclatée en plusieurs sous-sources. Aussi, lorsque nous avons une seule source, « les » valeurs sont forcément égales étant donné qu'il n'y en a qu'une seule ! L'inverse est erroné, c'est pourquoi les lignes de 19 à 24 doivent être retirées. Aux 36 lignes de départ, 9 inconsistantes sont éliminées. Il en reste 27, auxquelles nous ajoutons trois cas supplémentaires (0, 3.1 et 30.1) afin d'être encore plus complet. Le tableau comportera donc 30 lignes.

| | Valeur | Source | Occurrence | Fiabilité | Correctness | Valeur choisie |
|---|---|---|---|---|---|---|
| 0 | oui | 0 | 0 | - | 0 | - |
| 1 | oui | 1 | 1 | = | Fiab | Seule valeur disponible |
| 2 | oui | 1 | 1 | ≠ | inconsistent | - |
| 3 | oui | 1 | 2 | = | 1>Corr>Fiab | Seule valeur disponible |
| 3.1 | oui | 1 | n>2 | = | Corr>Corr(3) | Seule valeur disponible |
| 4 | oui | 1 | 2 | ≠ | inconsistent | - |
| 5 | oui | 1 | ∞ | = | 1 | Seule valeur disponible |
| 6 | oui | 1 | ∞ | ≠ | inconsistent | - |
| 7 | oui | 2 | 1 | = | 1>Corr>Fiab | Seule valeur disponible |
| 8 | oui | 2 | 1 | ≠ | 1>Corr>((Fiab1+Fiab2)/2) | Seule valeur disponible |
| 9 | oui | 2 | 2 | = | 1>Corr>Corr(3) | Seule valeur disponible |
| 10 | oui | 2 | 2 | ≠ | 1>Corr>Fiab1 | Seule valeur disponible |
| 11 | oui | 2 | ∞ | = | 1 | Seule valeur disponible |
| 12 | oui | 2 | ∞ | ≠ | 1 | Seule valeur disponible |
| 13 | oui | ∞ | 1 | = | 1 | Seule valeur disponible |
| 14 | oui | ∞ | 1 | ≠ | 1 | Seule valeur disponible |
| 15 | oui | ∞ | 2 | = | 1 | Seule valeur disponible |
| 16 | oui | ∞ | 2 | ≠ | 1 | Seule valeur disponible |
| 17 | oui | ∞ | ∞ | = | 1 | Seule valeur disponible |
| 18 | oui | ∞ | ∞ | ≠ | 1 | Seule valeur disponible |
| 19 | non | 1 | 1 | = | inconsistent | - |
| 20 | non | 1 | 1 | ≠ | inconsistent | - |
| 21 | non | 1 | 2 | = | inconsistent | - |
| 22 | non | 1 | 2 | ≠ | inconsistent | - |
| 23 | non | 1 | ∞ | = | inconsistent | - |
| 24 | non | 1 | ∞ | ≠ | inconsistent | - |
| 25 | non | 2 | 1 | = | Fiab/N Sour. | Valeur la plus récente |
| 26 | non | 2 | 1 | ≠ | Corr -> Corr1 & Corr>Corr(25) | Valeur issue de la source de meilleure confiance |
| 27 | non | 2 | 2 | = | Corr>Corr(25) | Valeur la plus récente |
| 28 | non | 2 | 2 | ≠ | Corr -> Corr1 & Corr>Corr(27) | Valeur issue de la source de meilleure confiance |
| 29 | non | 2 | ∞ | = | 1/ N source | Valeur la plus récente |
| 30 | non | 2 | ∞ | ≠ | 1/ N source | Valeur issue de la source de meilleure confiance |
| 30.1 | non | 2 | 1:∞ ; 2:1 | ≠ | Corr=Corr1 & Corr=1 | Valeur issue de la source 1 |
| 31 | non | ∞ | 1 | = | 0 | Valeur la plus récente |
| 32 | non | ∞ | 1 | ≠ | 0 | Valeur issue de la source de meilleure confiance |
| 33 | non | ∞ | 2 | = | 0 | Valeur la plus récente |
| 34 | non | ∞ | 2 | ≠ | 0 | Valeur issue de la source de meilleure confiance |
| 35 | non | ∞ | ∞ | = | 0 | Valeur la plus récente |
| 36 | non | ∞ | ∞ | ≠ | 0 | Valeur la plus récente |

FIGURE 5.6: Table des effets désirés de la correctness

Voici une brève explication de chaque ligne :

- 0 : Quand il n'y a aucune source, il n'y a pas de valeur à choisir ni de correctness.
- 1 : La correctness vaut la fiabilité de la seule source. La valeur choisie est celle de cette source.
- 3 : Lorsqu'il y a deux occurrences pour une source, la correctness doit être plus élevée que la fiabilité de la source étant donné que la source a fourni à deux reprises la même valeur, impliquant un renforcement de la croyance en l'exactitude de la donnée. La valeur choisie est celle de cette source.
- 3.1 : Pour les mêmes raisons qu'en 3, la correctness doit être plus élevée qu'en 3. La valeur choisie est celle de cette source.
- 5 : Le fait qu'une infinité d'occurrences indique la même valeur, nous incite à croire avec certitude que la donnée est exacte. C'est le même raisonnement qu'en 3 et 3.1. En effet, plus il y a d'occurrences d'une source, plus la correctness est élevée. Quand le nombre d'occurrences tend vers l'infini, la correctness tend vers sa limite supérieure, c'est-à-dire 1. La valeur choisie est celle de cette source.

- 7 : Lorsqu'il y a deux sources de données avec une même fiabilité, la correctness doit être plus élevée que cette valeur de fiabilité étant donné qu'à deux reprises, via deux sources différentes, la même valeur a été fournie, impliquant un renforcement de la croyance en l'exactitude de la donnée. La valeur choisie est celle de cette source. Etant donné que les sources ont fourni une valeur identique, il n'y a qu'un choix possible de valeur.
- 8 : Lorsqu'il y a deux sources de données de fiabilité différente, il serait logique de penser que la correctness est supérieure ou égale à la moyenne des fiabilités des sources. Pour s'en convaincre, imaginons l'inverse, la correctness est inférieure à la moyenne des fiabilités des sources. Cela ne suit pas la logique de « renforcement » expliquée jusqu'ici, qui veut que la correctness augmente (ou en tout cas ne diminue pas) lorsque différentes occurrences ou sources de données de valeurs égales sont présentes. Cette hypothèse doit donc être rejetée. La valeur active est la seule possible.
- 9 : Toujours en suivant cette logique de renforcement, la correctness doit être supérieure à celle de la ligne 3. La valeur active est la seule possible.
- 10 : La correctness doit être supérieure à celle de la ligne 8. La valeur active est la seule possible.
- De 11 à 18 : En appliquant la logique de renforcement à l'extrême (comme à la ligne 5), ces lignes prennent toutes la valeur 1. La valeur active est la seule possible.
- 25 : Lorsque nous avons deux sources possédant une même valeur de fiabilité, une occurrence chacune, mais fournissant des valeurs divergentes, la correctness doit valoir la fiabilité par le nombre de sources. Dans ce cas nous sommes face à un « affaiblissement » de la croyance que l'on a en une valeur par l'introduction d'une nouvelle valeur contradictoire. La valeur qui sera choisie sera celle dernièrement acquise étant donné qu'on ne peut pas faire plus confiance à une valeur qu'à une autre.
- 26 : La justification de cette ligne n'est pas triviale. Par exemple, si nous sommes en possession de deux valeurs, une issue d'une source de fiabilité « 0,9 » et une autre issue d'une source de fiabilité « 0,2 ». Il est logique de penser que si une des deux valeurs est la bonne, c'est certainement la première. Dès lors, la correctness devrait être plus proche de la fiabilité de la première que de la seconde. Plus proche, mais inférieure par le principe d'affaiblissement. C'est maintenant que les choses se compliquent. Admettons que les fiabilités des sources de la ligne 25 soit aussi de « 0,9 ». Dans ce cas la correctness de cette ligne doit être supérieure à celle de la ligne 25. En effet, par le principe d'affaiblissement, la deuxième source étant de moindre fiabilité, le « poids » que pèse

cette valeur contradictoire est moindre, et donc, la correctness ne devrait pas être aussi basse qu'en 25.

- 27 : Cette ligne se justifie de la même manière que la ligne 25 (affaiblissement). Cependant, la correctness devrait être légèrement plus élevée. En effet, à ce phénomène d'affaiblissement se joint celui d'enrichissement (car il y a deux occurrences pour chaque source). La valeur la plus récente sera la valeur choisie.
- 28 : S'obtient aisément avec les justifications des lignes 26 et 27. La valeur active sera celle issue de la source de meilleure fiabilité.
- 29 : S'obtient aisément avec les justifications des lignes 5 et 25. La valeur active sera la plus récente.
- 30 : S'obtient aisément avec les justifications des lignes 5 et 25. La valeur active sera celle issue de la source de meilleure fiabilité.
- 30.1 : Cette ligne correspond au cas où la première source possède une infinité d'occurrence alors que la seconde n'en possède qu'une seule. La correctness vaudra alors « 1 » et la valeur choisie sera celle issue de la première source.
- 31, 33, 35 et 36 : Lorsqu'il y a une infinité de sources avec une infinité de valeurs, peu importe le nombre d'occurrences, la correctness vaut 0. Ceci se démontre facilement. En effet, il y a une infinité de valeurs contradictoires, ce qui correspond au cas 25 avec un nombre de sources tendant vers l'infini. La limite d'une fraction où le dénominateur tend (seul) vers l'infini vaut « 0 ». La valeur choisie sera la plus récente.
- 32 et 34 : la correctness est égale à « 0 » (cf. justification précédente). La valeur choisie sera celle de meilleure fiabilité (ou éventuellement la plus récente si plusieurs valeurs issues de sources de même fiabilité).

#### 5.2.1.3 Formules

Après avoir découvert les éléments clés et étudié les effets désirés, nous allons aborder la définition de la formule de correctness. Nous allons procéder par cycle. Un cycle comportera deux étapes : (1) la recherche d'une formule, (2) la validation de la formule sur base de tests tirés de la table d'effets désirés. Si la seconde étape n'est pas satisfaite, nous revenons à la première, et ainsi de suite jusqu'à l'obtention d'une formule satisfaisante.

**Recherche d'une formule :** Avec les données dont nous disposons, la manière la plus triviale de définir la correctness serait la suivante (figure 5.7). D'abord, les sources de données (représenté par l'ensemble $S$) sont placées dans un tableau. Chaque ligne correspond à une source. Le tableau compte $n$ ligne(s) donc $n$ source(s). Chaque source $S_i$ ($\forall S_i \in S$ où $i$ est l'indice de la ligne du tableau et $0 \leq i \leq n$) possède un nom, une fiabilité ($FS_i$), un nombre d'occurrences ($NOS_i$), une date de dernière mise à jour et une valeur ($VS_i$). Sur base de ces informations, le poids de chaque source est calculé en multipliant la fiabilité par le nombre d'occurrences ($PoidsS_i = FS_i * NOS_i$), et la part du poids de chaque ligne sur le poids total est déterminé en divisant le poids de la ligne par le poids total de toutes les lignes ($PartPoidsS_i = \frac{PoidsS_i}{PoidsT}$ $où$ $PoidsT = \sum_i^n PoidsS_i$). La correctness 1 se calcule comme suit : $Correctness1 = FS * PartPoidsS$.

| Source | FS | NOS | Last Date | VS | PoidsS | PartPoidsS | Correctness1 |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont | 0,9 | 0,5 | **0,45** |
| Formulaire 2 | 0,9 | 1 | 15-oct | Dupond | 0,9 | 0,5 | **0,45** |
| | | 2 | | | 1,8 | 1 | 0,9 |

FIGURE 5.7: Détail du calcul de correctness 1 avec fiabilités égales

**Validation :** Un des problèmes de cette manière de procéder est la non prise en compte de l'effet de renforcement. Pour preuve, la figure 5.8 considère une source de données avec une seule occurrence. Nous obtenons « 0,9 » de correctness. Supposons maintenant que cette source ait été utilisée dix fois, en fournissant la même valeur (figure 5.9). La correctness est toujours de « 0,9 ». Pourtant, nous aurions aimé qu'il y ait, au fur et à mesure que le nombre d'occurrences s'élève, un renforcement de la croyance en l'exactitude de la valeur. Donc, la valeur de la correctness devrait être plus importante. Ceci nous oblige à devoir repenser, adapter la formule.

| Source | FS | NOS | Last Date | VS | PoidsS | PartPoidsS | Correctness1 |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont | 0,9 | 1 | **0,9** |
| Formulaire 2 | 0,9 | 0 | 15-oct | Dupond | 0 | 0 | **0** |
| | | 1 | | | 0,9 | 1 | 0,9 |

FIGURE 5.8: Détail du calcul de correctness 1, une seul source de données

| Source | FS | NOS | Last Date | VS | PoidsS | PartPoidsS | Correctness1 |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 10 | 10-oct | Dupont | 9 | 1 | **0,9** |
| Formulaire 2 | 0,9 | 0 | 15-oct | Dupond | 0 | 0 | **0** |
| | | 10 | | | 9 | 1 | 0,9 |

FIGURE 5.9: Détail du calcul de correctness 1, une seul source de données, dix occurences

**Recherche d'une formule :** Pour palier au manquement de la formule précédente, nous devons intégrer un moyen de quantifier le renforcement. Pour cela nous allons utiliser la fonction « racine nième » que nous appellerons « Renf » (pour « renforcement ») :

$$Renf(x, y) = \sqrt[x]{y}$$

Elle a l'avantage d'augmenter, de moins en moins fort, la valeur de y en fonction de la croissance de x. Bien sûr, x correspond au nombre d'occurrences et y à la fiabilité. La fonction devient :

$$Renf(NombreOccurences, Fiabilité) = \sqrt[NombreOccurences]{Fiabilité}$$

La figure suivante décrit l'évolution de la fonction au départ d'une fiabilité de « 0,5 ». Nous pouvons voir que le résultat de cette fonction tend vers « 1 » pour un nombre d'occurrences élevé.

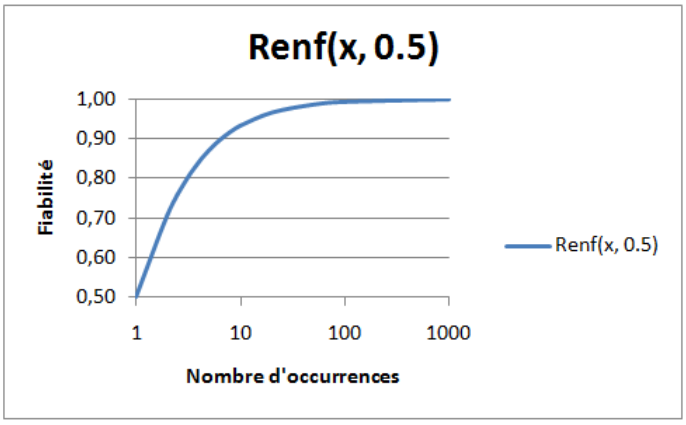



FIGURE 5.10: Graphe de la fonction de renforcement pour une fiabilité de "0,5"

Ci-dessous, le tableau de l'évolution des résultats de cette fonction pour différentes fiabilités.

Fonction: $Renf\,(NombreOccurences, Fiabilité) = \sqrt[NombreOccurences]{Fiabilité}$

| Fiabilité / NombreOccurence | 1 | 0,9 | 0,8 | 0,7 | 0,6 | 0,5 | 0,4 | 0,3 | 0,2 | 0,1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0,9 | 0,8 | 0,7 | 0,6 | 0,5 | 0,4 | 0,3 | 0,2 | 0,1 | 0 |
| 2 | 1 | 0,95 | 0,89 | 0,84 | 0,77 | 0,71 | 0,63 | 0,55 | 0,45 | 0,32 | 0 |
| 3 | 1 | 0,97 | 0,93 | 0,89 | 0,84 | 0,79 | 0,74 | 0,67 | 0,58 | 0,46 | 0 |
| 4 | 1 | 0,97 | 0,95 | 0,91 | 0,88 | 0,84 | 0,80 | 0,74 | 0,67 | 0,56 | 0 |
| 5 | 1 | 0,98 | 0,96 | 0,93 | 0,90 | 0,87 | 0,83 | 0,79 | 0,72 | 0,63 | 0 |
| 6 | 1 | 0,98 | 0,96 | 0,94 | 0,92 | 0,89 | 0,86 | 0,82 | 0,76 | 0,68 | 0 |
| 7 | 1 | 0,99 | 0,97 | 0,95 | 0,93 | 0,91 | 0,88 | 0,84 | 0,79 | 0,72 | 0 |
| 8 | 1 | 0,99 | 0,97 | 0,96 | 0,94 | 0,92 | 0,89 | 0,86 | 0,82 | 0,75 | 0 |
| 9 | 1 | 0,99 | 0,98 | 0,96 | 0,94 | 0,93 | 0,90 | 0,87 | 0,84 | 0,77 | 0 |
| 10 | 1 | 0,99 | 0,98 | 0,96 | 0,95 | 0,93 | 0,91 | 0,89 | 0,85 | 0,79 | 0 |
| 20 | 1 | 0,99 | 0,99 | 0,98 | 0,97 | 0,97 | 0,96 | 0,94 | 0,92 | 0,89 | 0 |
| 50 | 1 | 1,00 | 1,00 | 0,99 | 0,99 | 0,99 | 0,98 | 0,98 | 0,97 | 0,95 | 0 |
| 100 | 1 | 1,00 | 1,00 | 1,00 | 0,99 | 0,99 | 0,99 | 0,99 | 0,98 | 0,98 | 0 |
| 1000 | 1 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0 |

FIGURE 5.11: Tableau présentant l'évolution des valeurs de la fonction de renforcement pour différentes fiabilités

La figure suivante détaille le calcul de la correctness 2 prenant en compte le calcul du renforcement. Pour chaque ligne $i$, la part (nombre d'occurrences d'une ligne divisé par la somme du nombre d'occurrences de chaque ligne : $PartNOS_i = \frac{NOS_i}{NOST}$ $où$ $NOST = \sum_i^n NOS_i$) et le renforcement sont calculés. La correctness résulte simplement de la multiplication de ses deux colonnes : $Correctness2 = PartNOS * Renf$.

| Source | FS | NOS | Last Date | VS | PartNOS | Renf | Correctness2 |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont | 0,50 | 0,90 | **0,45** |
| Formulaire 2 | 0,9 | 1 | 15-oct | Dupond | 0,50 | 0,90 | **0,45** |
| | | 2 | | | 1 | 1,80 | 0,90 |

FIGURE 5.12: Détail du calcul de correctness 2, deux sources de données de fiabilités identiques, deux valeurs distinctes

Nous pouvons constater que le renforcement est effectivement pris en considération (figure 5.13).

| Source | FS | NOS | Last Date | VS | PartNOS | Renf | Correctness2 |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 2 | 10-oct | Dupont | 1,00 | 0,95 | **0,95** |
| Formulaire 2 | 0,9 | 0 | 15-oct | Dupond | 0,00 | 0,00 | **0,00** |
| | | 2 | | | 1 | 0,95 | 0,95 |

FIGURE 5.13: Détail du calcul de correctness 2, une source de donnée, deux occurences

**Validation :** A première vue, cette formule semble satisfaisante. Néanmoins, elle présente une importante lacune. En fait, le renforcement n'est que partiellement respecté. Comme illustré à la figure suivante, lorsque deux sources différentes fournissent la même valeur, le renforcement n'est pas pris en compte car le calcul de la fonction de renforcement se base sur le nombre d'occurrences. Or, ici il n'y a qu'une seule occurrence pour chaque ligne. Pire, dans ce cas, cette manière de faire réagit comme un affaiblissement (car la correctness est inférieure quand deux sources d'une certaine fiabilité fournissent la même valeur que lorsqu'une seule source fournit une valeur avec la même fiabilité) ! Il est donc indispensable de changer le procédé.

| Source | FS | NOS | Last Date | VS | PartNOS | Renf | Correctness2 |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont | 0,50 | 0,90 | **0,45** |
| Formulaire 2 | 0,9 | 1 | 15-oct | Dupont | 0,50 | 0,90 | **0,45** |
| | | 2 | | | 1 | 1,80 | 0,90 |

FIGURE 5.14: Détail du calcul de correctness 2, deux sources de données de fiabilités identiques, une même valeur

**Recherche d'une formule :** Pour pouvoir tenir compte à la fois du renforcement intra-source (multi-occurrences au sein d'une même source) et inter-sources, il est nécessaire d'adapter la représentation. Pour cela, nous allons utiliser, en plus, un tableau intermédiaire non plus orienté « multi-sources » (comme les précédents) mais multi-valeurs (figure 5.15). L'ensemble des valeurs se notera $V$. Chaque ligne de ce nouveau tableau représente une valeur. Le tableau compte $m$ ligne(s) donc $m$ valeur(s). Chaque valeur $V_j$ ($\forall V_j \in V$où $j$ est l'indice de la ligne du tableau orienté valeurs et $0 \leq j \leq m$) possède un nom de valeur ($VN_j$), un nombre d'occurrences ($NOV_j$), la fiabilité maximum de cette valeur dans le premier tableau ($MaxFiabV_j = Max_j\{FS_i | i \in S \wedge VS_i = VN_j\}$), la part que représente cette valeur par rapport aux autres ($PartNOV_j = \frac{NOV_j}{NOVT}$ $où$ $NOVT = \sum_j^m NOV_j$ [1] ), le renforcement et la correctness 3 ($Correctness3 = PartNOV * Renf(NOV, MaxFiabV)$).

1. Remarquons que $NOVT = NOST où\ NOST = \sum_i^n NOS_i$

| Source | FS | NOS | Last Date | VS |
|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont |
| Formulaire 2 | 0,9 | 1 | 15-oct | Dupond |
| Formulaire 3 | 0,9 | 0 | 20-oct | Dupont |
| | | 2 | | |

**Aspect multivaleurs:**

| VN | NOV | MaxFiabV | PartNOV | Renf | Correctness3 |
|---|---|---|---|---|---|
| Dupont | 1 | 0,9 | 0,50 | 0,90 | **0,45** |
| Dupond | 1 | 0,9 | 0,50 | 0,90 | **0,45** |
| | 2 | | 1,00 | 1,80 | 0,90 |

FIGURE 5.15: Détail du calcul de correctness 3, deux sources de données de fiabilités identiques, deux valeurs distinctes

Vérifions que cette fois, le renforcement est totalement pris en compte. La figure suivante montre que c'est en effet le cas. Les sources « Formulaire 1 » et « Formulaire 3 » ont fourni la même valeur. Cela se voit dans le second tableau où la valeur « Dupont » possède deux occurrences et le calcul de renforcement se fait sur base de cette agrégation au niveau valeur.

| Source | FS | NOS | Last Date | VS |
|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont |
| Formulaire 2 | 0,9 | 0 | 15-oct | Dupond |
| Formulaire 3 | 0,9 | 1 | 20-oct | Dupont |
| | | 2 | | |

**Aspect multivaleurs:**

| VN | NOV | MaxFiabV | PartNOV | Renf | Correctness3 |
|---|---|---|---|---|---|
| Dupont | 2 | 0,9 | 1,00 | 0,95 | **0,95** |
| Dupond | 0 | 0,9 | 0,00 | 0,00 | **0,00** |
| | 2 | | 1,00 | 0,95 | 0,95 |

FIGURE 5.16: Détail du calcul de correctness 3, deux sources de données de fiabilités identiques, une même valeur

**Validation :** Le renforcement est pris en compte mais n'est pas toujours très réaliste. Prenons le cas de la figure 5.17. Deux sources (« Formulaire 1 » et « Formulaire 2 ») possédant chacune une occurrence fournissent la même valeur. Le tableau orienté multi-valeurs va agréger celles-ci et indiquer qu'il y a deux occurrences de cette valeur. L'ennui se situe au niveau du calcul du renforcement. Nous avons vu que le renforcement prenait deux paramètres : le nombre d'occurrences et une fiabilité. C'est cette dernière entrée qui est la cause de l'appréciation non réaliste du renforcement. De fait, la fiabilité utilisée est celle de valeur maximale. Ceci implique que pour calculer le renforcement, les deux sources considérées ont toutes les deux une fiabilité de « 0,9 ». Or, il n'en est rien car la source « Formulaire 3 » possède une fiabilité de « 0,1 ». Nous sommes donc face à un problème de surévaluation

du renforcement. En fait, la valeur de renforcement ne devrait pas valoir « 0,95 » mais « 0,91 ». Une nouvelle fois, le procédé menant à un indice de correctness ne convient pas.

| Source | FS | NOS | Last Date | VS |
|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont |
| Formulaire 2 | 0,9 | 0 | 15-oct | Dupond |
| Formulaire 3 | 0,1 | 1 | 20-oct | Dupont |
| | | 2 | | |

**Aspect multivaleurs:**

| VN | NOV | MaxFiabV | PartNOV | Renf | Correctness3 |
|---|---|---|---|---|---|
| Dupont | 2 | 0,9 | 1,00 | 0,95 | **0,95** |
| Dupond | 0 | 0,9 | 0,00 | 0,00 | **0,00** |
| | 2 | | 1,00 | 0,95 | 0,95 |

FIGURE 5.17: Détail du calcul de correctness 3, deux sources de données de fiabilités différentes, une même valeur

**Recherche d'une formule :** Pour pallier au problème précédent, des informations intermédiaires supplémentaires ainsi qu'une nouvelle sous-fonction devront être intégrées. Reprenons l'exemple précédent (figure 5.17). Nous voulons obtenir une valeur $x$ résultante de la fonction de renforcement telle que :

$$0,95 > x > 0,90$$

$$\text{Donc : } \sqrt[2]{0,9} > \sqrt[t]{0,9} > \sqrt[1]{0,9}$$

Nous désirons obtenir $t$ telle que $1 < t < 2$. La borne « 2 » correspond au cas où les deux sources possèdent la même fiabilité. La borne « 1 » correspond au cas où la deuxième source a une fiabilité nulle. Nous allons donc utiliser une fonction dite de « correspondance » (« matching » en anglais) afin de faire correspondre une fiabilité $\alpha$ avec une valeur dans l'intervalle [0,1]. La fiabilité $\alpha_i$ $(FS_i)$ est comprise entre 0 et $\beta_i$. La fiabilité $\beta_i$ est celle de valeur maximale :

$\beta_i = MaxFiabS_i = Max\{FS_k | k \in S \wedge VS_k = VS_i\}$ où $i$ est l'indice de la ligne du tableau orienté sources pour lequel on désire calculer la fiabilité maximale. En d'autres termes, pour une ligne $i$ du tableau orienté source, $\beta_i$ représente la fiabilité maximale parmi les fiabilités de toutes les lignes de même $VS$ de ce tableau .

Plus $\alpha$ sera proche de $\beta$, plus le résultat de la fonction de correspondance sera proche de 1.

Inversement, plus $\alpha$ sera éloigné de $\beta$, plus le résultat tendra vers 0. Cette fonction se définit comme suit :

$$Match(\alpha, \beta) = \frac{\alpha}{\beta}$$

Dans l'exemple, la fiabilité la plus élevée est « 0,9 » (« Formulaire 1 »). Donc $\beta = 0,9$. La figure suivante montre l'évolution des valeurs de la fonction de correspondance pour différentes valeurs de $\alpha$ et pour$\beta = 0,9$.

$f(\alpha) = Match\,(\alpha, 0.9) = \frac{\alpha}{0.9}$

| α | f(α) |
|---|---|
| 0 | 0 |
| 0,1 | 0,11 |
| 0,2 | 0,22 |
| 0,3 | 0,33 |
| 0,4 | 0,44 |
| 0,5 | 0,56 |
| 0,6 | 0,67 |
| 0,7 | 0,78 |
| 0,8 | 0,89 |
| 0,9 | 1 |

FIGURE 5.18: Exemple de l'évolution des valeurs pour la fonction match de fiabilité "0,9"

Dans l'exemple de la figure 5.17, la fiabilité de la source « Formulaire 3 » valait « 0,1 ». La valeur résultante de la fonction de correspondance est par conséquent « 0,11 ».

Nous savons que $1 < t < 2$. Nous avons :

$$t = 1 + Match(\alpha, \beta)$$

C'est en fait la somme

$$Match(\beta, \beta) + Match(\alpha, \beta)$$

Où $Match(\beta, \beta)$ concerne la valeur de fiabilité maximale parmi toutes les sources de même valeur (donc de la source « Formulaire 1 »). Elle vaut « 1 » car $Match(0.9, 0.9) = 1$ (cf figure 5.18)

Nous obtenons :

$$t = 1 + 0,11 = 1,11$$

Nous avons donc :

$$\sqrt[2]{0,9} > \sqrt[1,11]{0,9} > \sqrt[1]{0,9}$$

Nous voulions que $x$ soit tel que :

$$0,95 > x > 0,90$$

C'est en effet le cas :

$$x = \sqrt[1,11]{0,9} = 0,91$$

Cette valeur est plus réaliste que la précédente (0,95).

Nous allons compléter le tableau afin d'intégrer la fonction de correspondance. La fonction « match » réclame deux paramètres : la fiabilité de la source considérée et la fiabilité maximale pour toutes les sources qui ont fournit la même valeur. Le premier paramètre est déjà présent dans la deuxième colonne du tableau orienté sources de données ($FS$). Le second ne l'est pas encore, il faut donc l'ajouter au premier tableau (figure 5.19). Nous appellerons cette colonne « MaxFiabS ». Le résultat de la fonction de correspondance se trouvera dès lors dans la colonne qui suit, nommée « Match ». Etant donné qu'une source peut posséder plus qu'une occurrence, il serait erroné de ne pas les prendre en compte. C'est pourquoi, la dernière colonne de ce premier tableau reprend le résultat de la fonction « match » multiplié par le nombre d'occurrences de cette source de données ($MatchNOS = Match * NOS$). Qu'en est-il du second tableau ? D'abord, le tableau orienté valeurs s'est vu ajouter une colonne supplémentaire : « SumMatchNOS ». Cette colonne contient, pour chaque ligne, la somme de la valeur de la colonne « MatchNOS » (du tableau orienté sources) pour chaque source de données fournissant la valeur correspondante à la valeur de la ligne considérée (du tableau orienté valeurs) : $SumMatchNOS_j = \sum_{i \in S \wedge VN_j = VS_i} MatchNOS_i$ où $j$ est l'indice de la ligne du tableau orienté valeur dont on veut trouver la valeur "SumMatchNOS" et $i$ représente l'indice des lignes du tableau orienté sources. Ensuite, le renforcement n'est plus calculé sur base de « NOV » mais sur « SumMatchNOS » : $Renf_j(SumMatchNOS_j, MaxFiabV_j)$. Enfin, comme précédemment, la correctness est le résultat de la multiplication de la colonne « PartNOV » par la colonne « Renf » : $Correctness4 = PartNOV * Renf$.

| Source | FS | NOS | Last Date | VS | MaxFiabS | Match | MatchNOS |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 2 | 10-oct | Dupont | 0,9 | 1,00 | 2,00 |
| Formulaire 2 | 0,9 | 1 | 15-oct | Dupond | 0,9 | 1,00 | 1,00 |
| Formulaire 3 | 0,9 | 1 | 20-oct | Dupont | 0,9 | 1,00 | 1,00 |
| | | 4 | | | | | |

**Aspect Multivaleurs**

| NV | NOV | MaxFiabV | PartNOV | SumMatchNOS | Renf | Correctness4 |
|---|---|---|---|---|---|---|
| Dupont | 3 | 0,90 | 0,75 | 3,00 | 0,97 | **0,72** |
| Dupond | 1 | 0,90 | 0,25 | 1,00 | 0,90 | **0,23** |
| | 4 | | 1 | 4,00 | 1,87 | 0,95 |

FIGURE 5.19: Détail du calcul de correctness 4, trois sources de données de fiabilités identiques, deux valeurs distinctes

Nous pouvons constater dans la figure 5.20 que le problème discuté précédemment concernant l'aspect peu réaliste du renforcement est en effet résolu.

| Source | FS | NOS | Last Date | VS | MaxFiabS | Match | MatchNOS |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont | 0,9 | 1,00 | 1,00 |
| Formulaire 2 | 0,9 | 0 | 15-oct | Dupond | 0,9 | 0,00 | 0,00 |
| Formulaire 3 | 0,1 | 1 | 20-oct | Dupont | 0,9 | 0,11 | 0,11 |
| | | 2 | | | | | |

**Aspect Multivaleurs**

| NV | NOV | MaxFiabV | PartNOV | SumMatchNOS | Renf | Correctness4 |
|---|---|---|---|---|---|---|
| Dupont | 2 | 0,90 | 1,00 | 1,11 | 0,91 | **0,91** |
| Dupond | 0 | 0,90 | 0,00 | 0,00 | 0,00 | **0,00** |
| | 2 | | 1 | 1,11 | 0,91 | 0,91 |

FIGURE 5.20: Détail du calcul de correctness 4, deux sources de données de fiabilités différentes, une même valeur

**Validation :** Appliquons la grille d'effets désirés définie précédemment afin de déterminer si cette manière d'évaluer la correctness peut convenir.

Comme le montre la figure 5.24, la correctness 4 réussit tous les tests à l'exception du 26 et du 28. Lorsque l'on a deux sources en contradiction avec une fiabilité différente, la correctness vaut, quoiqu'il arrive, 50% du renforcement (car la part vaut « 0,50 »). C'est ce qu'illustre la figure 5.21. En effet, si le formulaire 1 possède une fiabilité de « 0,9 » et le formulaire 2 une fiabilité de « 0,001 », nous sommes presque certains que le formulaire 1 dit vrai. Or la valeur « 0,45 » de correctness ne quantifie pas bien le fait de "presque certitude".

| Source | FS | NOS | Last Date | VS | MaxFiabS | Match | MatchNOS |
|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont | 0,9 | 1,00 | 1,00 |
| Formulaire 2 | 0,001 | 1 | 15-oct | Dupond | 0,001 | 1,00 | 1,00 |
| | | 2 | | | | | |

**Aspect Multivaleurs**

| NV | NOV | MaxFiabV | PartNOV | SumMatchNOS | Renf | Correctness4 |
|---|---|---|---|---|---|---|
| Dupont | 1 | 0,90 | 0,50 | 1,00 | 0,90 | **0,45** |
| Dupond | 1 | 0,001 | 0,50 | 1,00 | 0,00 | **0,00** |
| | 2 | | 1 | 2,00 | 0,90 | 0,45 |

FIGURE 5.21: Détail du calcul de correctness 4, deux sources de données de fiabilités différentes, deux valeurs distinctes

**Recherche d'une formule :** Pour solutionner ce problème, nous allons adopter une pondération, non plus avec la "Part" (colonne "PartNOV") par rapport au nombre total d'occurrences, mais en introduisant le concept de « Poids ». Comme nous l'avons introduit pour la correctness 1, le poids se calcule en multipliant la fiabilité d'une source par son nombre d'occurrences. Il est donc nécessaire d'ajouter une colonne « PoidsS » au tableau orienté sources ainsi qu'une colonne « SumPoids » au tableau orienté valeurs qui contiendra pour une ligne, la somme des poids (dans le tableau orienté sources) correspondant à cette valeur ($SumPoids_j = \sum_{i \in S \wedge VN_j = VS_i} PoidsS_i$ ) où $j$ est l'indice de la ligne du tableau orienté valeur dont on veut trouver la valeur "SumPoids" et $i$ représente l'indice des lignes du tableau orienté sources. Ensuite, on calculera, pour chaque ligne (du tableau orienté valeurs), la part totale du poids ($PartPoidsV_j = \frac{SumPoids_j}{SumPoidsT}$ où $SumPoidsT = \sum_j SumPoids_j$), qui servira à multiplier le résultat du renforcement afin d'obtenir la correctness ($Correctness5 = PartPoidsV * Renf$). La figure suivante montre ces adaptations.

| Source | FS | NOS | Last Date | VS | MaxFiabS | Match | MatchNOS | PoidsS |
|---|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 10-oct | Dupont | 0,9 | 1,00 | 1,00 | 0,9 |
| Formulaire 2 | 0,1 | 1 | 15-oct | Dupond | 0,1 | 1,00 | 1,00 | 0,1 |
| Formulaire 3 | 0,1 | 0 | 20-oct | Dupond | 0,1 | 0,00 | 0,00 | 0 |
| | | 2 | | | | | | 1 |

**Aspect Multivaleurs**

| NV | NOV | MaxFiabV | SumPoids | PartPoidsV | SumMatchNOS | Renf | Correctness5 |
|---|---|---|---|---|---|---|---|
| Dupont | 1 | 0,9 | 0,9 | 0,90 | 1,00 | 0,90 | **0,81** |
| Dupond | 1 | 0,1 | 0,1 | 0,10 | 1,00 | 0,10 | **0,01** |
| | 2 | | 1 | 1,00 | 2,00 | 1,00 | 0,82 |

FIGURE 5.22: Détail du calcul de correctness 5, deux sources de données de fiabilités différentes, deux valeurs distinctes

**Validation :** La figure 5.24 montre que la correctness 5 réussi tous les tests à l'exception du trentième. Le test numéro 30 concerne le cas où deux sources, avec chacune une infinité d'occurrences, fournissent des valeurs contradictoires et n'ont pas la même fiabilité. La valeur attendue est « 0,50 » pour chaque source. Pourtant, comme l'illustre la figure suivante, cela n'est pas le cas. Ceci est dû au fait que la correctness est pondérée non plus avec la part relative aux nombres d'occurrences mais avec la part relative au poids, qui est calculé en fonction du nombre d'occurrences et de la fiabilité d'une source. Par conséquent, étant donné que les fiabilités sont différentes, pour un même nombre d'occurrences, la part est différente et, ipso facto, la correctness aussi. Néanmoins, ce problème est difficilement corrigeable, et ne pose pas vraiment problème. D'ailleurs, avec une valeur de correctness égale à « 0,50 » pour les deux sources il subsiste une indétermination quant aux choix de la valeur à considérer. Ici, il n'en est rien. Nous pouvons donc considérer l'évaluation de la correctness 5 comme acceptable.

| Source | FS | NOS | Last Date | VS | MaxFiabS | Match | MatchNOS | PoidsS |
|---|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1000 | 10-oct | Dupont | 0,9 | 1,00 | 1000,00 | 900 |
| Formulaire 2 | 0,7 | 1000 | 15-oct | Dupond | 0,7 | 1,00 | 1000,00 | 700 |
| Formulaire 3 | 0,9 | 0 | 20-oct | Dupond | 0,9 | 0,00 | 0,00 | 0 |
| | | 2000 | | | | | | 1600 |

**Aspect Multivaleurs**

| NV | NOV | MaxFiabV | SumPoids | PartPoidsV | SumMatchNOS | Renf | Correctness5 |
|---|---|---|---|---|---|---|---|
| Dupont | 1000 | 0,9 | 900 | 0,56 | 1000,00 | 1,00 | **0,56** |
| Dupond | 1000 | 0,9 | 700 | 0,44 | 1000,00 | 1,00 | **0,44** |
| | 2000 | | 1600 | 1,00 | 2000,00 | 2,00 | 1,00 |

FIGURE 5.23: Détail du calcul de correctness 5, deux sources de données de fiabilités différentes, deux valeurs distinctes, mille occurences

| | Valeur | Source | Occurrence | Fiabilité | Correctness Attendue | Correctness 4 | Correctness 5 |
|---|---|---|---|---|---|---|---|
| 0 | oui | 0 | 0 | - | 0 | OK | OK |
| 1 | oui | 1 | 1 | = | Conf | OK | OK |
| 3 | oui | 1 | 2 | = | 1>Corr>Conf | OK | OK |
| 3.1 | oui | 1 | n>2 | = | Corr>Corr(3) | OK | OK |
| 5 | oui | 1 | ∞ | = | 1 | OK | OK |
| 7 | oui | 2 | 1 | = | 1>Corr>Conf | OK | OK |
| 8 | oui | 2 | 1 | ≠ | 1>Corr>((conf1+conf2)/2) | OK | OK |
| 9 | oui | 2 | 2 | = | 1>Corr>Corr(3) | OK | OK |
| 10 | oui | 2 | 2 | ≠ | 1>Corr>Conf1 | OK | OK |
| 11 | oui | 2 | ∞ | = | 1 | OK | OK |
| 12 | oui | 2 | ∞ | ≠ | 1 | OK | OK |
| 13 | oui | ∞ | 1 | = | 1 | OK | OK |
| 14 | oui | ∞ | 1 | ≠ | 1 | OK | OK |
| 15 | oui | ∞ | 2 | = | 1 | OK | OK |
| 16 | oui | ∞ | 2 | ≠ | 1 | OK | OK |
| 17 | oui | ∞ | ∞ | = | 1 | OK | OK |
| 18 | oui | ∞ | ∞ | ≠ | 1 | OK | OK |
| 25 | non | 2 | 1 | = | Conf/N Sour. | OK | OK |
| 26 | non | 2 | 1 | ≠ | Corr -> Corr1 & Corr>Corr(25) | KO | OK |
| 27 | non | 2 | 2 | = | Corr>Corr(25) | OK | OK |
| 28 | non | 2 | 2 | ≠ | Corr -> Corr1 & Corr>Corr(27) | KO | OK |
| 29 | non | 2 | ∞ | = | 1/ N source | OK | OK |
| 30 | non | 2 | ∞ | ≠ | 1/ N source | OK | KO |
| 30.1 | non | 2 | 1:∞ ; 2:1 | ≠ | Corr=Corr1 & Corr=1 | OK | OK |
| 31 | non | ∞ | 1 | = | 0 | OK | OK |
| 32 | non | ∞ | 1 | ≠ | 0 | OK | OK |
| 33 | non | ∞ | 2 | = | 0 | OK | OK |
| 34 | non | ∞ | 2 | ≠ | 0 | OK | OK |
| 35 | non | ∞ | ∞ | = | 0 | OK | OK |
| 36 | non | ∞ | ∞ | ≠ | 0 | OK | OK |

FIGURE 5.24: Table de tests de la fonction de correctness

Finalement, la formule de la correctness retenue est la suivante :

$$Correctness = PartPoidsV * Renf$$

Pour une valeur j :

$$Correctness_j = PartPoidsV_j * Renf_j$$

$$\iff \frac{SumPoids_j}{SumPoidsT} * Renf_j(SumMatchNOS_j, MaxFiabV_j)$$

$$\iff \frac{\sum_{i \in S \wedge VN_j = VS_i} FS_i * NOS_i}{\sum_j (\sum_{i \in S \wedge VN_j = VS_i} FS_i * NOS_i)} * \sum_{i \in S \wedge VN_j = VS_i} (\frac{FS_i}{Max\{FS_k | k \in S \wedge VS_k = VS_i\}}) * NOS_i \sqrt{Max_j\{FS_i | i \in S \wedge VS_i = VN_j\}}$$

### 5.2.2 Freshness

La précédente section concernait l'étude de la correctness et la découverte d'une formule permettant de l'évaluer. Reprenons cette démarche cette fois pour la freshness.

#### 5.2.2.1 Découverte des éléments clés

**Date de dernière mise à jour** Afin de pouvoir évaluer à quel point une information est à jour, il est nécessaire de posséder sa date d'introduction ou de dernière « validation » quand cette information est confirmée via plusieurs acquisitions identiques. Nous l'appellerons date de dernière mise à jour.

**Durée de vie** Pour appliquer une fonction qui va fournir une valeur qui décroit (passe de 1 à 0) au fur et à mesure de l'écoulement du temps, il est indispensable de déterminer une borne supérieure (de temps) permettant d'indiquer à la fonction quand cette valeur doit atteindre le « 0 ». Cette borne correspond à la durée de vie de l'information. Nous l'exprimerons en millisecondes.

**Fonction de dépréciation** La façon dont la freshness doit décroitre n'est pas identique pour chaque valeur. Parfois, la diminution peut-être linéaire. Autrement dit, la vitesse de décroissance est la même à chaque instant. La fonction de dépréciation linéaire s'écrit comme suit :

$$f_{linéaire}(x) = 1 - x$$

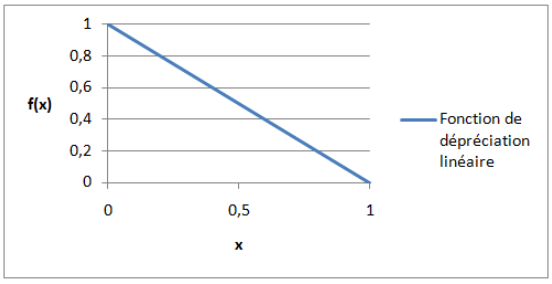



FIGURE 5.25: Graphe de la fonction de dépréciation temporelle linéaire

De façon plus réaliste, la diminution peut s'effectuer de plus en plus vite, ou inversement, de moins en moins vite correspondant respectivement à une fonction concave ou à une fonction convexe.

La fonction de dépréciation concave peut se définir de la sorte :

$$f_{concave}(x) = 1 - (x^3)$$

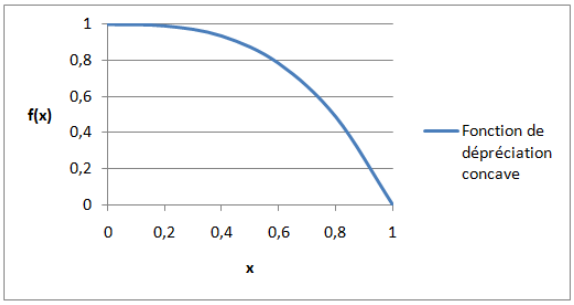



FIGURE 5.26: Graphe de la fonction de dépréciation temporelle concave

La fonction de dépréciation convexe peut se définir de la sorte :

$$f_{convexe}(x) = 1 - (x^{\frac{1}{3}})$$

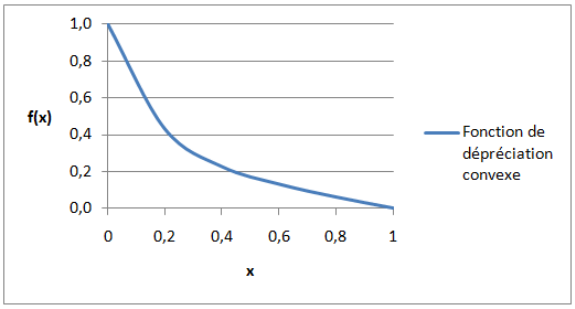



FIGURE 5.27: Graphe de la fonction de dépréciation temporelle convexe

En généralisant la fonction :

$$f(x) = 1 - (x^{\lambda})$$

- Lorsque $\lambda = 1$, la fonction $f(x)$ équivaut à la fonction $f_{linéaire}(x)$
- Lorsque $\lambda = 3$, la fonction $f(x)$ équivaut à la fonction $f_{concave}(x)$
- Lorsque $\lambda = 1/3$, la fonction $f(x)$ équivaut à la fonction $f_{convexe}(x)$
- Lorsque $\lambda$tend vers l'infini, la fonction $f(x)$ est concave et de la sorte :

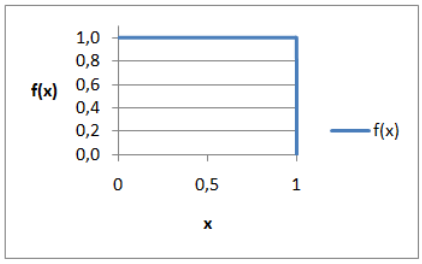



FIGURE 5.28: Graphe de la fonction de dépréciation temporelle concave lorsque $\lambda$ tend vers l'infini

– Lorsque $\lambda$ tend vers 0, la fonction f(x) est convexe et de la sorte :

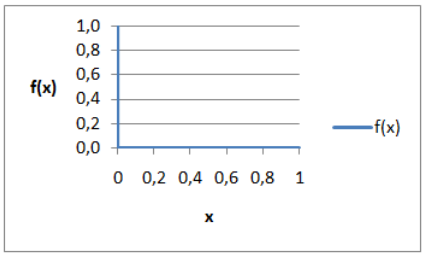



FIGURE 5.29: Graphe de la fonction de dépréciation temporelle convexe lorsque $\lambda$ tend vers l'infini

Nous voyons que cette fonction est très intéressante étant donné qu'on peut la faire changer de comportement (linéaire, concave et convexe) ainsi que d'intensité (vitesse de décroissance) en variant le seul paramètre $\lambda$. Nous l'utiliserons dans la suite de cette section.

Notons que n'importe quelle fonction décroissante peut être utilisée. Une approche non plus continue, mais discrète peut également être employée. Dans ce cas, la valeur diminuerait par paliers définis à différents temps $t_i$.

**Synthèse** Nous venons de découvrir trois éléments clés nécessaires à l'évaluation de la freshness d'une donnée. Il y a la date de dernière mise à jour, la durée de vie ainsi que la fonction de dépréciation temporelle. Ceux-ci devront donc être fournis pour chaque donnée dont on veut évaluer la freshness.

Le schéma du méta-modèle utilisateur doit tenir compte de ces modifications. La figure 5.30 est la nouvelle version de ce schéma. La durée de vie et la fonction de dépréciation ont été ajoutées en tant que champs de la table « VALEUR_ATTRIBUT » car chaque valeur peut avoir une durée de vie et

une fonction de dépréciation différentes. La date de dernière mise à jour a été, quant à elle, introduite dans l'association « possède » entre les tables « SOURCE » et « VALEUR_ATTRIBUT ».

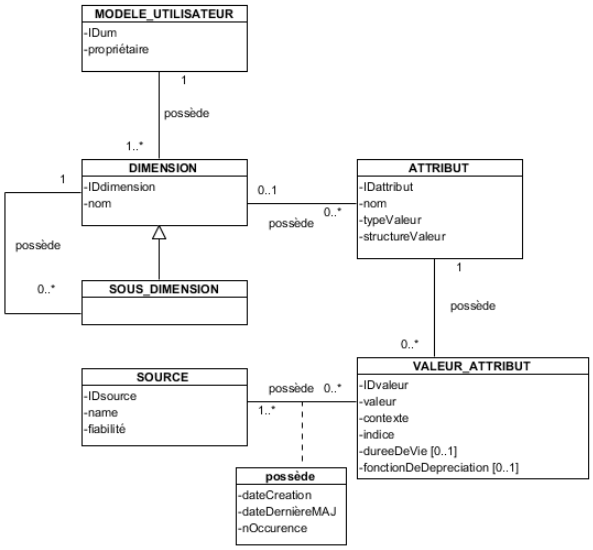



FIGURE 5.30: Méta-modèle utilisateur prennant en compte les données nécessaires pour la freshness

#### 5.2.2.2 Effets désirés

Les effets désirés de la freshness sont somme toute assez simple. En effet, elle ne dépend que de la durée de vie ($D$), de la date de dernière mise à jour ($T_0$) et de la fonction de dépréciation. Il y a donc très peu de cas.

Supposons que l'instant présent est l'instant $T$ :

- Si $T = T_0$ alors la freshness vaut 1.
- Si $T \geq T_0 + D$ alors la freshness vaut 0.
- Si $0 \leq T < T_0 + D$ alors on a $0 < Freshness < 1$. La valeur dépend de la fonction de dépréciation et de ses paramètres. Plus $T$ se rapproche de $T_0 + D$, plus la freshness doit être proche de 0. Inversement, plus $T$ est loin de $T_0 + D$, plus la freshness doit être proche de 1.

#### 5.2.2.3 Formules

Pour obtenir une évaluation de la freshness, nous allons nous servir de la fonction de dépréciation choisie. Reprenons sa forme concave :

$$f_{concave}(x) = 1 - (x^3)$$

L'argument « x » correspond à l'avancement. L'avancement est le rapport entre le temps écoulé depuis la dernière mise à jour et la durée de vie de la donnée :

$$Avancement = \frac{TempsEcoulé}{DuréeDeVie}$$

Le temps écoulé s'obtient en retranchant la date de dernière mise à jour de la date actuelle :

$$TempsEcoulé = DateActuelle - DateDeDernièreMiseAJour$$

Notons à titre informatif que les dates peuvent être représentées par des « timestamp » pour faciliter leur exploitation.

En somme la freshness s'obtient grâce à :

$$freshness = f(avancement)$$

$$\iff f(\frac{DateActuelle - DateDeDernièreMiseAJour}{DuréeDeVie})$$

Où $f(x)$est une fonction de décroissance quelconque prenant une valeur comprise dans l'intervalle [0,1] et produisant un résultat appartenant à ce même intervalle.

Illustrons cela avec un exemple (figure 5.31). Grâce à la date de dernière mise à jour (colonne « Last Date ») et à la date actuelle (colonne « Now Date ») le temps écoulé est déterminé. Ensuite le résultat de l'avancement s'obtient du rapport « 6 mois » sur « 1 an » (12mois). De là on applique chacune des fonctions de dépréciation temporelle décrites ci-dessus. Comme prévu, la fonction linéaire fournit « 0,5 » indiquant que la donnée sur laquelle nous travaillons a déjà perdu 50% de sa freshness. La fonction concave modélise une dépréciation faible au début et forte à la fin. C'est pourquoi, à mi-parcours, la valeur est supérieure à « 0,50 ». En revanche, la fonction convexe fournit une valeur en deçà de « 0,50 », manifestant une dégradation rapide dans les premiers instants et de moins en moins importante au fil du temps.

| Last Date | Durée Vie | Now Date | Temps écoulé | Avancement | Linéaire | Concave | Convexe |
|---|---|---|---|---|---|---|---|
| 15/06/2010 12:00 | 1 an | 15/12/2010 12:00 | 6 mois | 0,5 | 0,50 | 0,88 | 0,21 |

FIGURE 5.31: Détail du calcul de la freshness linéaire, concave et convexe

### 5.2.3 Indice de confiance

Nous venons de définir la manière d'évaluer les deux qualités qui entrent en jeux dans l'évaluation de la confiance en une donnée. Afin d'atteindre notre objectif, il est nécessaire de pouvoir agréger les deux valeurs résultantes de ces évaluations. C'est ce qui va nous occuper tout au long de cette section.

La correctness évalue l'aspect « correct » sans prendre en considération la perte d'exactitude de l'information dû à son âge. La freshness, quand à elle, s'occupe de déprécier une valeur (comprise entre 0 et 1) en fonction de l'écoulement du temps. Au temps $t$, l'indice de confiance résultera de la combinaison de la freshness et de la correctness. La fonction permettant d'évaluer la confiance prend donc en entrée deux paramètres (la freshness et la correctness) et fournit en sortie un indice dit de confiance compris entre 0 et 1.

$$IC_t(correctness_t, freshness_t)$$

Notons que l'indice résultant n'a de sens qu'au temps $t$.

Dans la suite de cette section, nous allons étudier une fonction d'agrégation permettant d'obtenir $IC_t$ par la combinaison de la $correctness_t$ et de la $freshness_t$.

#### 5.2.3.1 Agrégation par produit

La formule se définit comme suit :

$$IC_t(correctness_t, freshness_t) = freshness_t * correctness_t$$

Tentons d'appliquer cette fonction. La figure 5.32 intègre à la fois le calcul de la correctness et de la freshness ainsi que l'agrégation des deux par produit. La nouvelle colonne « Last Date » dans le tableau orienté valeur contient la date de dernière mise à jour pour cette valeur. La colonne « Av » contient la valeur d'avancement et celle intitulée « IC » la valeur de l'indice de confiance.

Tout d'abord, nous avons obtenu une évaluation de la correctness pour chacune des valeurs (l'adresse 1 et 2). Etant donné que chaque valeur prise en compte lors du calcul de la correctness peut avoir une date de dernière mise à jour différente, il est indispensable d'appliquer l'évaluation de la freshness sur chacune des valeurs du tableau orienté multi-valeurs. Effectivement, si la freshness n'était appliquée que sur la valeur activée, des effets indésirables pourraient se produire. Nous reviendrons sur ce phénomène dans la suite de ce chapitre. En attendant, nous admettrons cette hypothèse.

La correctness de la valeur « Adresse1 » est évaluée à « 0,51 » et la freshness à « 0,88 ». L'agrégation des deux vaut « 0,44 ». Pour obtenir ce résultat, nous avons utilisé la méthode d'agrégation par produit, utilisé une fonction de dépréciation concave et supposé que la « date actuelle » était le 15 juin 2010.

| Source | FS | NOS | Last Date | VS | MaxFiabS | Match | MatchNOS | PoidsS |
|---|---|---|---|---|---|---|---|---|
| Formulaire 1 | 0,9 | 1 | 15/06/2005 | Adresse1 | 0,9 | 1,00 | 1,00 | 0,9 |
| Formulaire 2 | 0,7 | 1 | 15/06/2000 | Adresse2 | 0,7 | 1,00 | 1,00 | 0,7 |
| Formulaire 3 | 0,9 | 0 | 1/06/2003 | Adresse1 | 0,9 | 0,00 | 0,00 | 0 |
| | | 2 | | | | | | 1,6 |

*Autres informations:*
*Now Date: 15/06/2010*

**Aspect Multivaleurs**

| NV | NOV | MaxFiabV | Last Date | SumPoids | PartPoidsV | SumMatchNOS | Renf | $Correctness_{now}$ | Durée Vie | Tps écoulé | Av | $Freshness_{now}$ | IC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Adresse1 | 1 | 0,9 | 15/06/2005 | 0,9 | 0,56 | 1,00 | 0,90 | **0,51** | 10 ans | 5 ans | 0,5 | **0,88** | 0,44 |
| Adresse2 | 1 | 0,7 | 15/06/2000 | 0,7 | 0,44 | 1,00 | 0,70 | **0,31** | 10 ans | 10 ans | 1 | **0** | 0,00 |
| | 2 | | | 1,6 | 1,00 | 2,00 | 1,60 | 0,81 | | | | | |

FIGURE 5.32: Détail de l'agrégation de la correctness et de la freshness par produit

Remarquons que l'indice de confiance pour l'« Adresse2 » est égal à 0 étant donné que la freshness est nulle. L'avantage d'une agrégation par produit est le zéro concédé par l'indice de confiance lorsqu'au moins un des deux paramètres d'agrégation est nul. Dans tout les cas, cela parait pertinent. En effet, si la freshness est nulle, l'information est donc jugée périmée peu importe sa correctness. Dès lors, l'indice de confiance est également nul manifestant le fait que nous ne pouvons pas faire confiance à une donnée périmée. Inversément, si la correctness est nulle, l'information est donc jugée non correcte peu importe sa fraicheur. Dès lors, l'indice de confiance est également nul manifestant le fait que nous ne pouvons pas faire confiance à une donnée totalement incorrecte. Le défaut de l'agrégation par produit est sa tendance à produire des valeurs assez basses. Toutefois, comme nous le verrons un peu plus loin, ce n'est qu'une histoire d'interprétation...

#### 5.2.3.2 Evolution de l'indice de confiance

La figure suivante présente l'évolution de l'indice de confiance (par rapport au temps) d'une donnée au sujet de l'adresse d'un internaute, possèdant une durée de vie de 4 unités de temps et une fonction de dépréciation concave.

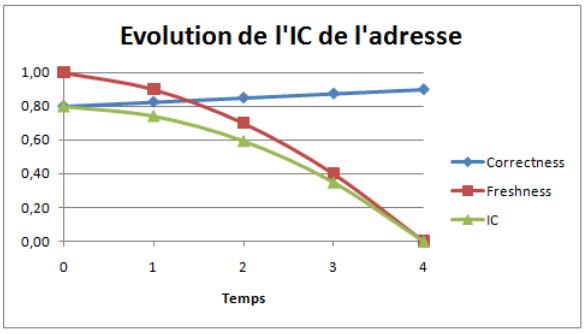



FIGURE 5.33: Evolution de l'indice de confiance de l'adresse d'un internaute

Assez logiquement, nous constatons que la correctness augmente légèrement, que la freshness diminue de façon concave (dû à sa modélisation par une fonction de dépréciation concave) et que l'IC s'estompe également. Rappelons qu'au temps $t$, l'IC est obtenu par multiplication de la valeur de freshness et de correctness. La figure 5.34 décrit les données relatives au graphique de la figure précédente.

| Temps | Correctness | Freshness | IC |
|---|---|---|---|
| 0 | 0,80 | 1,00 | 0,80 |
| 1 | 0,83 | 0,90 | 0,74 |
| 2 | 0,85 | 0,70 | 0,60 |
| 3 | 0,88 | 0,40 | 0,35 |
| 4 | 0,90 | 0,00 | 0,00 |

FIGURE 5.34: Données de la figure 5.33

#### 5.2.3.3 Valeur active

Qu'en est-il de la valeur active ? Quelle valeur le système va-t-il considérer ? A la section 5.2.1.1 nous avions discuté de ce sujet lorsque nous considérions uniquement la correctness. Ici le système possède l'indice de confiance résultant de l'agrégation de la correctness et de la freshness. La valeur choisie sera simplement la valeur correspondant à l'indice de confiance le plus élevé. La valeur active de la figure 5.32 est « Adresse1 » car elle possède un indice de confiance de 0,44, contre 0 pour « Adresse2 ». En cas d'égalité, la valeur active sera la dernière ayant été actualisée.

#### 5.2.3.4 Réflexions supplémentaires

Par soucis d'optimisation, il serait envisageable de ne pas stocker toutes les valeurs fournies par les sources de données étant donné que seule celle de meilleure confiance est retenue. Dans ce cas, l'unique

valeur présente en base de données est la valeur active. En fait, c'est moins la valeur proprement dite qui nous intéresse (dans ce contexte d'évaluation de la confiance) que l'information que nous en déduisons, c'est-à-dire, qu'elle soit égale ou différente (respectivement, renforcement ou affaiblissement) de la valeur active. Ainsi, la chaine de caractères contenant la valeur peut être remplacée par un "drapeau" indiquant si elle est divergente ou non. L'espace occupé est dès lors moins important.

Toutefois, cette optimisation est fortement déconseillée. Prenons un exemple pour nous en convaincre. La figure 5.35 décrit deux sources de données qui alimentent le modèle d'un utilisateur avec deux valeurs différentes pour l'adresse. L'axe des abscisses relate le temps (de $t_0$ à $t_8$ correspondant aux valeurs présentent sur l'axe de 0 à 8). L'axe des ordonnées exprime la valeur de l'indice de confiance. L'évaluation de la confiance nous permet d'activer la valeur « Adresse1 » en $t_0$ car elle est l'unique valeur acquise jusqu'alors. En $t_2$ survient une nouvelle valeur à propos de l'adresse de l'utilisateur : « Adresse2 ». A ce moment là, la valeur active reste « Adresse1 » car elle possède un meilleur indice de confiance. « Adresse2 » n'est donc pas sauvegardé en base de données par souci d'optimisation. Arrivé en $t_5$, il s'avère que l'indice de confiance de la première adresse devient inférieur à la seconde. Cela est un phénomène tout à fait normal étant donné que la confiance est sans cesse réévaluée, que la freshness diminue au fil du temps, et que de nouvelles sources peuvent diminuer la correctness. La logique voudrait que cette dernière valeur soit activée, cependant, cela est impossible. En effet, parce qu'elle n'a pas été enregistrée auparavant (par souci d'optimisation), nous ne possédons plus cette valeur ! Cela démontre la raison qui nous pousse à garder toutes les valeurs acquises concernant un utilisateur.

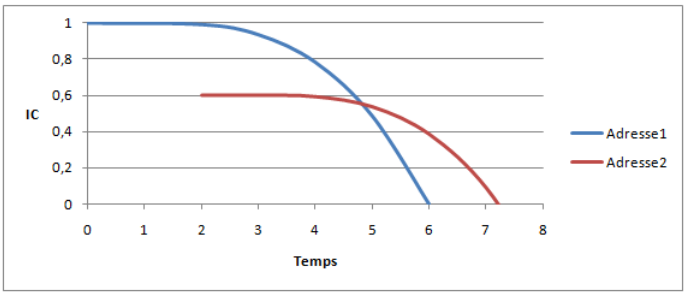



FIGURE 5.35: Exemple de l'évolution de l'indice de confiance de deux sources de données alimentant un attribut adresse

Une autre optimisation pensable est d'effectuer uniquement le calcul de la freshness sur la correct-

ness de la valeur active. La valeur active sera considérée comme telle simplement sur base de l'indice de correctness. Ce n'est également pas une bonne idée. Chaque donnée a été fournie à un temps « t » par une source de données. Deux données différentes pour un même attribut (dans l'exemple précédent : « Adresse1 » et « Adresse2 ») ont vraisemblablement été acquises à des moments différents. Cela signifie qu'elles ne possèdent pas la même freshness. Dès lors, le problème de la première optimisation pourrait également se produire.

## 5.3 Conclusion

L'objectif de ce chapitre, était de définir un indice de confiance le plus réaliste possible. C'est ce que nous avons fait. Pour cela nous avons intégré deux qualités : la correctness et la freshness, résultantes du chapitre précédent. Il fallait donc évaluer, sur base des informations que l'on dispose, si la donnée est bonne ou non d'abord vis-à-vis des valeurs que les sources de données fournissent (en considérant que les différentes sources n'ont pas toujours la même fiabilité), ensuite vis-à-vis de la fraicheur des données par rapport au temps. Après quoi, nous avons agrégé ces deux évaluations en une seule formant ce que l'on appelle l'indice de confiance.

Tout au long de notre démarche, nous avons émis des hypothèses basées sur le bon sens, sur la manière dont nous fonctionnons lorsque nous évaluons, implicitement, la pertinence de certaines informations. Et ceci afin d'obtenir un indice de confiance réaliste, représentatif de la « vérité ». Mais qu'est-ce que la vérité, la réalité ? Non, cette manière d'évaluer la confiance en une donnée ne donne en aucun cas LA vérité ! D'ailleurs, il n'y a pas vraiment de vérité, ni de réalité. Chacun a sa propre façon de percevoir les choses. Pour preuve, si nous devions effectuer une enquête auprès d'une population hétérogène afin d'évaluer la confiance que chacun a d'une même information, il y a de forte chance que nous obtenions des résultats différents. Néanmoins, il y aurait des tendances. C'est justement de ces tendances qu'il serait intéressant de s'approcher. Certes, ceci n'est pas l'objectif de ce mémoire, mais il aurait été dommage de ne pas en toucher un mot.

Nous n'avons aucunement la prétention d'avoir défini un indice permettant d'évaluer LA confiance. Cependant, une chose est certaine, nous avons défini un indicateur, qui, s'il est calculé à chaque fois de la même manière, permettra d'obtenir, pour une donnée, une signification concernant sa confiance. En effet, si on sait que lorsque l'indice de confiance vaut « 0,8 » pour une information on peut avoir

une bonne confiance en celle-ci, alors il présente une réelle plus value ! Et cela même si la valeur « 0,8 » n'est pas représentative de « la réalité ». La sémantique que l'on accorde à l'indice est plus importante que son aspect vraisemblable. C'est l'interprétation que l'on en fait qui compte. De plus, il permet la comparaison. Si cet indice vaut « 0,8 » pour une donnée « A » et « 0,5 » pour une donnée « B », nous pouvons affirmer, au vu des informations que nous possèdons, que nous avons plus confiance en « A » qu'en « B ». Voilà les véritables atouts de la définition d'un indice de confiance.

# Chapitre 6

# Prototype et pistes d'évaluation

Dans ce chapitre nous allons mettre en pratique les réflexions menées aux chapitres précédents par l'implémentation d'une application permettant l'évaluation de la qualité d'une valeur d'attribut, sur base des informations qui viennent alimenter cet attribut. Elle permettra d'une part, l'évaluation de la qualité sur base des informations présentes dans le modèle utilisateur, en calculant, pour chaque valeur d'attribut, un indice de confiance et en déterminant la valeur active, et d'autre part, l'utilisation d'une interface graphique afin de simuler, d'alimenter le modèle utilisateur et d'en constater les effets.

L'intérêt est double. D'abord elle permettra de prouver la faisabilité de nos réalisations. Ensuite, elle fournira un module-prototype du système d'évaluation de la qualité d'une valeur d'un attribut, tel que défini au chapitre précédent.

Nous évoquerons à la fin de ce chapitre quelques pistes de validation du modèle de qualité afin de s'assurer des bénéfices obtenus de cette étude.

Nous allons dans un premier temps présenter l'application et dans un second temps, discuter de l'intégration de ce dernier au sein d'un système de personnalisation. Nous cloturerons ensuite par quelques pistes de validation.

## 6.1 Application développée

L'application est avant tout un simulateur. Elle permet de simuler l'alimentation du modèle utilisateur avec des données issues de différentes sources possédant chacune une certaine fiabilité et de

calculer pour chaque valeur d'un attribut entrée, sa correctness, sa freshness et son indice de confiance. La valeur possédant l'indice de confiance le plus élevé est indiqué comme étant la valeur active.

Dans cette section nous décortiquerons d'abord l'interface de l'application avant de présenter la structure de donnée et l'algorithme de calcul de l'indice de confiance.

### 6.1.1 Interface graphique

L'interface de l'application développée est la suivante :

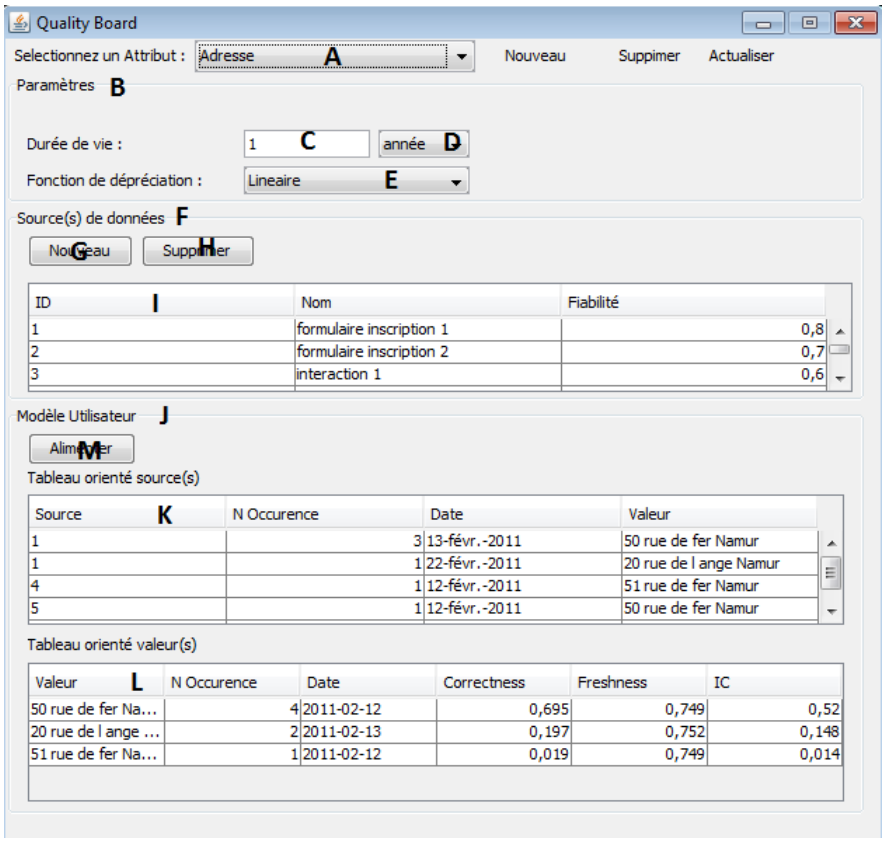



FIGURE 6.1: Interface graphique de l'application developpée

Pour une lisibilité optimale, l'interface de l'application (figure 6.1) ne présente qu'un seul attribut à la fois. Nous avons la possibilité, via une liste déroulante (A) de sélectionner l'attribut à considérer. Juste à droite, les boutons « Nouveau », « Supprimer » et « Actualiser » permettent respectivement d'ajouter un attribut, de supprimer un attribut et d'actualiser les données (recalculer et recharger les

données à afficher).

La rubrique « Paramètres » (B) va permettre de définir la durée de vie (via le champ « C » et la liste déroulante « D » permettant de sélectionner l'unité [1] de temps) et la fonction de dépréciation (via la liste déroulante « E » permettant de sélectionner la fonction [2] ) d'une valeur liée à cet attribut.

La rubrique « Source(s) de données » (F) permet d'ajouter une source de données via le bouton « G », de supprimer une source de données via le bouton « H » et de visualiser sous forme de tableau « I » la liste de toutes les sources de données répertoriées. Ce tableau contient trois colonnes : ID (identifiant de la source de données), Nom (nom de la source de données) et Fiabilité (fiabilité de la source de données). Lorsque le bouton « G » est actionné, une fenêtre permettant l'ajout d'une source (nom et fiabilité) apparait (figure 6.2).

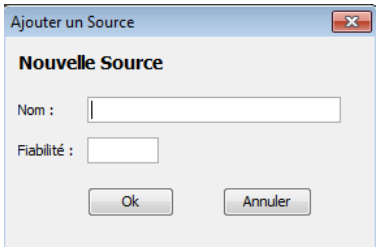



FIGURE 6.2: Fenêtre d'ajout d'une nouvelle source de données

La rubrique « Modèle Utilisateur » (J) contient le tableau orienté source(s) simplifié (K) et le tableau orienté valeur(s) simplifié (L) tous deux présentés au chapitre précédent. Le premier contient pour chaque ligne, l'identifiant de la source concernée, le nombre d'occurrences que la source a fourni pour cette valeur, la date à laquelle cette source a ajouté cette valeur pour la dernière fois, ainsi que la valeur. Le second tableau contient pour chaque ligne, la valeur de l'attribut, le nombre d'occurrence de cette valeur par le biais de n'importe quelle source, la date à laquelle cette valeur a été ajoutée pour la dernière fois, la correctness de cette valeur, la freshness de cette valeur ainsi que l'indice de confiance de cette valeur. Notons que la ligne correspondant à la valeur active dans ce second tableau est automatiquement stylée en « gras ». Le bouton « Alimenter » (M) également présent dans cette rubrique offre la possibilité de venir nourrir le modèle utilisateur avec une nouvelle valeur pour l'attribut sélectionné en « A ». Lorsque ce bouton est actionné la fenêtre suivante (figure 6.3)

1. Milliseconde, seconde, minute, heure, jour, mois ou année
2. Trois fonctions sont prédéfinies : linéaire (figure 5.25), concave (figure 5.26) et convexe (figure 5.27)

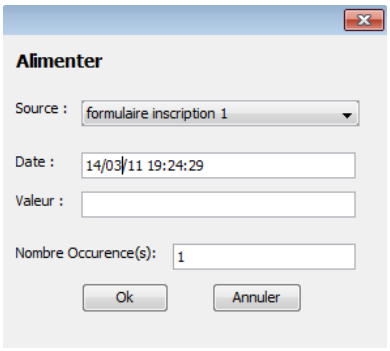



FIGURE 6.3: Fenêtre d'alimentation du modèle utilisateur

apparait. Elle permet de spécifier la source qui va alimenter le modèle utilisateur, la date à laquelle cette alimentation se produit, la valeur de l'attribut qui va alimenter le modèle utilisateur ainsi que le nombre de fois que cette alimentation va se produire (via le nombre d'occurrence(s)).

### 6.1.2 Structure de données

L'application utilise une base de données externe dont le schéma (figure 6.4) est un sous-ensemble du schéma de base de données présenté à la figure 5.30. La prise en compte des tables « MODELE_UTILISATEUR », « DIMENSION » et « SOUS_DIMENSION » n'est guère nécessaire pour les fonctionnalités de l'application. Celle-ci va se contenter, sur base d'un attribut donné, de ses valeurs et de ses sources, d'estimer les indices de correctness, de freshness et de confiance et de simuler l'alimentation du modèle utilisateur par une valeur de cet attribut issue d'une source de données. Le sous-ensemble de tables « ATTRIBUT », « SOURCE » et « VALEUR_ATTRIBUT » est donc suffisant. Notons tout de même que les attributs « dureeDeVie » et « fonctionDeDepreciation » sont ici liés à la table « ATTRIBUT » plutôt qu'à la table « VALEUR_ATTRIBUT » tel que représenté à la figure 5.30. Par conséquent, la durée de vie et la fonction de dépréciation d'une valeur seront donc identiques pour chaque valeur d'un attribut. Ce choix a été effectué dans le but de simplifier l'interface.

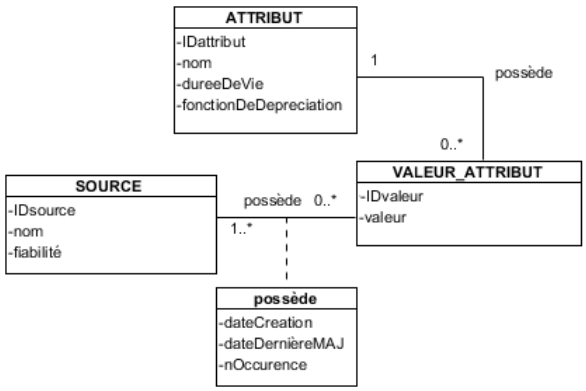



FIGURE 6.4: Schéma de la base de données de l'application

Au lancement de l'application, la connexion avec la base de données est établie et les données qui y sont présentes sont chargées dans une structure d'objets Java. Chaque table de la figure 6.4 possède son correspondant en liste d'objets Java. Ainsi, les données de la table « ATTRIBUT » sont chargées dans une liste d'objets « Attribut » possédant un identifiant, un nom, une durée de vie et une fonction de dépréciation. La classe d'association « possède » bénéficie également d'un correspondant en liste d'objets Java.

La manipulation des données se fait directement via les données chargées en mémoire. Lorsque l'utilisateur décide de quitter l'application, les données sont alors enregistrées en base de données. Le choix de ne pas interagir directement avec les données de la base de données se justifie par le fait qu'intégré à un système de personnalisation, le module permettant d'évaluer la qualité des données recevra probablement en entrée un modèle utilisateur, mais ne devra pas lui-même accéder directement à la base de données pour l'obtenir.

### 6.1.3 Algorithme d'évaluation de la qualité des informations web

#### 6.1.3.1 Description de l'algorithme

L'application contient un module permettant d'évaluer la qualité des informations web. Ce module est une implémentation de la formule obtenue au chapitre précédent, résultat de nos réflexions sur l'estimation d'un indice de confiance.

Cet algorithme prendra en entrée un attribut respectant la structure de données définies ci-dessus. Il contiendra donc zéro, une ou plusieurs valeur(s) d'attribut fournie(s) par au moins une source de données. Chaque entité (attribut, valeur d'attribut et source) possèdera les attributs de table définis à la figure 6.4.

En sortie, l'algorithme renverra l'indice de confiance telle que nous l'avons défini au chapitre précédent.

Nous avons vu que celui-ci s'obtenait en combinant, à un moment t, un indice de correctness avec un indice de freshness :

$$IC_t(correctness_t, freshness_t)$$

La formule de correctness est la suivante :

$$Correctness = PartPoidsV * Renf$$

Et la freshness s'obtient quant à elle de cette manière :

$$freshness = f(avancement)$$

où $Avancement = \frac{Tempsécoulé}{Duréedevie}$

où $TempsEcoulé = DateActuelle - DateDeDernièreMiseAJour$

Où f(x) est la fonction de dépréciation choisie.

#### 6.1.3.2 Algorithme

L'algorithme implémenté de calcul de l'indice de confiance est basé sur les formules précédentes. Il est présenté dans l'annexe C de ce document.

## 6.2 Intégration du module d'évaluation de la qualité des informations web au sein d'un système de personnalisation

Le module permettant d'évaluer la qualité des informations web de l'application s'intégrera plus ou moins facilement dans un système de personnalisation web selon la concordance de la structure de données de ce dernier avec celle définie à la figure 5.30.

Le module permet l'évaluation des données contenues dans le modèle utilisateur. Ainsi, pour chaque donnée nous disposons d'une information supplémentaire quant à la confiance que l'on peut lui accorder. Cette information devra être prise en compte par le système de personnalisation lorsqu'il décidera, sur base des données contenues dans le modèle utilisateur, de sélectionner un contenu à afficher. Rappelons-nous que l'étape consistant à sélectionner un contenu web sur base d'un modèle utilisateur n'est autre que le matching. Il serait dès lors pertinent d'inclure les indices de confiance dans le modèle utilisateur qui servira de base à l'algorithme du matching afin de sélectionner un contenu plus approprié.

L'intégration de la qualité dans le modèle utilisateur modifie le processus de personnalisation présenté à la section 3.2. Il comprend dorénavant une étape supplémentaire : l'évaluation de la qualité. Dans celle-ci, sur base de toutes les informations contenues dans le modèle utilisateur, la valeur active va être déterminée et le modèle utilisateur va être étendu en calculant un indice de confiance pour chaque valeur d'attribut. Le matching quant à lui prendra désormais en compte le modèle utilisateur étendu. Aucun changement n'est à envisager en ce qui concerne les activités parallèles.


FIGURE 6.5: Processus de personnalisation web incluant l'évaluation de la qualité

## 6.3 Pistes de validation

Afin de "vérifier" les résultats obtenus, il est nécessaire de procéder à une validation du modèle de qualité dégagé. Pour cela, nous proposons deux pistes de validation.

Rappelons que cet indice n'a pas la prétention d'être représentatif de la réalité, mais tente de donner une indication quant à la confiance d'une donnée. Toute validation doit donc en tenir compte.

### 6.3.1 Comparaison d'un système de personnalisation web avec et sans le module de qualité

Comme première piste de validation nous proposons d'effectuer une comparaison des résultats d'un système de personnalisation web traditionnel (SPWT) avec ce même système auquel a été intégré le

module de qualité (SPWQ). Notons que nous avons discuté à la section 6.2 de l'intégration de ce module dans un tel système. Voyons comment nous pouvons tester le modèle de qualité.

Nous distinguons trois manières de valider le modèle de qualité par comparaison.

La première façon est d'effectuer une batterie de tests sur les deux systèmes (SPWT et SPWQ) et d'observer et de comparer les résultats de ces tests. Afin de tester tous les cas, il est intéressant de reproduire les situations présentées dans le tableau repris à la figure 5.6 indiquant les effets désirés. Il est également intéressant d'effectuer ces tests à différents moments dans le temps (par exemple : au début de la création du modèle utilisateur, après quelques visites et après un nombre important de visites), afin d'étudier l'amélioration, la stabilité ou la dégradation des résultats au fur et à mesure de l'utilisation du système.

La seconde manière de procéder pourrait être via la réalisation d'une enquête de satisfaction générale sur l'utilisation du site avec un SPWT, une seconde avec un SPWQ et de comparer les résultats.

Une troisième façon serait d'effectuer une enquête spécifique sur la personnalisation du site web via un questionnaire dans lequel l'internaute donne une note de 0 à 10 concernant la pertinence du contenu affiché sur une page. Un internaute devra répondre deux fois au questionnaire : une fois concernant l'utilisation du SPWT et une seconde fois pour le SPWQ.

Enfin, un quatrième moyen d'effectuer une validation du module de qualité au sein d'un système de personnalisation serait de mettre, à coté de chaque contenu, une petite icone (par exemple une croix) permettant à l'internaute d'indiquer au système que le contenu n'est pas approprié. Sur base de ces données de feedback récoltées, et plus particulièrement du nombre de contenus non-adéquats, nous pourrons apprécier la pertinence de la personnalisation. Cette technique de validation présente l'avantage d'être "automatique", dans le sens qu'il n'est pas nécessaire de mobiliser des ressources afin de réaliser des tests ou des enquêtes.

### 6.3.2 Questionnaires

La seconde piste de validation que nous proposons, moins efficace et représentative, est d'effectuer une enquête en deux temps, non basée sur un système de personnalisation. D'abord, il serait demandé aux sujets de répondre à un certain nombre de questions, parfois redondantes mais sous différentes formes, permettant de dresser leur modèle utilisateur. Ensuite, ces données seront encodées et transmises au module d'évaluation de la confiance. Les indices de confiance pour chaque information seraient

alors proposés aux sujets. Après quoi, ces derniers leurs donneront une note (par exemple entre 0 et 10).

Cette méthode est moins efficace et représentative par le fait qu'elle ne reproduit pas exactement les conditions d'utilisations. Dans le cas où nous ne disposons pas du système de personnalisation web, il n'est pas évident d'évaluer un module (en l'occurence le module d'évaluation de la qualité des informations) indépendamment de son contexte d'utilisation. C'est pourquoi nous proposons tout de même cette approche.

# Chapitre 7

# Conclusion et Perspectives

## 7.1 Conclusion

La personnalisation web, inscrite dans une philosophie centrée sur l'individualisation des produits et services plutôt que sur la diffusion de masse, et solution à un égarement certain dans une quantité de plus en plus conséquente de données présentes sur certains sites internet, prend de plus en plus d'ampleur. La question de la pertinence des résultats, intrinsèquement liée à la qualité des informations collectées, est dès lors plus qu'à l'ordre du jour. C'est dans ce cadre que s'est inscrit ce mémoire.

Les enjeux concernent tant le business que l'internaute. Du premier découlent des propositions de contenu (informations, produits ou services) plus approprié, de manière à susciter un réel intérêt de la part de l'utilisateur. Pour les sites dits d' « e-business », cet utilisateur n'est autre qu'un consommateur potentiel. Dans ce cas, la personnalisation fait office de technique marketing visant à augmenter les ventes. Egalement, des rapports, statistiques et généralisation sur les utilisateurs peuvent être générés afin de mieux cibler les campagnes de marketing, les promotions de produits ou services. Le second, verra sa navigation améliorée point de vue contenu, visuel et interactionnel. En effet, les informations qui lui sont proposées sont plus pertinentes et le site bénéficie d'une meilleure flexibilité (visuelle et de navigation) par la prise en compte de ses intérêts et préférences. Le but ultime de la personnalisation web est avant tout de satisfaire l'utilisateur.

Afin d'étudier cette question à propos de la qualité des informations de personnalisation web, nous avons adopté une méthodologie comprenant 5 étapes.

Dans la première étape (chapitre 2), nous avons étudié la question de la modélisation utilisateur.

Pour cela, nous avons commencé par contextualiser, définir le modèle utilisateur et d'en proposer une représentation (basée sur une synthèse de la littérature). Nous avons ensuite taché de déterminer un processus de modélisation utilisateur dans un contexte d'utilisation. Enfin, nous avons examiné la modélisation de stéréotypes, cas particuliers de la modélisation utilisateur.

La seconde étape de notre méthodologie (chapitre 3) consistait en l'étude de la littérature scientifique au sujet de la personnalisation web et la définition d'un processus (de personnalisation web). Nous avons également abordé le sujet des limites et inconvénients des techniques actuelles.

Le chapitre 4 sur l'analyse de la qualité des informations web constituait la troisième étape de notre étude globale. Dans celui-ci nous avons d'abord dressé un bref état de l'art sur la qualité des informations, puis étudié les défauts présents au sein d'un système de personnalisation avant de sélectionner, en deux temps, les critères de qualités pertinents pour notre objet d'étude.

Après avoir posé les bases lors des trois premières étapes, nous nous sommes penchés au chapitre 5, sur le cœur du sujet en tentant d'améliorer la pertinence des systèmes de personnalisation web via la définition d'un indice de confiance basé sur l'étude des qualités des informations précédemment réalisée. Nous sommes partis du constat qu'il est à priori difficile de garantir la qualité intrinsèque d'une donnée. C'est pourquoi nous sommes remontés jusqu'aux sources de données. Celles-ci nous ont permis, via une analyse, d'obtenir un bon nombre d'informations précieuses quant aux données qu'elles ont fourni. Partant de là, nous avons dégagé une première formule permettant d'estimer le niveau de vérité, de justesse d'une information (la correctness) et une seconde pour l'évaluation de la fraicheur d'une donnée (la freshness). Ces deux formules sont ensuite multipliées afin d'obtenir l'indice de confiance de l'information.

Enfin, la dernière étape fut de prouver la faisabilité du modèle d'évaluation de la qualité précédemment défini par l'implémentation d'un prototype (chapitre 6). Celui-ci appuie les résultats obtenus des réflexions faites lors de ce travail non seulement en implémentant les formules définies au chapitre 5 mais également en offrant un simulateur permettant l'alimentation du modèle utilisateur tel que défini au chapitre 2. Pour clôturer ce chapitre, quelques pistes d'évaluation ont été proposées afin de valider le modèle de qualité.

Ce mémoire apporte aux techniques de personnalisation basées sur la modélisation utilisateur un réel atout. Jusqu'alors, ces techniques résonnaient sur une incertitude trop profonde dont les résultats affichaient un manque de pertinence flagrant. Il y a bon espoir qu'avec l'intégration au sein d'un

système de personnalisation d'un module permettant d'évaluer la qualité, nous constations une nette amélioration de la pertinence de cette personnalisation.

Il est important de noter que ce travail ne prétend en rien résoudre totalement le problème lié à l'incertitude des données. Celles-ci, font intrinsèquement l'objet d'une certaine qualité, définie selon un ensemble de critères variant d'un contexte à l'autre. L'évaluation de ces critères reste subjective et réductrice. Tout au long de ce mémoire, des hypothèses ont été faites, des choix, certes réducteurs, ont été réalisés, débouchant sur une formule d'estimation de la confiance d'une donnée qui se veut plus indicative que représentative. Derrière les données évaluées, se trouve une certaine réalité à laquelle nous n'avons pas accès, non seulement par le manque d'informations, mais aussi par la limite de représentativité de ces informations. Il est dès lors impossible, à l'heure actuelle et avec les moyens dont nous disposons, de déterminer un indice de confiance réaliste au point de donner LA vérité. Néanmoins, avec celui-ci nous affichons clairement l'ambition de fournir un indicateur qui, par comparaison des résultats qu'il produit, apporte une signification quant à la confiance que l'on peut avoir en une information. La sémantique que nous lui accordons doit être plus importante que son aspect vraisemblable. C'est donc principalement l'interprétation que l'on en fait qui prime.

## 7.2 Perspectives

Plusieurs travaux peuvent être réalisés afin de compléter les résultats de ce mémoire. En voici un bref aperçu.

### 7.2.1 Modèle utilisateur sémantique

Dans notre modélisation utilisateur, lorsqu'une source de données vient alimenter un modèle utilisateur, aucune vérification sémantique n'est prise en compte. Si la valeur de l'attribut fournie par la source y est déjà présente, le nombre d'occurrences augmente d'une unité. Dans le cas contraire, la nouvelle valeur d'attribut est enregistrée et fait donc figure de valeur discordante vis-à-vis des autres. Le mécanisme permettant de vérifier si cette valeur est déjà présente ou non dans le modèle utilisateur ne se base que sur le seul critère syntaxique (cf distance syntaxique [34, 10, 24]).

L'amélioration proposée serait la prise en compte d'un critère sémantique plutôt que syntaxique. Pour cela, il serait opportun d'associer à chaque attribut du modèle utilisateur un domaine de valeur

attendu. Chaque attribut serait lié à une base de connaissance particulière.

Ainsi, si l'attribut « intérêt » d'un modèle utilisateur possède préalablement la valeur « Théâtre », l'ajout de la valeur « Art dramatique » par une source de données ne devrait pas donner lieu à l'ajout d'une nouvelle ligne dans la table « VALEUR_ATTRIBUT », mais plutôt à l'accroissement d'une unité du nombre d'occurrences de la valeur « Théâtre ». En effet, avoir pour centre d'intérêt le théâtre est significativement la même chose que l'art dramatique. Sans cette amélioration, et en supposant qu'un individu ne puisse posséder qu'un seul centre d'intérêt, le module d'évaluation de la qualité considérerait les valeurs comme distinctes et accorderait dès lors une moindre confiance en celles-ci.

Outre ce système de gestion de synonymes, il serait également intéressant d'intégrer un mécanisme permettant le calcul de la distance sémantique. Si deux valeurs sont sémantiquement très proches, mais possèdent une syntaxe quelque peu différente, il y a de forte chance pour qu'elles représentent le même concept [34, 10, 24]. Par exemple, lorsque pour l'attribut « Moyen de locomotion » l'utilisateur entre « Voitrue », en comparant cette valeur avec les valeurs déjà présentes dans le modèle utilisateur pour cet attribut et/ou en la comparant par rapport à une base de connaissances sur les moyens de locomotions, le système pourrait déduire que l'utilisateur avait plutôt voulu entrer « Voiture ». Pareillement, si le modèle utilisateur possède pour l'attribut « Adresse » la valeur « 50, rue de fer », et qu'une source de données fournissait la nouvelle valeur « rue de fer, 50 », par le calcul de la distance sémantique de ces deux valeurs le système pourrait déduire qu'elles sont identiques.

Enfin, intégrer la sémantique dans la modélisation utilisateur [8] permettrait également la déduction. Si un modèle utilisateur tolère l'ajout de plusieurs centres d'intérêts et que le centre d'intérêt « Programmation » y est déjà présent, lorsqu'une source de données ajoute l'intérêt « Java », le système pourrait être capable de résoudre l'ambiguïté liée à la polysémie de ce terme, et de déduire (avec une certaine probabilité) qu'il s'agit du langage de programmation plutôt que de la danse ou même de l'île [50].

Passer du simple modèle utilisateur au modèle utilisateur sémantique [8] accroîtrait d'avantage connaissance que l'on a d'un utilisateur et par conséquent, de la pertinence de la personnalisation web. Néanmoins, ces mécanismes doivent être utilisés avec beaucoup de précaution. Trop de sémantique pourrait diminuer la connaissance au sujet d'un utilisateur. Prenons le cas de M. Dupond. Le système pourrait considérer, qu'au vu du nombre de personnes possédant le nom « Dupont », celui-ci s'écrive

avec un « t » et non un « d » comme l'a indiqué cet utilisateur. Or ce n'est pas le cas. Le système se baserait donc sur un nom qu'il pense être correct alors qu'il n'en est rien.

### 7.2.2 Fédération de modèles utilisateurs

Le modèle utilisateur tel que défini dans ce document n'est pas prévu pour être partagé avec d'autres systèmes de personnalisation ou bases de connaissances. A l'heure de l'intégration des données et de la « business intelligence », l'intérêt que peut présenter l'importation et l'exportation de modèles utilisateurs et de modèles de stéréotypes ou leur utilisation croisée (utilisation de plusieurs modèles utilisateur au sujet d'un utilisateur) est substantiel. La naissance d'une « fédération » de modèles utilisateurs, regroupant les modèles utilisateurs issus de différents sites internet, applications ou autres systèmes, apporterait une réelle plus value. Un modèle utilisateur n'étant pas l'autre (chacun possédant ses propres dimensions, attributs, etc.), il serait nécessaire d'associer une sémantique à chaque élément de la structure d'un modèle utilisateur. Sans cela, l'attribut « Nom » du méta-modèle utilisateur d'un site web A ne correspondrait pas à l'attribut « Name » du méta-modèle utilisateur d'un site web B, alors que sémantiquement, il modélise la même connaissance au sujet de l'utilisateur.

La mise en place d'un système de personnalisation basé sur une modélisation utilisateur fédérée incluant un module d'évaluation de la qualité de données présentes dans les modèles utilisateurs verra ce module légèrement adapté. Les données au sujet d'un utilisateur proviendraient dès lors de sources de données de différents sites internet. Un site web n'étant pas l'autre, la fiabilité d'une source devra également prendre en compte la fiabilité du site internet (car certains sites web sont moins dignes de confiance que d'autres).

### 7.2.3 Fiabilité des sources de données

Lors de l'étude de la fiabilité des sources de données nous avons souligné les limites que présentait le système quant à son adaptation automatique en fonction du temps. Rappelons que la fiabilité d'une source de données est le résultat de la combinaison de la fiabilité de la méthode d'acquisition de données et la fiabilité de l'utilisateur. A force de constater qu'une certaine technique d'acquisition de données (par exemple un certain formulaire HTML) fournit des valeurs peu fiables, le système pourrait réévaluer la fiabilité qui lui a été attribuée dans le cas où elle serait trop élevée. Bien sûr, l'idée serait d'application autant pour un ajustement négatif que positif. Similairement, si un utilisateur

a tendance à fournir des valeurs erronées, le système pourrait automatiquement diminuer la fiabilité de cet utilisateur et inversement. L'amélioration proposée consiste donc en un système d'apprentissage [7, 28] sur les sources de données.

### 7.2.4 Validation d'un système de personnalisation basé sur la modélisation utilisateur et l'évaluation de la qualité

Dans le chapitre 6 nous avons réalisé un prototype permettant d'évaluer la qualité des informations web. Nous avons évoqué brièvement à la section 6.2 comment un tel module pourrait s'intégrer dans un système de personnalisation web basé sur la modélisation utilisateur telle que définie au chapitre 2. La prochaine étape serait de réaliser un tel système afin de constater, mesurer les apports et les bénéfices de ce module d'évaluation de la qualité telle qu'expliqué à la section 6.3.1, ou à défaut, une enquête via questionnaires comme l'explique la section 6.3.2.

# Bibliographie


[1] G. Amato and U. Straccia. User profile modeling and applications to digital libraries. *In Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries, Paris, France*, 1999.

[2] L. Ardissono and A. Goy. Tailoring the interaction with users in electronic shops. *Proceedings of the Seventh International Conference, New York*, 1999.

[3] S. Berisha-Bohe and B. Rumpler. Modèle évolutif d'un profil utilisateur. application à la recherche d'information dans une bibliothèque numérique de thèses. *Quatrième conférence francophone en Recherche d'Information et Applications. Ecole Nationale Supérieure des Mines de Saint-Etienne*, 28 - 30 mars 2007.

[4] L. Berti. Qualité de données multi sources et recommandation multicritère. *INFORSID*, 1999.

[5] P. Bertolazzi and M. Scannapieco. Introducing data quality in a cooperative context. *in Proceedings of the 6th International conference on Information Quality (IQ'01), Boston, MA, USA*, 2001.

[6] M. Bouzeghoub. A framework for analysis of data freshness. *Proceedings of the 2004 international workshop on Information quality in information systems*, 2004.

[7] A. Cornuejols and L. Miclet. *Apprentissage Artificiel : Concepts et Algorithmes.* Eyrolles, 2002.

[8] B. Brandherm M. Schmitz D. Heckmann, T. Schartz and M. von Wilamowitz-Moellendorff. Gumo - the general user model ontology. *Proceedings of 10th International User Modeling Conference*, 2005.

[9] Y. Lee D. Strong and R. Wang. Data quality in context. *Communications of the ACM*, 40(5), 1997.

[10] F. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 1964.

[11] K Djemai and K. Ghouali. Conception et réalisation d'un système personnalisé pour l'interrogation d'une base de données. 2008.

[12] F. Geerts X. Jia G. Cong, W. Fan and S. Ma. Improving data quality consistency and accuracy. *VLDB S07*, September 2007.

[13] R. Harrathi. Facteurs de qualité et personnalisation de l'information. *Mémoire de master, Institut National des Sciences Appliquées de Lyon*, 2005.

[14] R. Harrathi and S. Calabretto. Un modèle de qualité de l'information. *Actes des Journées Extraction et Gestion de Connaissances (EGC), Lille, France*, 2006.

[15] K. Sugiyama. K. Hatano and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. *In Proceeding of the 13th International World Wide Web Conferences (WWW), New York, USA, p. 675-684*, 2004.

[16] M. Jarke and Y. Vassiliou. Data warehouse quality design : A review of the dwq project. *In Proceedings of the International Conference on Information Quality (IQ), Cambridge.*, 1997.

[17] J. Kay. Stereotypes, student models and scrutability. *Intelligent tutor systems, 5th international conference*, 1839, 2000.

[18] N. J Kelly. Understanding implicit feedback and document preference : a naturalistic study. *In PHD dissertation. Ritgers University, New Jersey*, January 2004.

[19] A. Kobsa. Generic user modeling systems. *Journal On User Modeling and User-Adapted Interaction*, 2007.

[20] D. Kostadinov. Personnalisation de l'information et gestion des profils utilisateurs. *Rapport de DEA, Université de Versailles, France*, 2003.

[21] D. Kostadinov. Personnalisation de l'information - une approche de gestion de profils et de reformulation de requêtes. 2008.

[22] L. Console L. Ardissono and L. Torre. On the application of personalization techniques to news servers on the www. *Lamma and P.Mello (eds.) : AI*IA : Advances in Artificial Intelligence. Lecture Notes in Artificial Intelligence*, 2000.

[23] P. Torasso F. Bellifemine A. Chiarotto A. Difino L. Ardissono, C. Gena and B. Negro. User modeling and recommendation techniques for personalized electronic program guides. *Personalization and User-adaptive Interaction in Digital TV. Dordrecht : Kluwer Academic Publishers*, 2004.

[24] V. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Probl. Inf. Transmission*, 1965.

[25] H. Lieberman and L. Letizia. An agent that assists web browsing. *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, p. 924-929*, 1995.

[26] S. Calabretto N. Denos D. Kostadinov A. Nguyen M. Bouzeghoub, R. Harrathi and V. Peralta. Accès personnalisé aux informations : approche dirigée par la qualité. *Actes du XXVème congrès*, 2007.

[27] G. Saake M. Fugini, M. Tamer Ozsu and K. Sattler. Data quality on the web. *Report on the Dagstuhl Seminar*, 2003.

[28] D. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[29] C. Marchetti and M. Mecella. Data quality notification in cooperative information systems. *Project DaQuinCTS - Methodologies and tools for Data Quality Cooperative Information Systems, MIUR, COFIN*, 2001.

[30] D. Meddouri. http ://www-lor.int-evry.fr/ maknavic/documents/cookies/html/etat.html. 2004.

[31] E. Michlmayr and S. Cayzer. Learning user profiles from tagging data and leveraging them for personal(ized) information access. *WWW2007, Banff, Canada*, 2007.

[32] S. Alshawi Missi, Farouk and Guy Fitzgerald. Why crm efforts fail ? a study of the impact of data quality and data integration. *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS '05), Hawaii*, 2005.

[33] B. Mobasher. Web usage mining and personalization. *In Practical Handbook Of Internet Computing. Singh Munindar*, 2005.

[34] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 2001.

[35] I. Nonaka and H. Takeuchi. The knowledge-creating company. *The Knowledge-Creating Company*, 1995.

[36] B. Pernici and M. Scannapieco. Data quality in web information systems. *Journal on Data Semantics*, 2003.

[37] A. Pretschner and S. Gauch. Ontology based personalized search. *11th IEEE Intl. On Tools with Artificial Intelligence*, 1999.

[38] A. Pretschner and S. Gauch. Personalization on the web. *Technical report, In- formation and Telecommunication Technology Center, Department of Electrical Engineering and Computer Science, The University of Kansas*, 1999.

[39] L. Razmerita. Modèle utilisateur et modélisation utilisateur dans les systèmes de gestion des connaissances : une approche fondée sur les ontologies. 2003.

[40] T.C. Redman. Data quality for the information age. *Artech House*, 1996.

[41] E. Rich. User modeling via stereotypes. *Cognitive Science 3*, 1979.

[42] E. Rich. Users are individuals individualizing user models. *Int. J. Man-Machine Studies*, (18), 1983.

[43] D.M. String R.Y. Wang and L.M. Guarascio. An empirical investigation of data quality dimensions : a data consumer's perspective. *Total Data Quality Management (TDQM) Research Program, MIT Sloan School of Management*, 1993.

[44] V.C Storey R.Y. Wang and C.P. Firth. A framework for analysis of data quality. *Research IEEE Transaction on Knowledge and Data Engineering*, 7(4), 1995.

[45] S.A. Sarabjot. On the deployment of web usage mining. *Lecture notes in computer science*, 2004.

[46] J. Schmid. The main steps to data quality. 2003.

[47] E. Stoops. Data quality in user models : State of the art.

[48] E. Stoops and P. Thiran. Data update management for adaptive web-based systems. *PReCISE Research Center, University of Namur, Belgium.*

[49] S. Muthukrishnan T. Dasu, T. Johnson and V. Shkapenyuk. Mining database structure; or, how to build a data quality browser. *ACM SIGMOD*, June 2002.

[50] P. Thiran. Infom213 - aspects technologiques de l'e-business.

[51] W. Wahlster and A. Kobsa. Dialogue-based user models. *In Proceedings of IEEE*, 74(7), 1986.

[52] R.Y. Wang and D.M. Strong. Beyond accuracy : What data quality means to data consumers. *Journal of Management Information Systems*, 12(4), 1996.

# Annexe A

# Analyse interactionnelle : indicateurs de prédiction et de déduction de données concernant l'internaute

L'analyse interactionnelle ou comportementale est une technique visant à inférer un certain nombre d'informations au sujet de l'utilisateur sur base de ses comportements de navigation. Nous allons découvrir ci-dessous un panel (incomplet) d'indicateurs [11] pour la prédiction et la déduction de données concernant l'internaute. Les informations résultant de l'analyse de ces indicateurs seront pour la plupart des préférences ou centre d'intérêts plutôt que des données « brutes » telles que l'adresse ou encore le numéro de téléphone, qui sont elles obtenues de manière explicite (cf. Acquisition explicite de données). Ces indicateurs pourront s'obtenir (entre autres) via le langage JavaScript, qui, exécuté coté client, va pouvoir disposer d'une multitude de données interactionnelles (notamment les événements liés à la souris).

- Vote sondage
  - Interaction : L'internaute a parfois la possibilité de répondre à des sondages (figure A.1).
  - Déduction/Prédiction : L'avis de l'internaute sur le sujet.

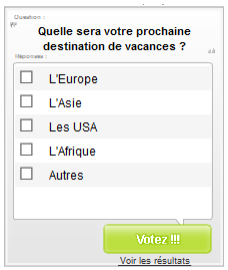



FIGURE A.1: Sondage web

- J'aime/Je n'aime pas
  - Interaction : L'internaute a parfois la possibilité de cliquer sur un bouton « J'aime » (figure A.2), « Like » ou « +1 » (ou son opposé « Je n'aime pas ») lié à un contenu.
  - Déduction/Prédiction : L'internaute aime (n'aime pas) le contenu.


FIGURE A.2: Bouton "J'aime"

- Evaluation
  - Interaction : L'internaute a parfois la possibilité d'évaluer un contenu (figure A.3).
  - Déduction/Prédiction : Ce que l'internaute pense du contenu évalué.



FIGURE A.3: Etoiles indiquant une évaluation

- Commentaires
  - Interaction : L'internaute a parfois la possibilité de commenter un contenu.
  - Déduction/Prédiction : Un système intelligent pourrait tenter d'analyser ce commentaire pour en retirer un certain nombre d'informations.
- Messages forums
  - Interaction : L'internaute a parfois la possibilité de poster des messages sur un forum.
  - Déduction/Prédiction : Un système intelligent pourrait tenter d'analyser ces messages pour en retirer un certain nombre d'informations. Parfois, les sujets des forums sont regroupés en

catégories. Il peut être intéressant d'analyser les statistiques de réponses ou création de sujets dans ces différentes catégories. Ainsi, on pourrait déduire qu'un utilisateur actif dans une section porte un certain intérêt pour ce domaine.

- Analyse souris
  - Interaction : L'internaute déplace, positionne la souris ou sélectionne du contenu.
  - Déduction/Prédiction : Intérêts pour le(s) contenu(s) sélectionné(s), intérêts pour le(s) contenu(s) sous le curseur, résultats d'analyse des déplacements de la souris.
- Copier/coller
  - Interaction : L'internaute copie du contenu dans presse-papier.
  - Déduction/Prédiction : Intérêts pour le(s) contenu(s) copié(s).
- Analyse du temps passé sur une page
  - Interaction : L'internaute navigue plus ou moins longtemps sur différentes pages.
  - Déduction/Prédiction : Le temps passé sur une page peut être chronométré. On peut penser que plus l'internaute aura passé du temps sur une page, plus il a tendance à préférer cette page ou son contenu.
- Nombre de visualisations d'une page ou d'un document
  - Interaction : L'internaute, lors de sa navigation, visualise un certain nombre de fois une page ou un document.
  - Déduction/Prédiction : On peut penser que plus une page ou un document aura été visualisé, plus l'internaute a tendance à avoir un intérêt pour le contenu de la page ou du document.
- Analyse du « clic » sur un contenu web
  - Interaction : L'internaute, lors de sa navigation, clique sur un contenu web (par exemple un lien).
  - Déduction/Prédiction : L'internaute porte un intérêt à ce contenu web.
- Impression d'une page ou d'un document
  - Interaction : L'internaute, lors de sa navigation, imprime une page ou un document.
  - Déduction/Prédiction : L'internaute a de l'intérêt pour cette page ou ce document.
- Sauvegarde d'un document
  - Interaction : L'internaute, lors de sa navigation, sauvegarde une page ou un document.
  - Déduction/Prédiction : L'internaute est intéressé par cette page ou ce document.

- Recherche
  - Interaction : L'internaute, effectue une recherche via un module de recherche présent sur le site.
  - Déduction/Prédiction : L'internaute est intéressé par les informations qu'il a recherchées.

  Notons que l'analyse comportementale peut combiner plusieurs de ces techniques de déduction afin d'obtenir d'autres informations, parfois plus riches. Prenons le cas où l'utilisateur clique sur un lien, ensuite, s'empresse de quitter la page qu'il vient de charger. En combinant les déductions de l'analyse des clics avec celles de l'analyse du temps passé sur les pages, on pourrait conclure que l'internaute ne trouve pas la page intéressante.

# Annexe B

# Méthodologie de construction de stéréotypes

Dans cette annexe nous proposons deux méthodologies de construction de stéréotypes. Elles vont être illustrées par l'exemple concernant le site d'appareils photographiques.

1. Quelle est votre public cible ?

   Répondre à cette première question va nous permettre d'établir un profil type (large) recherché sur lequel une réflexion sur des attributs va pouvoir être menée.

   **Réponse** Les amateurs de photographie.

   En s'attaquant directement au public cible on pourra dégager des caractéristiques importantes (pour le « business » du site), et directement élaguer celles de moindre importance.

2. Quelles en sont les caractéristiques (attributs) principales ?

   Répondre à cette seconde question va nous permettre d'établir une liste d'attributs importants pour le site. Ces attributs seront appelés « critères marginalisants ».

   **Réponse**

   – Manager, jardinier, paysagiste, architecte, artiste
   – 30-50 ans
   – Homme

   Il en ressort trois attributs : la profession, l'âge et le genre. Ceux-ci sont les critères marginalisants qui seront à la base de la division de la population des internautes en sous-populations.

3. Pour chacun de ces critères, nous allons à présent devoir étudier leurs valeurs. Il faudra s'assurer que le domaine de valeur ne soit pas trop large (en général, pas plus que quatre ou cinq valeurs

pour éviter une complexité trop importante). Pour cela, dans le cas ou le nombre de valeurs possibles excède cinq, il sera nécessaire de diviser le domaine de valeur. Autrement dit, pour un critère marginalisant, on va distinguer un certain nombre de groupement de valeurs, ceux-ci seront au maximum de cinq. Ces groupements devront être pertinents. C'est-à-dire que pour toute valeur présente dans un groupement, on considérera l'individu plus ou moins de la même manière. Il est important de couvrir tout le domaine, sinon, certains internautes pourraient ne rentrer dans aucun stéréotype.

Commençons avec le cas le plus simple : le genre. Le domaine de valeur du critère « Genre » comporte deux éléments : « Masculin » ou « Féminin ». Le domaine est fermé et tout le domaine est couvert.

En ce qui concerne la profession, le domaine n'est pas fermé car de nouvelles professions peuvent naître chaque jour. Il serait donc intéressant de définir un ensemble de catégories de professions de manière à ce que toutes professions puissent être classées dans une catégorie (afin de couvrir tout le domaine). Par exemple :

- Management : emploi lié à la gestion
- Art et Nature : emploi lié à l'art et/ou à la nature (par exemple jardinier ou paysagiste)
- Autre : autres emplois (de manière à fermer le domaine et à le couvrir dans son entièreté)

Enfin, pour l'âge, le domaine s'étend de 0 à l'infini. Cependant, un être humain n'est a priori pas encore immortel ce qui nous permet de réduire le domaine à l'intervalle [1,150]. Rappelons que nous voulons autant que possible réduire le nombre de valeurs pour un critère. L'attribut « Age » en comporte 150 ! Il est donc judicieux de décomposer le domaine en sous-intervalles pertinents. Par exemple :

- $<$25 ans
- De 26 à 40 ans
- De 41 à 65 ans
- $>$ 65 ans

4. Effectuer le produit cartésien des différents domaines obtenus de (3). Nous obtenons 24 [1] stéréotypes :

1. 4 intervalles d'âge * 3 catégories de profession * 2 genres = 24

| Num | Age | Profession | Genre |
|---|---|---|---|
| 1 | <20 | Management | Féminin |
| 2 | <20 | Management | Masculin |
| 3 | <20 | Art et Nature | Féminin |
| 4 | <20 | Art et Nature | Masculin |
| 5 | <20 | Autres | Féminin |
| 6 | <20 | Autres | Masculin |
| 7 | 21-40 | Management | Féminin |
| 8 | 21-40 | Management | Masculin |
| 9 | 21-40 | Art et Nature | Féminin |
| 10 | 21-40 | Art et Nature | Masculin |
| 11 | 21-40 | Autres | Féminin |
| 12 | 21-40 | Autres | Masculin |
| 13 | 41-65 | Management | Féminin |
| 14 | 41-65 | Management | Masculin |
| 15 | 41-65 | Art et Nature | Féminin |
| 16 | 41-65 | Art et Nature | Masculin |
| 17 | 41-65 | Autres | Féminin |
| 18 | 41-65 | Autres | Masculin |
| 19 | >65 | Management | Féminin |
| 20 | >65 | Management | Masculin |
| 21 | >65 | Art et Nature | Féminin |
| 22 | >65 | Art et Nature | Masculin |
| 23 | >65 | Autres | Féminin |
| 24 | >65 | Autres | Masculin |

FIGURE B.1: Tableau reprenant la liste de stéréotypes obtenus en effectuant le produit cartésien des différentes valeurs de chaque paramètre

5. Effectuer une simplification en élaguant les stéréotypes ne constituant pas un réel intérêt en soit. Par exemple, les stéréotypes de 1 à 6 peuvent être regroupés en un unique stéréotype. Peu importe leur profession ou leur genre, il y a beaucoup de chance qu'il ne soit pas (encore) intéressé par la photographie de par leur jeune âge. Nous ferons pareil pour ceux numérotés de 19 à 24. Le gestionnaire du site pense que son public cible est masculin (voir ci-dessus), un stéréotype spécifique féminin peut dès lors être créé, où la profession et l'âge importent peu. Ceci permet de supprimer toutes les lignes de genre « Féminin ». Il reste neuf lignes, celles-ci constituent une bonne base pour l'exploitation qui sera faite par après. Notons que le gestionnaire pourrait encore regrouper quelques lignes s'il désire se restreindre à un ensemble de moindre taille.

| Num | Age | Profession | Genre |
|---|---|---|---|
| 1 | <20 | - | - |
| 2 | 21-40 | Management | Masculin |
| 3 | 21-40 | Art et Nature | Masculin |
| 4 | 21-40 | Autres | Masculin |
| 5 | 41-65 | Management | Masculin |
| 6 | 41-65 | Art et Nature | Masculin |
| 7 | 41-65 | Autres | Masculin |
| 8 | - | - | Féminin |
| 9 | >65 | - | - |

FIGURE B.2: Tableau simplifié reprenant la liste de stéréotypes

Idéalement, toutes dépendances entre critères marginalisants devraient être évitées. Dans cet exemple, la profession est partiellement dépendante de l'âge. En effet, tout individu ne possédant pas l'âge requis pour travailler ne devrait (en principe) pas avoir de profession. Derrière de telles dépendances se cachent souvent de potentielles simplifications. Le problème est qu'elles ne sont pas simplement détectables...

6. Que peut-on déduire de pertinent de chaque stéréotype ?

   Répondre à cette question permettra de définir les attributs de la partie de prédiction du stéréotype. Nous entendons par pertinent les informations qu'il serait intéressant de déduire d'un stéréotype pour le « business » du système. Ces informations doivent apporter une plus-value au système.

   **Réponse** Le niveau en photographie (ou plus concrètement : « intéressé par un appareil photo de niveau : »)

| Num | Age | Profession | Genre | Prédiction |
|---|---|---|---|---|
| 1 | <20 | - | - | Débutant |
| 2 | 21-40 | Management | Masculin | Amateur |
| 3 | 21-40 | Art et Nature | Masculin | Amateur |
| 4 | 21-40 | Autres | Masculin | Débutant |
| 5 | 41-65 | Management | Masculin | Professionnel |
| 6 | 41-65 | Art et Nature | Masculin | Professionnel |
| 7 | 41-65 | Autres | Masculin | Amateur |
| 8 | - | - | Féminin | Débutant |
| 9 | >65 | - | - | Débutant |

FIGURE B.3: Tableau simplifié reprenant la liste de stéréotypes avec prédiction

Pour chaque stéréotype, le gestionnaire a défini une valeur de prédiction concernant le niveau en photographie.

7. La dernière étape est celle de transformation. Nous voulons obtenir un stéréotype avec une représentation semblable à la figure 2.19. Pour cela, il suffit, pour chaque ligne de la Figure B.3, de créer un stéréotype où, pour chaque attribut, toutes les valeurs soient présentes et associées à une probabilité. Il n'y a pas vraiment de méthode miracle pour déterminer les probabilités. Néanmoins, les informations de la figure B.3 doivent apparaitre. La figure B.4 exprime le cas de la ligne 5.

| | | | |
|---|---|---|---|
| **Nom:** | Photographe professionnel | | |
| **Profil:** | | | |
| | Age | *<20* | **0** |
| | | *20-40* | **0,15** |
| | | *41-65* | **0,8** |
| | | *>65* | **0,05** |
| | Genre | *Homme* | **0,9** |
| | | *Femme* | **0,1** |
| | Job | *Manager* | **0,8** |
| | | *Art & Nature* | **0,1** |
| | | *Autre* | **0,1** |
| **Prédictions/intérêts** | | | |
| | Niveau en Photographie: | *Professionnel* | **0,8** |
| | | *Amateur* | **0,6** |
| | | *Débutant* | **0,2** |

FIGURE B.4: Stéréotype correspondant à la ligne 5 du tableau reprennant la liste des stéréotypes

Ici, nous avons conçu les stéréotypes sur base des données « profil » (partie descriptive). Nous avons donc utilisé une conception « orientée profil ». Une autre approche peut également être utilisée en se basant sur la partie prédictive. Cette conception est « orientée prédiction ». Les attributs et valeurs de prédictions sont d'abord recherchés. Pour chaque valeur de chaque attribut de prédiction, une recherche de profil doit être effectuée. Ainsi on aurait d'abord obtenu que le niveau en photo soit une prédiction et possède les valeurs « professionnel », « amateur » et « débutant ». Ensuite, pour chacune de ces valeurs, on aurait bâti un stéréotype. Prenons par exemple la valeur « Professionnel ». Un stéréotype nommé « Photographe Professionnel » aurait été créé. Les probabilités pour chaque valeur de chaque attribut serait alors complétées de manière à respecter le fait que le niveau professionnel est prépondérant et de manière à faire ressortir le profil d'un professionnel. Typiquement, la figure B.5 décrit un photographe professionnel comme

étant âgé entre 40 et 65 ans (à 55%), masculin (à 80%) et possédant un emploi lié à l'art ou la nature (à 45%).

| | | | |
|---|---|---|---|
| **Nom:** | Photographe professionnel | | |
| **Profil:** | | | |
| | Age | *<20* | **0,05** |
| | | *20-40* | **0,35** |
| | | *41-65* | **0,55** |
| | | *>65* | **0,05** |
| | Genre | *Homme* | **0,8** |
| | | *Femme* | **0,2** |
| | Job | *Manager* | **0,4** |
| | | *Art & Nature* | **0,45** |
| | | *Autre* | **0,15** |
| **Prédictions/intérêts** | | | |
| | Niveau en Photographie: | *Professionnel* | **0,8** |
| | | *Amateur* | **0,6** |
| | | *Débutant* | **0,2** |

FIGURE B.5: Stéréotype orienté prédiction

Alors que la première façon de procéder dégageait neuf stéréotypes, celle-ci en dégage seulement trois. De plus ces stéréotypes sont plus orientés « business » que « clichés démographiques ». Ce qui, dans la majorité des cas, est plus optimal.

# Annexe C

# Algorithme de calcul de l'indice de confiance

Cette annexe présente l'algorithme de calcul de l'indice de confiance implémenté dans le prototype du chapitre 6.

L'algorithme C.1 est la méthode englobante permettant d'initialiser les variables de bases, de construire le tableau orienté valeur, d'itérer sur ce dernier en calculant à chaque itération (1) un certain nombre de variables intermédiaires nécessaires au calcul de la correctness, (2) la correctness de la valeur d'attribut, (3) la freshness de la valeur d'attribut, (4) l'indice de confiance de la valeur d'attribut, et de construire un tableau qui contiendra les indices de confiance de chaque valeur d'attribut.

**Algorithm C.1** Méthode globale permettant l'évaluation de l'indice de confiance pour chaque valeur d'attribut

```
public static Float[] processQuality(Attribute attribute) {
    //Building of value oriented array
    Object[][] tabValuesO = constructValuesOArray(attribute);

    //Variables initialization
    float correctness_j = 0.0f; //to store correctness of attribute value j
    float freshness_j = 0.0f; //to store freshness of attribute value j
    float confidenceIndex_j = 0.0f; //to store confidence index of attribute value j
    Float[] results = new Float[tabValuesO.length]; //to store confidence index for each attribute value

    //Initialize intermediate variable partWeight
    float partWeight = 0.0f; //to store the part of sWeight for a value j on the sWeightTot

    //Compute Correctness, Freshness and Confidence Index
    //loop on value oriented array (j)
    for (int j = 0; j<tabValuesO.length; j++){

        //compute intermediate variables
        Float [] intermediateVariables = computeIntermediateVariables(attribute, (String)tabValuesO[j][0]);
        //compute correctness_i
        correctness_j = computeCorrectness((Float) getMaxReliability(attribute.getValues().get(j)),
                intermediateVariables[1], intermediateVariables[2], partWeight, intermediateVariables[0]);
        //compute freshness_i
        freshness_j = computeFreshness(((Date)tabValuesO[j][2]).getTime(),(float)attribute.getLifetimeMillis(),
                attribute.getDepFct());
        //compute confidenceIndex_j
        confidenceIndex_j = computeConfidenceIndex(freshness_j, correctness_j);

        //put the confidence index in results array
        results[j] = floor(confidenceIndex_j, 4);
    }

    return results;
}
```

L'algorithme C.2 permet de calculer la valeur des variables "sMatch", "sWeight" et "sWeightTot" nécessaire à l'algorithme C.3 de calcul de la correctness. Ces valeurs sont stockées dans un tableau d'entiers. Le rôle de ces variables dans le calcul de l'indice de correctness a été expliqué à la section 5.2.1.3.

**Algorithm C.2** Méthode permettant le calcul de variables intermédiaires (liées à une valeur d'attribut) nécessaires à l'évaluation de la correctness (algorithmeC.3)

```
private static Float[] computeIntermediateVariables(Attribute attribute, String value) {
    //Building of source oriented array
    Object[] tabSourcesO = constructSourcesOArray(attribute);

    //initialize intermediate variables
    float match = 0.0f; //to store from the reliability of source i with respect to maximum reliability
    //of sources i for a given value
    float matchOcc = 0.0f; //to store the multiplication of match and the number of occurrence of a source i
    float sMatch = 0.0f; //to store the sum of the matchOcc of all sources i if the value of the source i
    //match with the value of value j
    float weight = 0.0f; //to store the weight of a source i (reliability of a source i * number of
    //occurrence of a source i)
    float sWeightTot = 0.0f; //to store the sum of the weights of all sources i
    float sWeight = 0.0f; //to store the sum of the weights of all sources i if the value of the source i
    //match with the value of value j

    //loop on source oriented array (i)
    for (int i = 0 ; i<tabSourcesO.length;i++){
        //set matchOcc variable to 0
        matchOcc = 0.0f;
        //compute weight of a source i
        weight = (((Integer)((Object[])tabSourcesO[i])[2])*((Float)((Object[])tabSourcesO[i])[1]));

        //test if the value of value j equals the value of source i
        if (value.equals(((Object[])tabSourcesO[i])[4])) {
            //compute match of a source i
            match = (((Float)((Object[])tabSourcesO[i])[1]) / ((Float)((Float)getMaxReliability(tabSourcesO,
                    (String)((Object[])tabSourcesO[i])[4])))) ;
            //compute matchOcc of a source i
            matchOcc = match * ((Integer)((Object[])tabSourcesO[i])[2]) ;
            //compute sMatch
            sMatch = sMatch + matchOcc;
            //compute sWeight
            sWeight = sWeight + weight;
        }
        //compute sWeightTot
        sWeightTot = sWeightTot + weight;
    }

    return new Float[]{sMatch,sWeight,sWeightTot};
}
```

L'algorithme C.3 permet de calculer la correctness.

**Algorithm C.3** Méthode permettant le calcul de la correctness d'une valeur d'attribut

```
private static float computeCorrectness(Float maxReliability, float sWeight, float sWeightTot, float partWeight,
        float sMatch) {

    //test if sMatch equals 0
    if (sMatch != 0.0f) {
        //compute partWeight
        partWeight = (float)((float)sWeight / (float)sWeightTot);
        //compute correctness_j
        return (float) ((Math.pow((double)maxReliability,(double)(1 / (double)sMatch) )) * partWeight);
    } else {
        //set correctness_j to 0
        return 0.0f;
    }
}
```

L'algorithme C.4 se charge de calculer la freshness sur base de la date de la dernière mise à jour d'une valeur d'attribut, de la durée de vie d'une valeur d'attribut et de sa fonction de dépréciation. Notons que trois fonctions de dépréciation temporelle ont été implémentées : les fonctions linéaire,

concave et convexe (telles que définies à la section 1.2.2.1 "Fonction de dépréciation").

**Algorithm C.4** Méthode permettant le calcul de la freshness d'une valeur d'attribut

```
public static float computeFreshness(long lastDateMillis, float lifetimeMillis, int depFct) {
    //initializations
    //initialize freshness variable to 0;
    float freshness = 0.0f;
    //initialize now variable to current time in milliseconds
    long now = System.currentTimeMillis();
    //initialize elapsedTime to the difference between lastDateMillis parameter and now variable
    long elapsedTime = now - lastDateMillis;
    //initialize progress to the progress between elapsedTime and lifetime
    float progress = (float)((float)elapsedTime / (float) lifetimeMillis);

    //test if progress is between 0 and 1
    if (progress>=0 && progress <1){
        //test which depreciation function is chosen for the attribute
        if (depFct == 0){
            //compute freshness for a linear depreciation function
            freshness = 1 - progress;
        } else if (depFct == 1) {
            //compute freshness for a concave depreciation function
            freshness = 1 - ((float) (Math.pow((double)progress,(double)3 )));
        } else if (depFct == 2) {
            //compute freshness for a convex depreciation function
            freshness = 1 - ((float) (Math.pow((double)progress,(double)(1 / (double)3 ) )));
        } else {
            //depreciation function unknown
            System.out.println("-Depreciation function unkown");
            return -1;
        }
    } else {
        //set freshness_j to 0
        freshness = 0;
    }

    //set freshness to 0 where freshness_j is negative
    if (freshness < 0){
        freshness = 0;
    }
    return freshness;
}
```

L'algorithme C.5 évalue l'indice de confiance sur base de la freshness et la correctness, précédemment calculées. Comme nous l'avons vu, cet indice s'obtient en multipliant la correctness par la freshness.

**Algorithm C.5** Méthode permettant le calcul de l'indice de confiance d'une valeur d'attribut

```
private static float computeConfidenceIndex(float freshness, float correctness) {

    //initialize confidenceIndex to the result of the multiplication of freshness and correctness
    float confidenceIndex = freshness * correctness ;

    //set confidenceIndex_j to 0 where confidenceIndex_j is negative
    if (confidenceIndex < 0) {
        confidenceIndex = 0;
    }
    return confidenceIndex;
}
```

L'algorithme C.6 permet de sélectionner la valeur active sur base d'un ensemble de valeur d'attribut.

**Algorithm C.6** Méthode permettant la sélection de la valeur active d'un attribut

```
/**
 * @param tab[0] = date Last Update value ; tab[1] = confidence Index
 * @return index of the line of tab which contains the active value
 */
public static int selectActiveValue(Object [][] tab){

    //initialize variables
    float bestConfidenceIndex = 0.0f;
    int active = -1;
    long valueLastUpdateTimestamp = 0;
    int valueLastUpdateValue = -1;

    //loop on tab
    for (int j = 0 ; j < tab.length ; j ++) {
        //test if confidence index of tab[j] is greater than all precedent greater confidence index
        if ((Float)tab[j][1] > bestConfidenceIndex) { //the value of tab[j] become the active value
            bestConfidenceIndex = (Float)tab[j][1];
            active = j;
            valueLastUpdateValue = j;
            valueLastUpdateTimestamp = ((Date)tab[j][0]).getTime();
        } else if ((Float)tab[j][1] == bestConfidenceIndex) {
            //else if the confidence index of tab[j] equals the greater precedent confidence index

            //test if the date of last update of tab[j] is more recent than the other which has
            //the precedent best confidence index
            if (((Date)tab[j][0]).getTime()>valueLastUpdateTimestamp){ //the value of tab[j] become the active value
                active = j;
            }
        }
    }

    //test if the best confidence index equals 0
    if (bestConfidenceIndex==0){ //the value which is more recently update become the active value
        active = valueLastUpdateValue;
    }

    return active;
}
```