



HAL
open science

Approches de modélisation et d'optimisation pour la conception d'un système interactif d'aide au déplacement dans un hypermarché

Ismahène Hadj Khalifa

► **To cite this version:**

Ismahène Hadj Khalifa. Approches de modélisation et d'optimisation pour la conception d'un système interactif d'aide au déplacement dans un hypermarché. Autre. Ecole Centrale de Lille, 2011. Français. NNT : 2011ECLI0008 . tel-00605118

HAL Id: tel-00605118

<https://theses.hal.science/tel-00605118>

Submitted on 30 Jun 2011

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

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

THÈSE

Présentée en vue d'obtenir le grade de

DOCTEUR

Discipline : Automatique et Informatique Industrielle

Par

Ismahène HADJ KHALIFA

Maîtrise en Hautes Etudes Commerciales (IHEC Carthage)

Master Systèmes d'Information et d'Ingénierie de la Conception (EC Lille)

Approches de modélisation et d'optimisation pour la conception d'un système interactif d'aide au déplacement dans un hypermarché

Soutenue publiquement le 16 juin 2011 devant le jury :

M. Pierre BORNE	Ecole Centrale de Lille	Président
M. Jean-Pierre CAMPAGNE	INSA de Lyon	Rapporteur
M. Gaston LEFRANC	Pontificia Universidad Catolica de Valparaiso, Chili	Rapporteur
M. Khaled MELLOULI	Institut des Hautes Etudes Commerciales de Carthage, Tunisie	Rapporteur
M. Abdelhakim ARTIBA	Université de Valenciennes et du Hainaut-Cambrésis	Examineur
M. Bernard BARFETY	Leroy Merlin	Invité
M. Abdelkader EL KAMEL	Ecole Centrale de Lille	Directeur de thèse
M. Sami FAIZ	Faculté des Science Juridiques, Economiques et de Gestion de Jendouba, Tunisie	Examineur

A Baba Fadhil et Hanéna

A mon adorable mère et mon cher père

A Mehdi, mon cher mari, et à mes beaux parents

A tous ceux qui me sont chers

Avant-propos

Les travaux présentés dans cette thèse ont été réalisés au sein du Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) de l'Ecole Centrale de Lille et s'inscrivent dans le cadre des activités du Pôle de Compétitivité des Industries du Commerce (PICOM). Cette thèse a été labellisée par le PICOM et a ainsi bénéficié d'une bourse cofinancée Leroy Merlin/Région Nord-Pas-de-Calais. Je tiens à remercier chaleureusement Monsieur Bernard Barfety, Directeur Organisation et Systèmes d'information de Leroy Merlin et Monsieur Didier Lieven, Coordonnateur du PICOM pour la confiance qu'ils m'ont accordée en finançant mes trois années de thèse.

C'est un agréable devoir pour moi d'exprimer ma profonde gratitude à mon directeur de thèse Monsieur Abdelkader EL KAMEL. Je tiens à le remercier et lui manifester ma sincère reconnaissance pour ses grandes qualités humaines, son encadrement plein d'enthousiasme et de rigueur et pour la confiance dont il a fait preuve à mon égard.

Je suis particulièrement sensible au grand honneur que Monsieur Pierre BORNE me fait en acceptant de présider mon Jury de thèse. Qu'il trouve ici l'expression de ma profonde reconnaissance.

Je tiens à exprimer ma vive gratitude à Monsieur Khaled MELLOULI, Monsieur Jean-Pierre CAMPAGNE et Monsieur Gaston LEFRANC pour l'intérêt qu'ils ont bien voulu porter à mes travaux en acceptant d'être les rapporteurs de cette thèse. Qu'ils trouvent ici le témoignage de ma profonde reconnaissance.

Je souhaite également témoigner de mes vifs remerciements à Monsieur Sami FAIZ et Monsieur Abdelhakim ARTIBA d'avoir accepté de participer à mon Jury d'Examen.

Je tiens à remercier mes collègues de l'équipe dirigée par Monsieur EL KAMEL et en particulier Jin Zhao, Minzhi Luo, Stanislas Legendre, Harald Pillu ainsi que tous mes

collègues du LAGIS et tout le personnel de l'Ecole Centrale de Lille pour leur bonne humeur et leur disponibilité.

Un témoignage de ma profonde reconnaissance s'adresse à toute ma famille et mes amis : mes parents, mon frère et ma sœur, mes grands-parents, mes tantes et oncles, mes cousins et cousines, ma belle famille, mes amis en Tunisie et en France qui ont toujours cru en moi et encouragé pour aller jusqu'au bout. Un MERCI à mon mari pour sa patience, ses sacrifices et ses encouragements.

RESUME

Les travaux présentés dans cette thèse ont porté sur l'étude de faisabilité technique et logicielle du système i-GUIDE, système interactif de guidage des personnes dans les hypermarchés. Nous avons détaillé l'analyse fonctionnelle du besoin du système. Ensuite, nous avons étudié l'impact de l'intégration du système dans le magasin à travers le diagramme BPMN. Nous avons opté pour l'approche UML pour décrire les principales fonctionnalités de notre système ainsi que les objets nécessaires pour son bon fonctionnement. Une architecture du système i-GUIDE, basée sur la technologie RFID avec une application sous Android, a été présentée. Par ailleurs, nous avons proposé des approches d'optimisation de parcours dans un hypermarché basées sur la méthode de recherche tabou pour deux problèmes. Pour le premier problème, nous avons choisi le critère de la plus courte distance pour la détermination du chemin et pour le deuxième nous avons ajouté une contrainte de temps pour des articles en promotion. Avant de chercher le chemin le plus court à parcourir pour trouver les articles existants dans la liste de courses, nous avons proposé une méthode pour la détermination des distances entre les articles de l'hypermarché pris deux à deux.

Mots clés : Système interactif de guidage, Intelligence ambiante, BPMN, UML, Technologies de localisation, RFID, Android, Optimisation de parcours avec et sans fenêtres de temps, Heuristique de plus proche voisin adaptée, Recherche tabou avec ensembles.

ABSTRACT

The present work focuses on the technical feasibility study of i-GUIDE system which is a real time indoor navigation system dedicated to assist persons inside hypermarkets. We detailed its functional analysis. Then, we studied the impact of integrating the system inside hypermarkets. We opted for an UML design to describe its main functionalities and objects required. We presented architecture of i-GUIDE system based on RFID technology with an Android application. Furthermore, we introduced optimization approaches based on tabu search to compute the route visiting items existing in a shopping list for two problems. The first one treats the shortest path to pick up items and the second one adds a time constraint for promotional items. Before computing the shortest path, we introduced a method to determine distance between each two items existing in the hypermarket.

Keywords: Intelligent Navigation system, Ambient intelligence, BPMN, UML, Localization technologies, RFID, Android, itinerary optimization with and without time constraint, Adapted Nearest Neighbour Heuristic, Tabu Search with sets.

LISTE DES FIGURES

Figure 1 : Fonctionnement du système BlueEyes [Mab09]	24
Figure 2 : Le Robocart [Kul07]	26
Figure 3 : Schéma de fonctionnement du dispositif [RAS09]	27
Figure 4 : Utilisation du mobile NTT DoCoMo [Pou07]	28
Figure 5 : Plan du magasin avec le chemin à parcourir [Pue10]	32
Figure 6 : La " bête à cornes" du système global	33
Figure 7 : Diagramme pieuvre du système global	34
Figure 8 : Exemple de grille GRAI [Lat06]	45
Figure 9 : Les diagrammes dans UML 2.0 [OMG05]	46
Figure 10 : Extrait du diagramme BPMN	52
Figure 11 : Modélisation BPMN de la chaîne logistique d'achat et de vente de produits	53
Figure 12 : Diagramme de Cas d'utilisation UML de notre système de guidage [HEB10a]	56
Figure 13 : Diagramme d'activité UML de la fonction « Guider l'utilisateur et lui donner des instructions » [HEB10a]	57
Figure 14 : Diagramme de séquence de la fonction calcul du chemin le plus court [HEB10a]	59
Figure 15 : Architecture globale du système de navigation temps réel	67
Figure 16 : Prototype de la solution	68
Figure 17 : Disposition des repères sur les gondoles	68
Figure 18 : Comparaison entre les parts de marché des ventes de Smartphones par systèmes d'exploitation en 2009 et 2010 [Alt11]	71
Figure 19 : Architecture du système de guidage avec un calculateur et une BDD embarqués	72
Figure 20 : Architecture du système de guidage avec un calculateur et une base de données externes	73
Figure 21: Exemple de bloc : Gestion de la connexion Bluetooth	74
Figure 22 : Ecran d'accueil pour Android	74
Figure 23 : Architecture de l'application sous Android en cours	75

Figure 24 : Présentation du problème d'optimisation de l'itinéraire dans un hypermarché	85
Figure 25 : Base de données d'un magasin.....	88
Figure 26: Division d'un magasin Leroy Merlin en 6 zones.....	89
Figure 27 : Attribution de repères sur les gondoles.....	90
Figure 28 : Extrait de l'algorithme de calcul des distances.....	92
Figure 29 : Détermination des distances entre deux articles appartenant à deux zones différentes	93
Figure 30 : Algorithme de Branch and Bound	96
Figure 31 : Exemple de l'algorithme de Christofides	97
Figure 32 : Amélioration de la solution avec l'algorithme PPVA [HEB10b].....	98
Figure 33 : Mouvement d'échange entre deux éléments successifs	101
Figure 34 : Mouvement d'échange entre l'élément i et l'élément N-i-1	101
Figure 35 : Amélioration de la solution Tabou après définition des ensembles.....	102
Figure 36 : Algorithme de recherche Tabou adapté [HKB11a]	103
Figure 37 : Comparaison entre la solution tabou et la solution du PPVA pour P1	106
Figure 38 : Comparaison entre la solution tabou et la solution du PPVA pour P2	106
Figure 39 : Convergence de la solution pour P1	107
Figure 40 : Convergence de la solution pour P2	107
Figure 41 : Différentes dates de l'article i en promotion avec fenêtre de temps.....	110
Figure 42: Algorithme d'insertion adapté au POPHFT.....	114
Figure 43 : Algorithme de recherche tabou adapté au POPHFT	115
Figure 44 : Amélioration apportée par la méthode de recherche tabou adaptée	117
Figure 45 : Modélisation UML de la dynamique applicative du système en cas de modification de la liste de courses	117
Figure 46 : Résolution du problème en cas de modification de la liste de courses	118
Figure 47 : Algorithme d'insertion d'un article ajouté à la solution de RTA.....	119
Figure 48 : Modélisation UML de la dynamique applicative du système en cas de changement de direction.....	120

LISTE DES TABLEAUX

Tableau 1 : Diagrammes UML 2.0 [OMG05].....	47
Tableau 2 : Différents types de Flow Objects	49
Tableau 3 : Différents types de Connecting Objects.....	50
Tableau 4 : Différents types de Swimlanes	50
Tableau 5 : Différents types d'Artifacts	51
Tableau 6 : Caractéristiques des trois grandes familles du problème de transport.....	82
Tableau 7 : Analogie entre le problème du voyageur de commerce et notre problème	86
Tableau 8 : Résultats de la méthode de recherche Tabou et la méthode du PPV pour P1	105
Tableau 9 : Résultats de la méthode de recherche Tabou et la méthode du PPV pour P2	105
Tableau 10 : Tableau récapitulatif des tenures et du nombre des itérations pour P1	108
Tableau 11 : Tableau récapitulatif des tenures et du nombre des itérations pour P2	108
Tableau 12 : Caractéristiques des articles en promotion avec fenêtre de temps	115
Tableau 13: Tableau récapitulatif des résultats de l'AIA et de l'ARTA	116

TABLE DES MATIERES

INTRODUCTION GENERALE.....	17
CHAPITRE 1 : CONTEXTE GENERAL DE L'ETUDE	21
1.1 Introduction et problématique de l'étude	21
1.2 Etude d'antériorité.....	22
1.2.1 Systèmes d'aide au déplacement des personnes avec handicap visuel	22
1.2.1.1 Le système Audioplage.....	22
1.2.1.2 Le système BlueEyes.....	23
1.2.1.3 Le système SmartEyes.....	25
1.2.1.4 Le système Wayfinder Access.....	25
1.2.1.5 Le système RoboCart.....	25
1.2.1.6 Le Réseau SESAMONET.....	26
1.2.1.7 Le système de NTT DoCoMo.....	27
1.2.2 Systèmes d'aide aux personnes à mobilité réduite	28
1.2.2.1 Le service Handi-Cité.....	29
1.2.2.2 Le système Afmaster Robotique.....	29
1.2.3 Systèmes d'aide au déplacement des personnes	29
1.2.3.1 Le système MediaCart.....	29
1.2.3.2 Le système MIGS	30
1.2.3.3 Le système Keyshopping.....	31
1.3 Analyse fonctionnelle du besoin/ Cahier des charges du système i-GUIDE	32
1.3.1 La « bête à cornes »	33
1.3.2 Le diagramme pieuvre	33
1.3.3 Le diagramme FAST	35
1.3.3.1 Personnes avec handicap visuel.....	35
1.3.3.2 Personnes à mobilité réduite.....	38

1.3.3.3	Système global.....	39
1.4	Conclusion.....	42
CHAPITRE 2 : CONCEPTION DU SYSTEME I-GUIDE.....		43
2.1	Introduction.....	43
2.2	Modélisation de la chaîne logistique de vente/achat des produits dans le magasin.....	43
2.2.1	Etat de l’art sur les méthodes de modélisation des processus de l’entreprise.....	43
2.2.2	Modélisation avec GRAI-GIM.....	44
2.2.2.1	Modélisation avec UML.....	45
2.2.2.2	Modélisation avec BPMN.....	48
2.2.3	Cas Leroy Merlin.....	51
2.2.3.1	Analyse de l’existant.....	51
2.2.3.2	Modélisation du processus d’achat et de vente des produits.....	52
2.2.3.3	Avantages de la modélisation BPMN.....	54
2.3	Analyse des exigences du système i-GUIDE.....	54
2.3.1	Diagramme de Cas d’utilisation UML.....	55
2.3.2	Diagramme d’activités UML.....	56
2.3.3	Diagramme de séquences UML.....	58
2.3.4	Avantages du système i-GUIDE.....	59
2.3.4.1	Avantages pour le client.....	59
2.3.4.2	Avantages pour le magasin.....	60
2.4	Conclusion.....	61
CHAPITRE 3 : TECHNOLOGIES DE LOCALISATION DANS UN ESPACE FERME.....		63
3.1	Introduction.....	63
3.2	Technologies de localisation dans un espace fermé.....	63
3.2.1	La technologie Bluetooth.....	63
3.2.2	La technologie RFID.....	64

3.2.3	La technologie WiFi	65
3.2.4	Le système de GPS indoor.....	65
3.2.5	La technologie NFC.....	66
3.3.	Proposition d'un système de localisation dans un hypermarché	66
3.3.1	Présentation de l'architecture du système i-GUIDE	66
3.3.2	Présentation du prototype réalisé.....	68
3.4.	Présentation de l'application mobile	70
3.4.1	Présentation du système d'exploitation Android	70
3.4.2	Application sous Android avec calculateur et base de données embarqués	72
3.4.3	Application sous Android calculateur base de données externes	73
3.4.4	Application sous Android	75
3.5.	Conclusion.....	76
CHAPITRE 4 : APPROCHES D'OPTIMISATION DE PARCOURS EN TEMPS REEL DANS UN HYPERMARCHÉ.....		79
4.1.	Introduction	79
4.2.	Définitions générales.....	79
4.2.1	Optimisation combinatoire et théorie de complexité	79
4.2.2	Définitions utiles.....	80
4.3.	Etat de l'art sur les méthodes d'optimisation de parcours.....	81
4.3.1	Méthodes d'optimisation liées aux problèmes de tournées	81
4.3.2	Méthodes de résolution des problèmes de guidage des utilisateurs par GPS	84
4.4.	Position du problème.....	85
4.5.	Méthodologie d'optimisation de parcours dans les hypermarchés.....	87
4.6.	Création de la base de données.....	87
4.7.	Détermination des distances entre les articles deux à deux.....	89

4.7.1	Détermination des distances entre deux articles appartenant à la même zone.....	91
4.7.2	Détermination des distances entre deux articles appartenant à deux zones différentes.....	92
4.8.	Approches d'optimisation de parcours dans un hypermarché.....	94
4.8.1	Problème d'optimisation de parcours dans un hypermarché (POPH)	94
4.8.1.1	Formulation mathématique du problème.....	94
4.8.1.2	Proposition des méthodes de résolution du problème d'optimisation	96
4.8.1.2.1	La méthode de Branch and Bound.....	96
4.8.1.2.2	La méthode de Christofides	97
4.8.1.2.3	La méthode du Plus Proche Voisin	98
4.8.1.2.4	La méthode du Plus Proche Voisin Adaptée.....	98
4.8.1.2.5	La méthode de Recherche Tabou.....	99
4.8.1.3	Etude expérimentale et résultats	103
4.8.2	Problème d'optimisation de parcours dans un hypermarché pour des articles en promotion avec fenêtre de temps (POPHFT).....	109
4.8.2.1	Détermination des fenêtres de temps	109
4.8.2.2	Formulation mathématique du problème :.....	111
4.8.2.3	Proposition d'une approche de résolution du problème	113
4.8.2.3.1	La méthode d'insertion	113
4.8.2.3.2	La méthode de Recherche Tabou adaptée au POPHFT	114
4.8.2.4	Etude expérimentale et résultats	115
4.9.	Gestion de la partie dynamique de guidage.....	117
4.9.1	Cas de modification de la liste de courses	117
4.9.2	Cas de changement de direction	119
4.10.	Conclusion.....	121
	CONCLUSIONS ET PERSPECTIVES	123
	BIBLIOGRAPHIE	127

INTRODUCTION GENERALE

Faire des courses dans les grandes surfaces représente un fardeau pour certaines personnes car passer des heures et des heures à chercher des articles semble fatigant et ennuyeux notamment si c'est la première visite à un hypermarché. Pour cette raison, certaines personnes préfèrent aller chez l'épicerie ou le petit magasin du coin. De plus, avec l'apparition des nouveaux services de vente en ligne même pour les produits agroalimentaires, le client peut commander des articles en quelques clics, chez lui, depuis son ordinateur, et ses achats lui seront livrés à domicile. Après le commerce électronique (e-commerce), le commerce mobile (m-mobile) est la nouvelle tendance qui représente un nouveau canal de distribution complémentaire aux autres. En effet, les applications sur les téléphones mobiles offrent aux entreprises la possibilité d'être présentes dans le quotidien du consommateur. Grâce aux technologies sans contacts telles que le Bluetooth et le WiFi permettant d'établir une connexion avec l'appareil mobile sur une large portée ou encore le NFC (Near Field Communication) et le code 2D permettant un échange de contenu, on passe du commerce mobile au commerce ubiquitaire (u-commerce).

Le présent travail de recherche vise à concevoir un système nomade intelligent de guidage des personnes handicapées dans les hypermarchés pouvant être généralisé à toute la clientèle. Il s'agit d'un système de géolocalisation multi-canal appelé système i-GUIDE. Il assistera l'utilisateur tout au long de son parcours, l'aidera à préparer ses achats, choisir ses produits, les localiser dans le magasin, obtenir des conseils et suggestions... De plus, ce système lui fera gagner du temps sur les achats besoins pour dégager du temps pour les achats plaisirs. Par ailleurs, ce système sera également bénéfique pour l'entreprise puisqu'elle pourra booster ses achats en proposant des produits et en diffusant des promotions personnalisées. Ce système favorisera une attraction et une fidélisation des clients ayant trouvé le confort et le plaisir de faire leurs courses.

Dans ce travail, nous nous intéressons à l'étude de faisabilité technologique et logicielle du système. Pour cela, une étude d'antériorité s'imposera pour se positionner par rapport aux projets et systèmes réalisés ou en cours de réalisation. Ces dernières années, plusieurs

systèmes de guidage des personnes dans différents endroits sont apparus. Ces systèmes seront présentés et analysés. L'analyse fonctionnelle du besoin représente une étape indispensable pour la création d'un système. Dans notre étude, nous traitons des personnes avec handicap visuel et d'autres à mobilité réduite. Ces personnes n'ont pas les mêmes besoins ni les mêmes contraintes. Par conséquent, il faut étudier les besoins de chaque cas et proposer un système global qui répond à leurs attentes.

Par ailleurs, nous proposons de faire une modélisation de la chaîne logistique d'achat et de vente des produits et vérifier l'adaptabilité d'un système de navigation intelligent dans un magasin. De plus, il faut étudier les différentes fonctionnalités du système, les acteurs qui interviennent ainsi que les composantes nécessaires pour le faire fonctionner.

Un système de guidage indoor ne peut pas fonctionner sans une technologie de localisation. Nous présentons l'état de l'art sur les méthodes de localisations et leurs domaines d'application pour choisir la technologie la plus appropriée au système.

Par ailleurs, le système nécessite un calculateur permettant de déterminer le chemin à parcourir pour récupérer les articles existants dans la liste de courses. Cependant, avant de calculer le chemin le plus court, il faut estimer la distance entre les articles du magasin pris deux à deux.

Les travaux présentés dans ce mémoire s'articulent autour de quatre chapitres. :

Dans le premier chapitre, nous détaillons le contexte de l'étude, l'étude de l'antériorité ainsi que l'analyse fonctionnelle du besoin du système.

Dans le deuxième chapitre, nous présentons une revue de la littérature sur les méthodes de modélisation des processus de l'entreprise. Une méthode sera retenue pour la modélisation de la chaîne logistique d'achat/vente des produits dans le magasin Leroy Merlin. Une analyse des exigences du système basée sur la modélisation UML fera l'objet de la deuxième partie du chapitre.

Dans le troisième chapitre, après avoir exposé les technologies de localisation existantes sur le marché, une sera retenue pour déterminer la position actuelle de l'utilisateur dans le magasin. Ensuite, nous présentons un prototype du système. Deux applications sous Android ayant deux architectures différentes seront présentées et discutées.

Dans le quatrième chapitre, nous proposons des approches d'optimisation de parcours dans un hypermarché. Après une revue de la littérature, une modélisation mathématique du problème sera présentée. Plusieurs méthodes seront présentées et testées. Nous introduisons une méthodologie pour le calcul des distances entre articles pris deux à deux. Dans ce chapitre, nous traitons le problème d'optimisation de parcours dans un hypermarché pour des articles avec et sans fenêtres de temps.

CHAPITRE 1 : CONTEXTE GENERAL DE L'ETUDE

1.1 Introduction et problématique de l'étude

Cette étude s'inscrit dans le cadre du projet i-GUIDE du Pôle de Compétitivité des Industries du Commerce (PICOM) et des programmes régionaux sur l'aide aux personnes, en particulier avec handicap, basée sur les nouvelles technologies pour une mobilité autonome et accrue. En effet, le PICOM représente un territoire défini par toutes les activités amont et aval qui concourent à améliorer le positionnement concurrentiel des entreprises, dans une logique de filière visant à la création de valeur pour le consommateur. Il réunit les entreprises dans différents secteurs tels que : la distribution, la vente à distance, la logistique et le secteur des Technologies de l'Information et de la Communication en plus les Universités, les Grandes Ecoles, les centres de recherche et les acteurs publics. Ainsi, les activités du pôle se résument dans le transfert de technologies et la valorisation de la recherche, l'analyse économique et la veille prospective, l'appui des start-up ainsi que la promotion et le marketing international de la filière en vue d'avoir des entreprises leaders dans leur secteur sur tous les marchés de consommation du monde et faire de la métropole lilloise la capitale internationale du commerce du futur.

Cette étude est réalisée en collaboration entre l'Ecole Centrale de Lille, la région Nord Pas de Calais et l'entreprise Leroy Merlin spécialiste de la grande distribution en bricolage et aménagement de la maison. L'objectif de notre travail est de développer des approches de modélisation et d'optimisation pour la conception d'un système interactif d'assistance aux personnes avec handicap, en particulier, et les personnes, en général, dans la grande distribution. Par ailleurs, le système i-GUIDE est un système orienté « service au client » avec une démarche citoyenne pour une meilleure gestion de la relation clients via un système intelligent.

Le présent chapitre représente la première phase de ce travail de recherche. Il a pour objet l'étude du besoin, la définition d'un système répondant à ce besoin ainsi que l'étude de sa faisabilité technologique. Pour ce faire, nous commençons par une étude d'antériorité en vue d'avoir une vision globale sur ce qui a été proposé et réalisé auparavant. Ensuite, nous passons à l'analyse fonctionnelle. Etant donné que, dans notre étude, nous traitons deux types de personnes avec handicap qui n'ont pas les mêmes besoins ni les mêmes contraintes, nous

commençons par l'étude de chaque cas ensuite nous proposons un système global qui répond à leurs attentes.

1.2 Etude d'antériorité

L'étude de l'antériorité consiste à rechercher les projets réalisés ou en cours de réalisation ayant des objectifs proches de ceux de notre projet. Ceci nous permettra d'avoir une idée sur les technologies utilisées et les résultats obtenus.

Nous présentons, dans la suite, quelques systèmes réalisés pour faciliter le déplacement des personnes handicapées dans différents endroits et favoriser ainsi leur autonomie.

1.2.1 Systèmes d'aide au déplacement des personnes avec handicap visuel

Le handicap visuel concerne les personnes aveugles et les personnes malvoyantes [Mon07]. Par ailleurs, l'Association pour les Personnes Aveugles et Malvoyantes (APAM) définit la malvoyance comme suit :

« Une personne malvoyante est une personne dont la déficience visuelle entraîne une incapacité dans l'exécution d'une ou plusieurs des activités suivantes :

- lecture et écriture (vision de près) ;
- appréhension de l'espace et déplacements (vision de loin) ;
- activités de la vie quotidienne (vision à moyenne distance) ;
- communication (vision de près et à moyenne distance) ;
- poursuite d'une activité exigeant une attention visuelle prolongée. »

Plusieurs systèmes ont été réalisés pour faciliter le déplacement de ces personnes dans divers endroits tels que : la plage, le métro, la rue, l'épicerie... Ces systèmes seront présentés brièvement dans la suite.

1.2.1.1 Le système Audioplage

L'audioplage est un système qui a été conçu par la société Urtech [Urt09], à l'initiative de l'association de non-voyants Cap Horizon. Sa deuxième version a été mise en place en 2009. C'est un système qui permet aux déficients visuels de se baigner librement, en toute autonomie et dans des conditions de sécurité optimales. Il s'agit d'un bracelet-montre électronique équipé de trois boutons d'appel [Urt09] :

- Le premier pour se localiser : les balises en mer diffusent un message sonore.

- Le deuxième pour retourner à la plage : le totem de plage émet un signal facilitant le retour vers la plage.
- Le troisième pour demander une assistance : le poste de secours reçoit le message de détresse.

Une fois dans l'eau, la simple activation d'une touche du bracelet imperméable active la bouée sonore la plus proche et donne à l'utilisateur des informations sur la profondeur de l'eau et la distance de la bouée suivante. Ainsi, il peut nager librement d'une bouée à l'autre. En cas de danger, il appuie sur une touche du bracelet pour appeler directement le poste de secours le plus proche. Des balises tactiles sont également aménagées dans l'eau pour ne pas perdre son chemin. Cependant, ces équipements n'occupent qu'une petite partie de la plage.

1.2.1.2 Le système BlueEyes

BlueEyes est un système proposé par l'entreprise Faber Novel en partenariat avec la RATP (Régie Autonome des Transports Parisiens) et l'APAM [Nie10], [Mab09], [Gan09]. Ce système a vu naissance en 2006. Son objectif est de faciliter le déplacement d'une personne avec handicap visuel dans le métro et RER.

Ce système est composé de trois éléments structurels:

- un serveur à distance hébergeant une base de données cartographique ;
- un réseau de balises radio fixes ;
- une application sur téléphone mobile Bluetooth communiquant avec les 2 autres éléments.

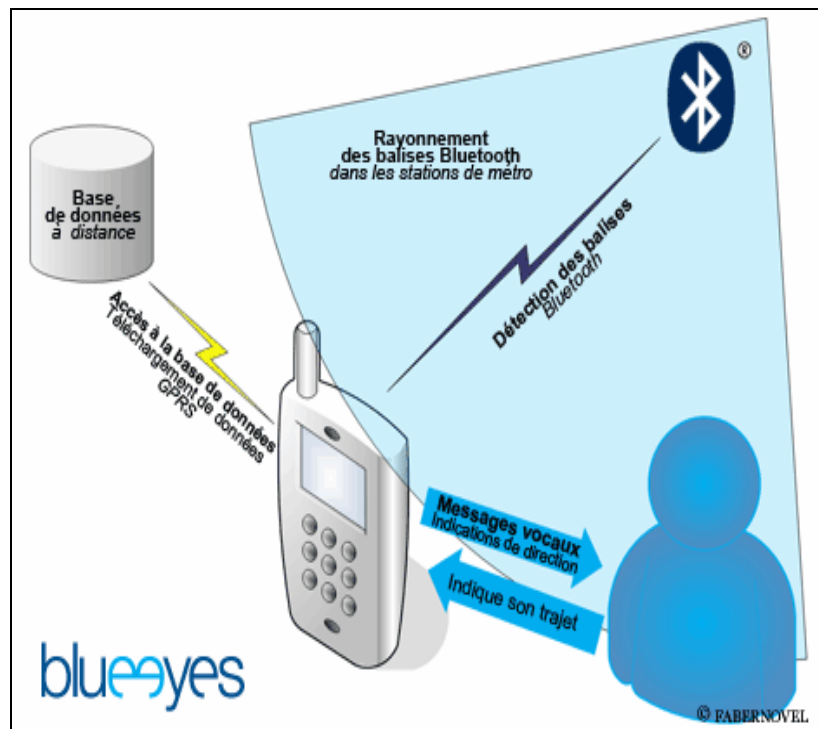


Figure 1 : Fonctionnement du système BlueEyes [Mab09]

Comme l'indique la figure 1, le téléphone mobile accède à la base de données à distance et télécharge les données pertinentes. Les balises se trouvent sur les panneaux ou sur les murs. Elles indiquent au terminal mobile leur position ce qui permet à l'application de guider l'utilisateur en fonction du trajet spécifié. Lorsque l'application BlueEyes repère une balise, elle en récupère les références. A ces dernières correspondent un certain nombre d'informations de guidage qui sont transmises à l'utilisateur vocalement par le biais d'une oreillette filaire. Ainsi, à son entrée dans le métro, l'utilisateur sélectionne l'application BlueEyes dans le menu de son téléphone mobile et précise le trajet souhaité oralement ou en utilisant le clavier. Il est détecté à chaque fois qu'il passe dans la zone de rayonnement d'une balise. A chaque balise, l'application calcule la position de l'utilisateur et son orientation. Un accès à la base de données BlueEyes permet de déterminer ensuite quel message dispenser à l'utilisateur. Cette application permet le calcul du trajet et le guidage de l'utilisateur dans les couloirs du métro ou RER (Réseau Express Régional), en temps réel, à l'aide des messages vocaux ou graphiques indiquant les directions à prendre. Cependant, ce système reste en phase d'expérimentation dans les métros de Paris.

1.2.1.3 Le système SmartEyes

Le système SmartEyes a été conçu, en 2006, pour aider les non ou malvoyants à se déplacer en ville sans avoir à demander leur chemin [Mic06]. Pour ce faire, des chercheurs grecs de l'université de Salonique ont mis au point un programme informatique pilote utilisant le guidage par satellite (GPS). Ainsi, l'utilisateur entre le lieu choisi dans son ordinateur de poche et le programme le guide de rue en rue ou dans le transport public et l'informe, par voie vocale, de l'environnement qui l'entoure (présence de feux de circulation, déroulement de manifestations de rue, présence de marchés...).

Après avoir testé le système par 150 volontaires à Athènes et à Salonique, les chercheurs ont constaté que ce système peut guider une personne malvoyante mieux qu'un chien incapable de guider sur des itinéraires inconnus. En revanche, cela ne se substituera pas à la canne blanche, indispensable pour repérer les obstacles imprévus [Mic06].

1.2.1.4 Le système Wayfinder Access

La société Wayfinder, connue pour ses solutions de navigation GPS, a présenté une application appelée Wayfinder Access. Cette application a été développée pour guider les personnes aveugles ou malvoyantes dans leurs déplacements. Wayfinder Access est un système de navigation pour téléphone mobile. Il fonctionne avec un logiciel de synthèse vocale, tels que le logiciel « Talks » de Nuance Communications ou « Mobile Speak » de Code Factory. Ce logiciel représente une interface sonore permettant d'utiliser le mobile entièrement à l'oreille. De plus, l'application intègre des cartes numérisées du partenaire TéléAtlas permettant de se repérer en Europe ou aux États-Unis. Ainsi, l'utilisateur entre la destination souhaitée, puis le système calcule l'itinéraire. Ensuite, le système précise vocalement la direction et le nom des rues, ainsi que les prochaines intersections se présentant sur la direction empruntée [RNIB10]. Par ailleurs, Wayfinder Access permet à l'utilisateur de mieux cerner son environnement et de chercher d'autres points d'intérêts tels que les banques, les bureaux de poste, les restaurants...

1.2.1.5 Le système RoboCart

Le RoboCart est l'un des résultats de l'équipe de recherche du professeur Vladimir Kulyukin, de l'université d'Etat de l'Utah aux États-Unis [Kul07]. C'est un robot conçu pour aider des personnes avec handicap visuel à se déplacer dans une épicerie. Les performances de ce système sont améliorées grâce à l'utilisation des étiquettes RFID (Radio Frequency

Identification). En détectant ces dernières, ce robot peut mener les personnes malvoyantes au rayon de leur choix et leur donner des informations vocales pour les aider à se repérer et à trouver l'article de leur choix.

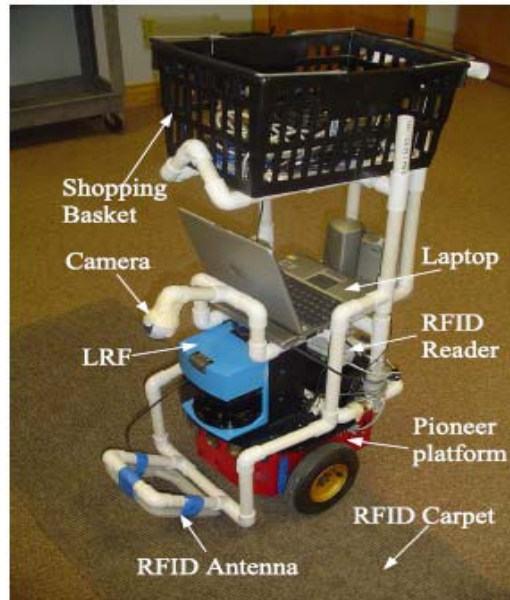


Figure 2 : Le Robocart [Kul07]

Comme le montre la figure 2, le système est composé de :

- un ordinateur portable connecté à la plateforme,
- des capteurs Laser ou LRF (Laser Range Finder) ;
- un RFID Reader attaché à une antenne RFID ;
- une Camera Logitech fixée verticalement vers le bas ;
- un panier pour mettre ses courses.

L'utilisateur utilise un clavier numérique, attaché au poigné du chariot, pour communiquer avec le robot. Un menu vocal est disponible pour permettre à l'utilisateur de sélectionner les produits, de se déplacer, de s'arrêter et de reprendre son chemin. De plus, il existe un lecteur de codes-barres sans fil rattaché à l'ordinateur portable pour scanner l'article et vérifier s'il est l'article recherché avant de le mettre dans son panier.

1.2.1.6 Le Réseau SESAMONET

Un réseau SESAMONET (SEcure and SAfe MObility NETwork) a été mis en place dans la ville de Laveno Mombello, au nord de l'Italie, dans le cadre du projet SESAMONET du Centre Commun de Recherche de la Commission Européenne. Ce réseau comprend un millier de micropuces RFID enfouies dans le sol à un intervalle de 60 cm permettant aux

malvoyants de se déplacer en connaissant précisément leur position [CMRS06], [RAS09], [CMSABR09].

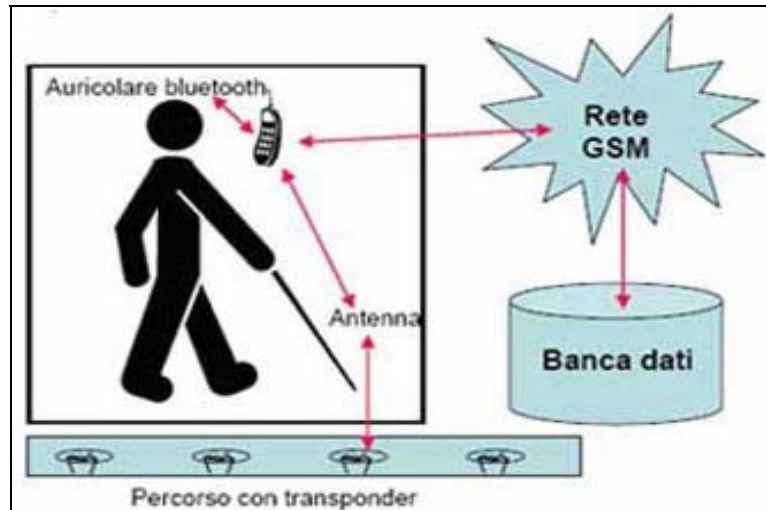


Figure 3 : Schéma de fonctionnement du dispositif [RAS09]

Comme le montre la figure 3, les malvoyants sont équipés d'une canne spéciale intégrant une antenne qui envoie des signaux aux puces RFID. Ces dernières les transmettent à une borne contenant une base de données sur la localisation. Ainsi, les promeneurs connaissent précisément leur position grâce à des informations qu'ils reçoivent via une oreillette Bluetooth [RAS09].

1.2.1.7 Le système de NTT DoCoMo

L'opérateur de téléphonie mobile japonais NTT DoCoMo, numéro 1 au Japon, a conçu un prototype de téléphone qui, pour dialoguer avec son environnement direct, utilise le Bluetooth et toute autre technologie de communication sans fil de proximité [Pou07]. Ce terminal utilise le corps comme mode de locomotion pour transmettre des données à des capteurs au sol, aux poignées, aux portiques de métro et autres objets. Le principe consiste à utiliser la conductivité naturelle du corps pour envoyer des informations. Cette technologie est censée faciliter la vie de son utilisateur dans ses actes quotidiens une fois intégrée dans un téléphone portable. Ainsi, par exemple, deux personnes équipées d'un tel mobile pourront se transmettre leurs coordonnées et d'autres informations contenues dans leurs terminaux simplement en se serrant la main.



Figure 4 : Utilisation du mobile NTT DoCoMo [Pou07]

De plus, un aveugle peut être guidé vocalement par son terminal cellulaire à synthèse vocale en recevant des informations depuis le sol par des étiquettes électroniques, via sa canne blanche et sa main, sans utiliser la localisation par satellite GPS ni le réseau cellulaire comme le montre la figure 4. Pour le moment, les débits du prototype actuel NTT DoCoMo sont relativement faibles, mais cela n'est pas une limite car les chercheurs ont déjà démontré que cette technologie permet d'atteindre des performances nettement supérieures au Bluetooth. La technique mise en œuvre a aussi pour avantage d'être économe en énergie. Elle ne nécessite en outre pas de jumelage initial, contrairement aux techniques hertziennes comme Bluetooth ou WiFi. De plus, les chercheurs de NTT DoCoMo prévoient de recharger la batterie des mobiles en utilisant les mouvements du corps [Pou07].

Plusieurs chercheurs travaillent sur les effets de cette technologie économe sur le corps humain [Lou08].

1.2.2 Systèmes d'aide aux personnes à mobilité réduite

Le handicap moteur recouvre l'ensemble des troubles pouvant entraîner une atteinte partielle ou totale de la motricité, notamment des membres supérieurs et/ou inférieurs et la personne trouvera :

- des difficultés pour se déplacer,
- des difficultés pour conserver ou changer de position
- des difficultés pour prendre et manipuler
- des difficultés pour effectuer certains gestes.

Il existe quelques services qui sont mis en place en collaboration avec des associations pour aider les personnes à mobilité réduite à surmonter ces difficultés. Parmi ces services, on peut citer le service Handi-cité et le service Afmaster Robotique.

1.2.2.1 Le service Handi-Cité

Le service Handi-cité, lancé par l'opérateur SFR, est le premier guide des lieux de sortie et des lieux accessibles en ville aux personnes à mobilité réduite [OTeN07], [SFR08]. Il est consultable sur téléphone mobile. C'est un service gratuit hors coût de connexion. Il est accessible depuis le portail multimédia de SFR [SFR08]. Il répertorie tous les lieux et équipements accessibles aux handicapés moteurs dans les villes concernées tels que les places de stationnement réservées, les parkings accessibles, les distributeurs de billets accessibles, les toilettes accessibles et adaptées ainsi que les lieux publics accessibles (restaurants, café, bars, cinéma...) et les arrêts de bus et tramway accessibles. L'expérimentation de ce service avait lieu en collaboration avec l'association "Mobile en ville", qui milite pour favoriser l'accessibilité des villes aux roulettes et est soutenue par l'Association Française contre les Myopathies (AMF).

1.2.2.2 Le système Afmaster Robotique

La société française AFMA-ROBOTS a développé, en collaboration avec l'association A.P.P.R.O.C.H.E (Association pour la Promotion des Plates-formes RObotiques Concernant les personnes HandicapEes), la robotique d'aide aux personnes handicapées [GCLLB99]. C'est une station robotisée permettant à la personne handicapée d'accomplir, à l'aide d'un bras télémanipulé, un certain nombre de tâches qu'elle choisit en fonction de ses motivations, ses désirs et ses besoins. Le pilotage du robot se fait par un ordinateur et une interface personnalisée. Ce robot peut être utilisé dans différents domaines tels que : la bureautique, le télétravail, la vie domestique, les loisirs...

1.2.3 Systèmes d'aide au déplacement des personnes

Il existe des systèmes de guidage des personnes dans les épiceries, les magasins et les hypermarchés pour trouver les articles recherchés. Parmi ces systèmes, on peut citer le système MediaCart de Microsoft, le système MIGS (Mobile Information & Guidance Services) d'IBM et le système Keyshopping de Keyneosoft.

1.2.3.1 Le système MediaCart

La société MediaCart Holdings a mis au point le système MediaCart en collaboration avec Microsoft [Med08], [Mil08]. Il s'agit d'un caddie équipé d'un écran plasma placé au bout du chariot face au client et d'une manette de contrôle incrustée dans la barre qui sert à

pousser l'engin. Ce caddie sert à guider le client pendant qu'il fait ses courses en lui permettant de gagner du temps et d'effectuer un meilleur choix. Le système va, tout d'abord, indiquer au client où trouver les produits qu'il cherche. Une fois arrivé dans le rayon, plusieurs informations s'affichent sur l'écran. Parmi ces informations, on peut trouver les promotions du jour, les meilleurs prix, les compositions des produits et surtout des publicités sur des produits au rayon. Ainsi, le système MediaCart permettra d'avoir une influence sur le comportement d'achat de l'utilisateur [Mic09]. Des étiquettes RFID ont été utilisées pour suivre les mouvements de l'utilisateur et lui envoyer des informations personnalisées [Mil08]. Par ailleurs, le client n'a plus à décharger et recharger ses achats à la caisse. Il suffit de passer les articles devant le lecteur de codes situé à l'avant du chariot avant de les poser dedans. Les chariots seront également équipés d'une mémoire pour afficher la liste des produits préférés de l'utilisateur ainsi qu'un plan du magasin pour l'aider à se déplacer dans les rayons.

Il est vrai que l'idée du système MediaCart pourrait intéresser les hypermarchés puisqu'il permet d'attirer des fournisseurs pour intégrer leurs publicités dans le système et d'avoir une influence sur la consommation des utilisateurs mais il n'est pas adapté aux besoins et aux contraintes des personnes avec une déficience visuelle et les handicapés moteurs.

1.2.3.2 Le système MIGS

MIGS est une solution développée par IBM. Elle permet de guider une personne dans un lieu déterminé et d'obtenir des informations contextuelles dépendant de sa position et de son profil. L'utilisateur a le choix de s'inscrire sur le portail de l'entreprise en enregistrant ses besoins et ses préférences ou de télécharger l'application directement à l'entrée du lieu souhaité. Ainsi, il est détecté dès son entrée par l'intermédiaire de son téléphone portable ou PDA via le Bluetooth ou le Wifi et il peut demander à être guidé à l'intérieur du lieu et récupérer ainsi les indications soit sur un plan s'affichant sur son téléphone ou PDA soit des instructions vocales. De plus, le client peut recevoir des informations selon ses préférences et ses habitudes d'achat et il peut même recevoir des coupons de réduction sur son téléphone ou PDA. L'utilisateur peut être identifié et suivi dans sa visite du lieu grâce aux moyens de localisation et d'analyse de son parcours. Les concepteurs de la solution proposent différentes technologies de localisation telles que les étiquettes RFID, le réseau Wifi s'il existe déjà, les balises Bluetooth... Le choix de la technologie dépend de la capacité de l'appareil mobile, de l'infrastructure existante ou à mettre en œuvre et de la précision nécessaire. Dans son rapport

[Mon07], Gilbert Montagné précise que ce système sera aussi adapté aux personnes aveugles. Cependant ce système reste encore en phase de test.

1.2.3.3 Le système Keyshopping

Keyshopping est une des solutions développées par Keyneosoftware. Ce système a été conçu pour aider les utilisateurs à trouver les articles entrés à l'aide d'un plan du magasin affiché sur l'écran. Ce système a été installé dans quelques magasins Leroy Merlin et Auchan sous forme de borne. Nous avons utilisé la borne Keyshopping qui existe dans le magasin Leroy Merlin à Verquin et nous avons remarqué la lenteur du système (dû à la lenteur de son algorithme d'optimisation de l'itinéraire), la complexité de son utilisation même pour des personnes sans handicap (noms de produits très techniques sans photos) ainsi que l'absence d'entretien du système (il se plante, il n'y a pas d'encre pour imprimer la liste...) et surtout l'absence de communication et de formation. Le système n'était pas vraiment remarquable et peut être les clients ne savent pas qu'il existe un système pour les aider à trouver leurs articles et il n'y a personne pour les aider à l'utiliser. Ensuite, le système a été amélioré, dans les magasins Auchan, et Keyneosoftware a ajouté à la borne un terminal de poche qu'on peut récupérer à l'entrée du magasin. Ce terminal permet d'ajouter une liste de catégories d'articles, trace le chemin le plus court pour les trouver sur le plan du magasin qui sera affiché sur l'écran et à chaque fois où l'utilisateur scanne et valide son produit, son prix sera ajouté à l'addition et l'utilisateur payera à la fin cette addition sans décharger et charger ses articles ce qui lui évitera de faire la queue. Pour le moment, seulement deux magasins Auchan disposent de ce terminal de poche Keyshopping. L'utilisateur peut aussi se connecter sur le site de Keyshopping, choisit un de ces deux magasins et préparer sa liste de courses. Comme le montre la figure 5, un plan du magasin sera affiché avec le chemin à parcourir. Ensuite, l'utilisateur peut imprimer ce plan.

Certes le système Keyshopping facilite les achats des clients dans un magasin mais ce système reste toujours statique et ne répond pas aux besoins des personnes à déficience visuelle ou à mobilité réduite d'où l'intérêt de notre étude.



Figure 5 : Plan du magasin avec le chemin à parcourir [Pue10]

Récapitulation :

Cette étude d'antériorité a permis d'avoir une idée précise sur les nouvelles technologies employées pour l'aide au déplacement des personnes telles que les étiquettes RFID et les balises Bluetooth pour la localisation des objets, le Wifi, le Bluetooth et même le corps humain pour le transfert des informations et les satellites GPS pour le positionnement des utilisateurs. Par ailleurs, à travers cette étude, nous nous sommes rassurés que le système qu'on veut concevoir n'existe pas encore sur le marché et même les applications mobiles déjà faites sont des applications sur téléphone d'ancienne génération plutôt orientées échanges d'information. Ainsi, notre système va apporter une grande aide aux personnes handicapées, en particulier, et à n'importe quel utilisateur en général. Dans la section suivante, nous allons définir le cahier des charges fonctionnel de notre système.

1.3 Analyse fonctionnelle du besoin/ Cahier des charges du système i-GUIDE

L'analyse fonctionnelle du besoin est une étape indispensable pour la création ou l'amélioration d'un produit. Elle se présente le plus souvent sous forme graphique [CaB10], [Tas06]. Il existe plusieurs méthodes et outils d'analyse fonctionnelle permettant de caractériser les fonctions de service attendues et générées par l'usage d'un produit. Parmi ces

méthodes, on peut citer la méthode APTE (APplication des Techniques d'Entreprise), la plus ancienne et la plus utilisée par la plupart des cabinets de conseil [CaB10].

Dans la suite, nous présentons deux outils de la méthode APTE que sont : la « bête à corne » et le « diagramme pieuvre » ainsi que le diagramme FAST (Functional Analysis System Technic) du système. Etant donné que dans notre étude nous traitons deux cas de personnes avec handicap qui n'ont pas les même besoins ni les même contraintes, deux systèmes seront étudiés selon les attentes de chaque utilisateur. Ensuite, un système global sera proposé.

1.3.1 La « bête à cornes »

Cet outil définit le besoin auquel répond le système [CaB10], [Tas06]. Pour cela, il est essentiel de se poser les trois questions suivantes :

- A qui/ quoi le produit rend-il service ?
- Sur qui/quoi agit-il ?
- Dans quel but ?

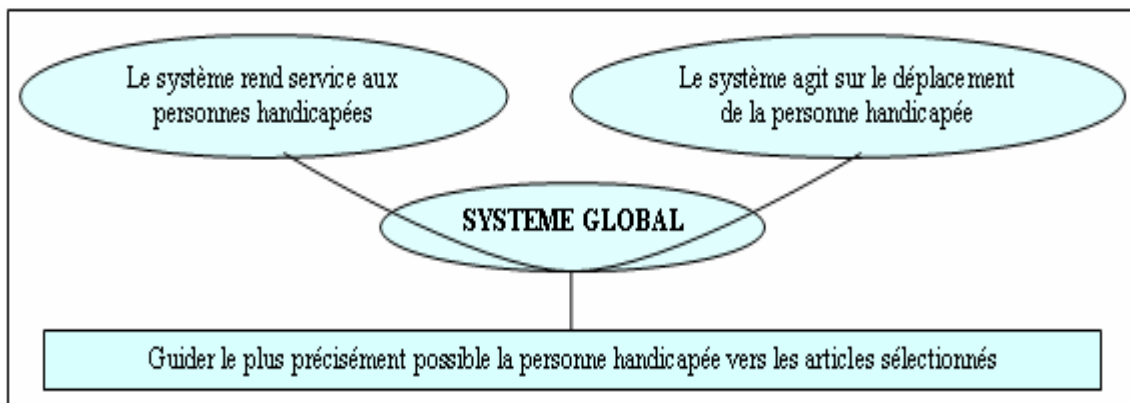


Figure 6 : La " bête à cornes" du système global

La figure 6 représente la « bête à cornes » du système.

1.3.2 Le diagramme pieuvre

Le diagramme pieuvre est un diagramme qui met en évidence les relations entre le système et les différents éléments de l'environnement qui l'entourent. Ces relations sont appelées les fonctions de service et elles sont de deux types [Tas06], [CaB10] :

- *Les fonctions principales (FP)* : ce sont les fonctions pour lesquelles le produit est élaboré, donc celles qui pourraient répondre aux exigences de l'utilisateur.

- *Les fonctions contraintes (FC):* elles ont pour but de définir les contraintes imposées au produit par son milieu extérieur d'utilisation. Elles identifient les relations entre l'objet et un élément du milieu extérieur.

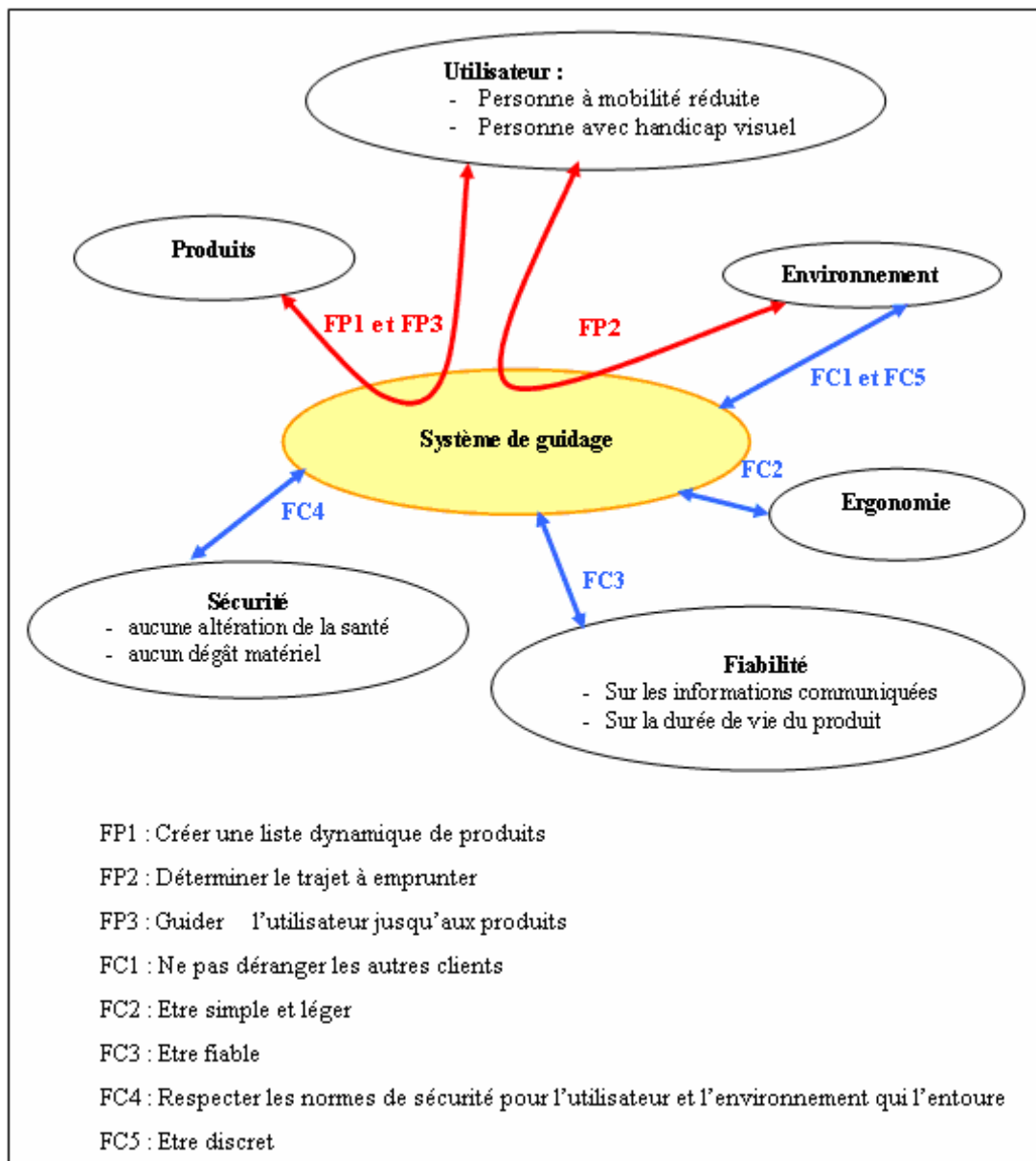


Figure 7 : Diagramme pieuvre du système global

La figure 7 illustre le diagramme pieuvre du système. Ses principales fonctions principales sont :

- Créer une liste dynamique de produits (FP1) : cette fonction permet d'ajouter une liste de courses, supprimer un article et ajouter un nouvel article si l'utilisateur le désire.

- Déterminer le trajet à emprunter (FP2) : pour cela il faut localiser les produits dans le magasin et déterminer la position actuelle de l'utilisateur ainsi que son orientation.
- Guider l'utilisateur jusqu'aux produits (FP3)

Par ailleurs, le système de guidage doit être léger et son utilisation doit être simple et facile pour que la personne handicapée se trouve à l'aise et retrouve le plaisir du shopping en l'utilisant. De plus, il doit être discret et ne dérange pas les autres clients du magasin et doit respecter les normes de sécurité pour l'utilisateur et l'environnement qui l'entoure.

Dans la section suivante, nous détaillons ces fonctions et proposons des solutions technologiques de réalisation.

1.3.3 Le diagramme FAST

Lorsque les fonctions sont identifiées, la méthode FAST les ordonne et les décompose logiquement pour aboutir aux solutions techniques de réalisation.

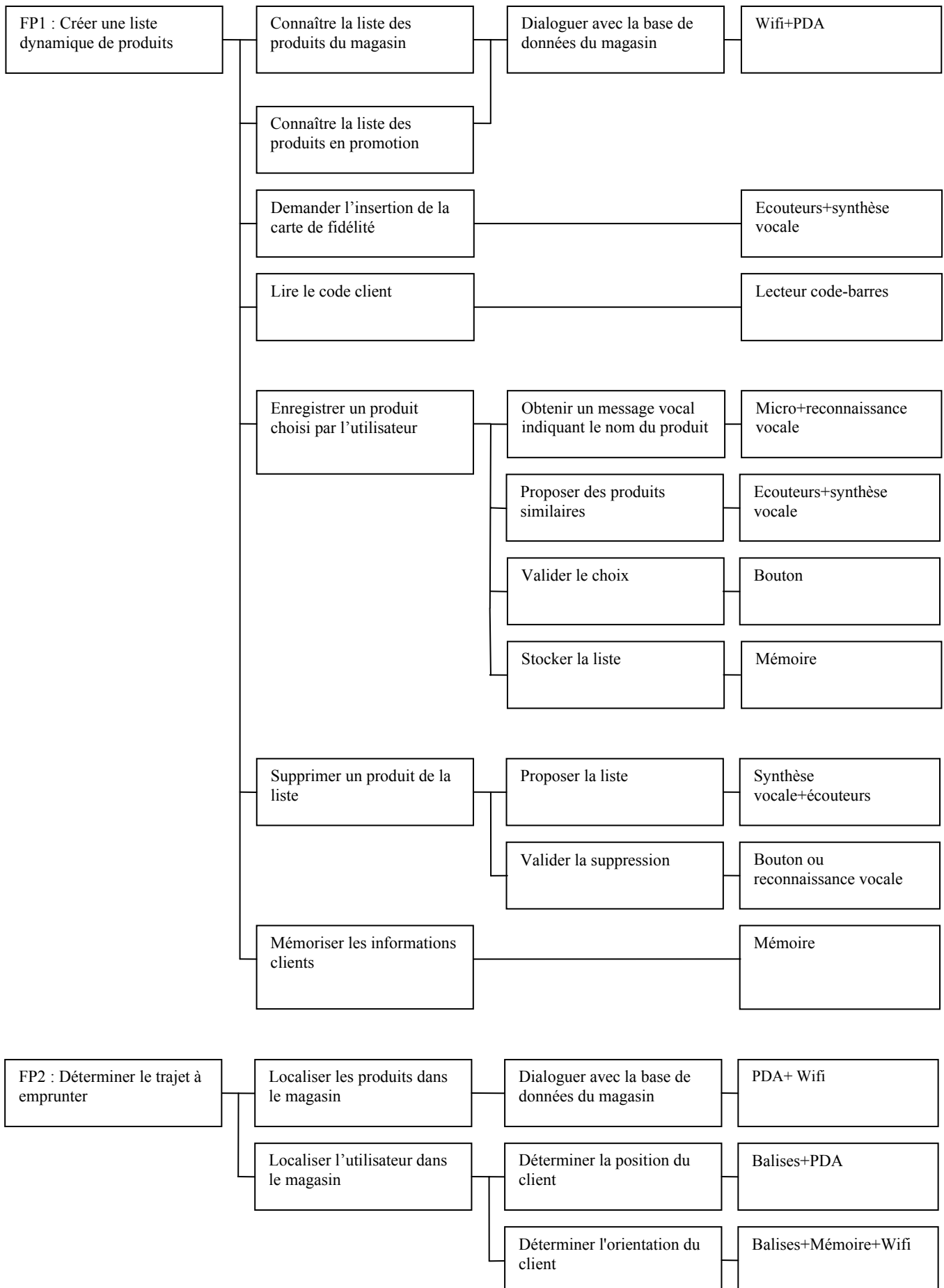
En partant de la fonction de service, le diagramme FAST présente les fonctions techniques et les solutions technologiques associées dans une organisation logique répondant aux trois questions essentielles :

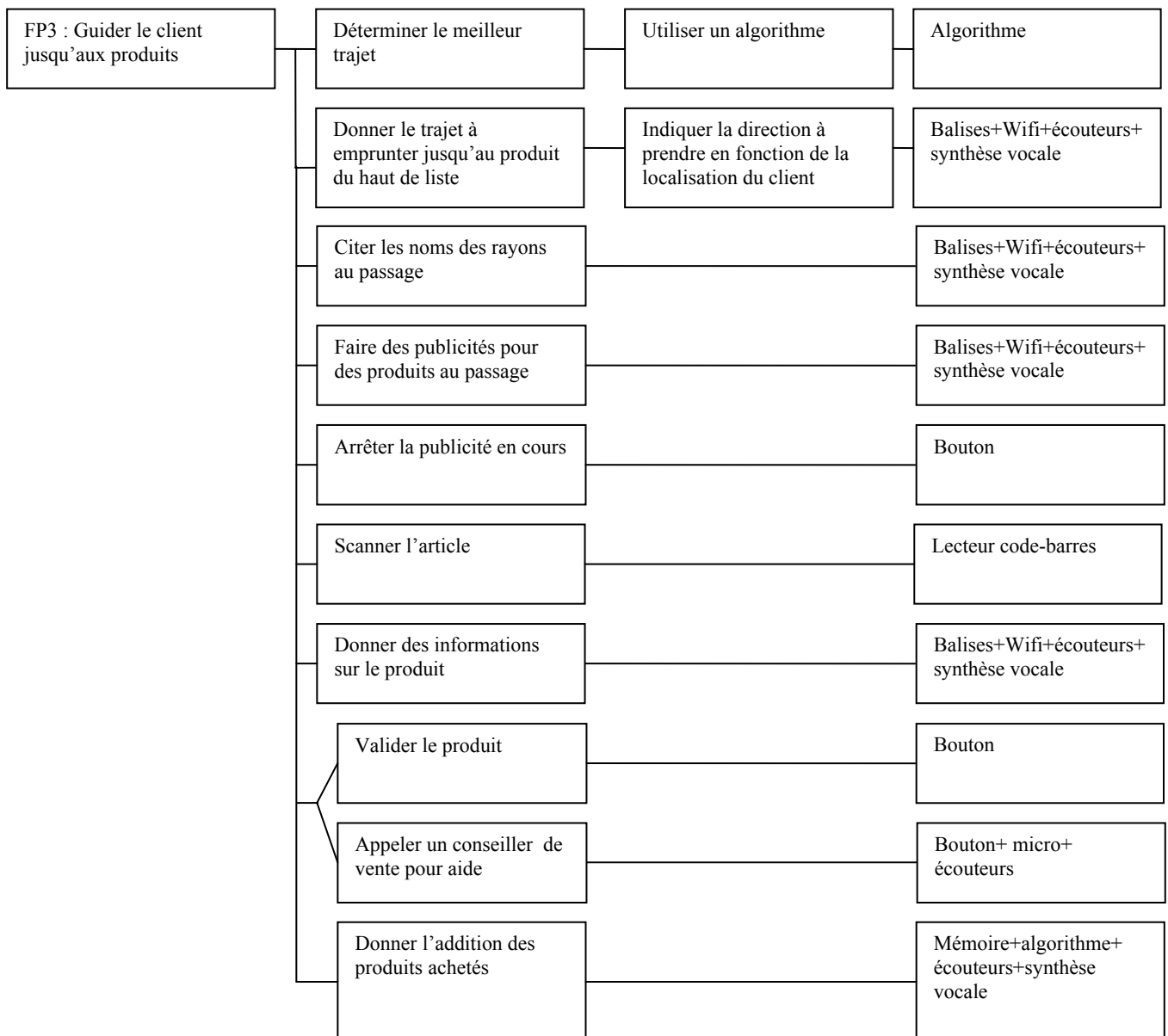
- Pourquoi cette fonction doit-elle être assurée ?
- Comment cette fonction doit-elle être assurée ?
- Quand cette fonction doit-elle être assurée ?

Nous avons contacté quelques associations telles que l'association des paralysés de France (l'APF) qui nous a expliqué les capacités et les nécessités de ce type d'handicap. En effet, un handicapé moteur se comporte comme une personne normale pour la recherche des produits dans un magasin. Cependant, son seul handicap est de pouvoir atteindre les objets en hauteur ou les objets éloignés donc il a besoin d'une aide à son arrivée devant le produit. Par ailleurs, nous avons rencontré un responsable de l'association de Valentin Haüy (AVH) qui nous a parlé des difficultés rencontrées par les personnes aveugles et malvoyantes et des nouvelles technologies d'aide à ces personnes telles que l'utilisation des ordinateurs, l'accès à internet et même l'utilisation des téléphones portables pour les personnes âgées.

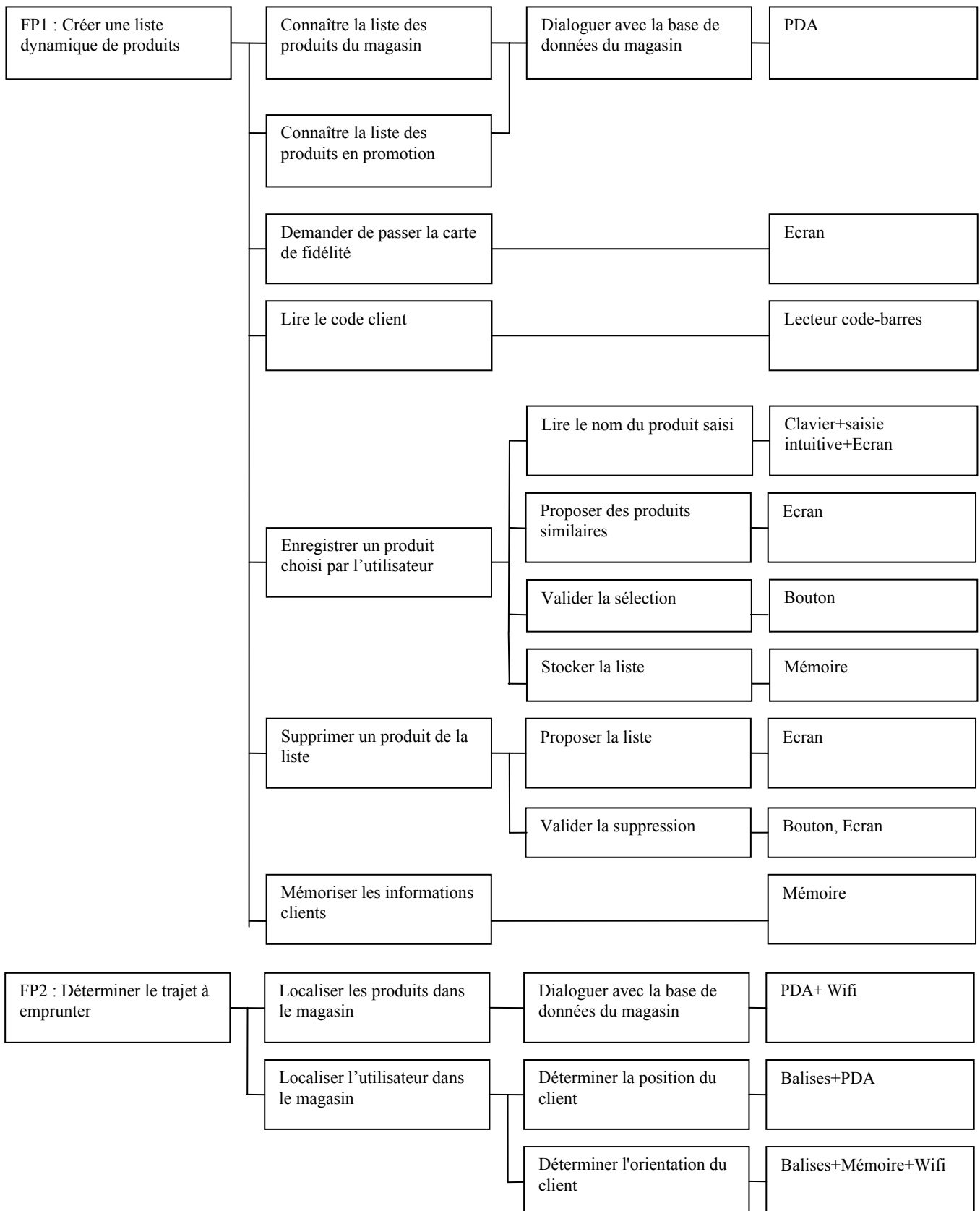
1.3.3.1 Personnes avec handicap visuel

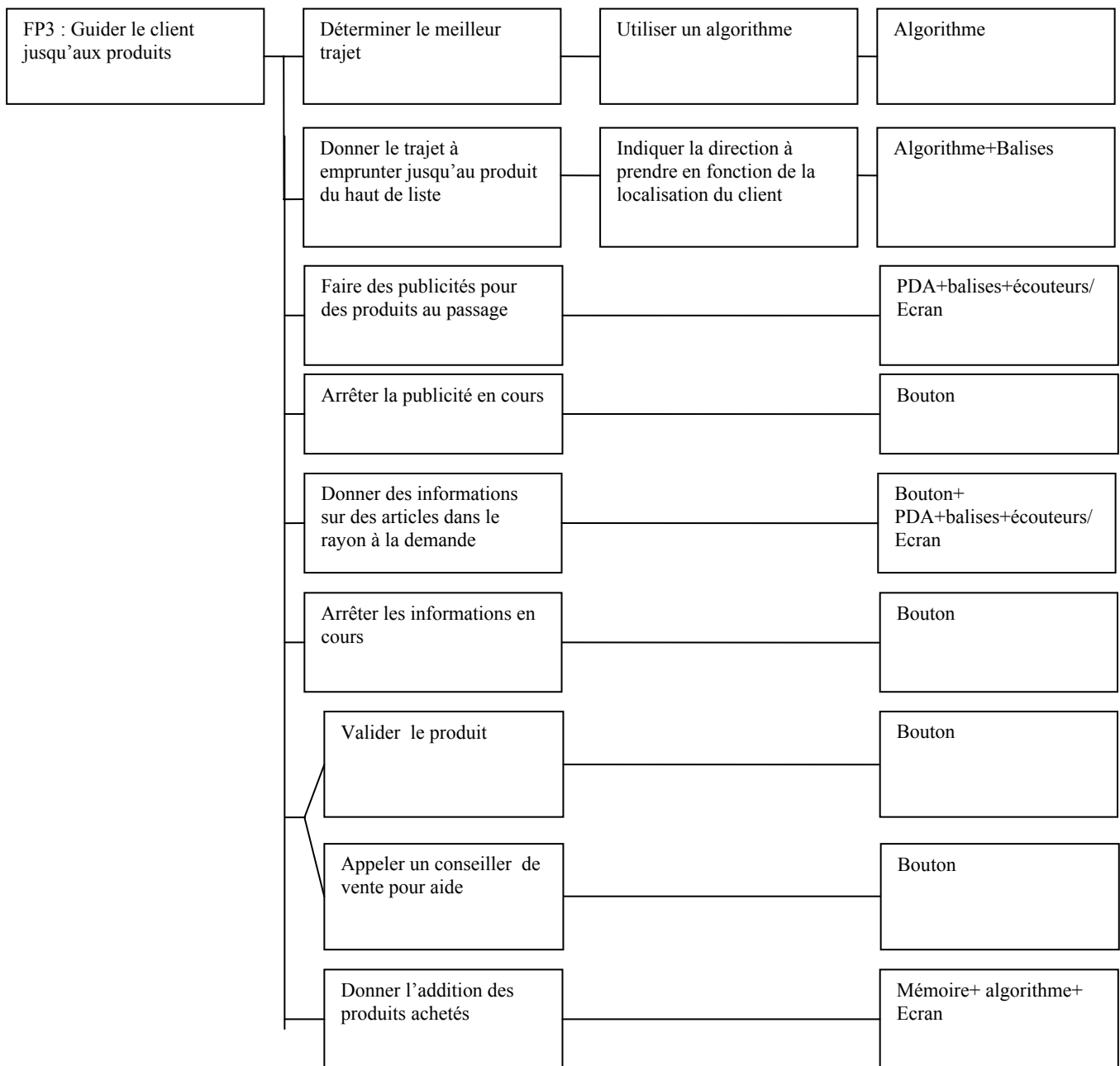
Nous proposons un système avec les technologies de synthèse et reconnaissance vocale afin de faciliter la communication entre l'utilisateur et le système.





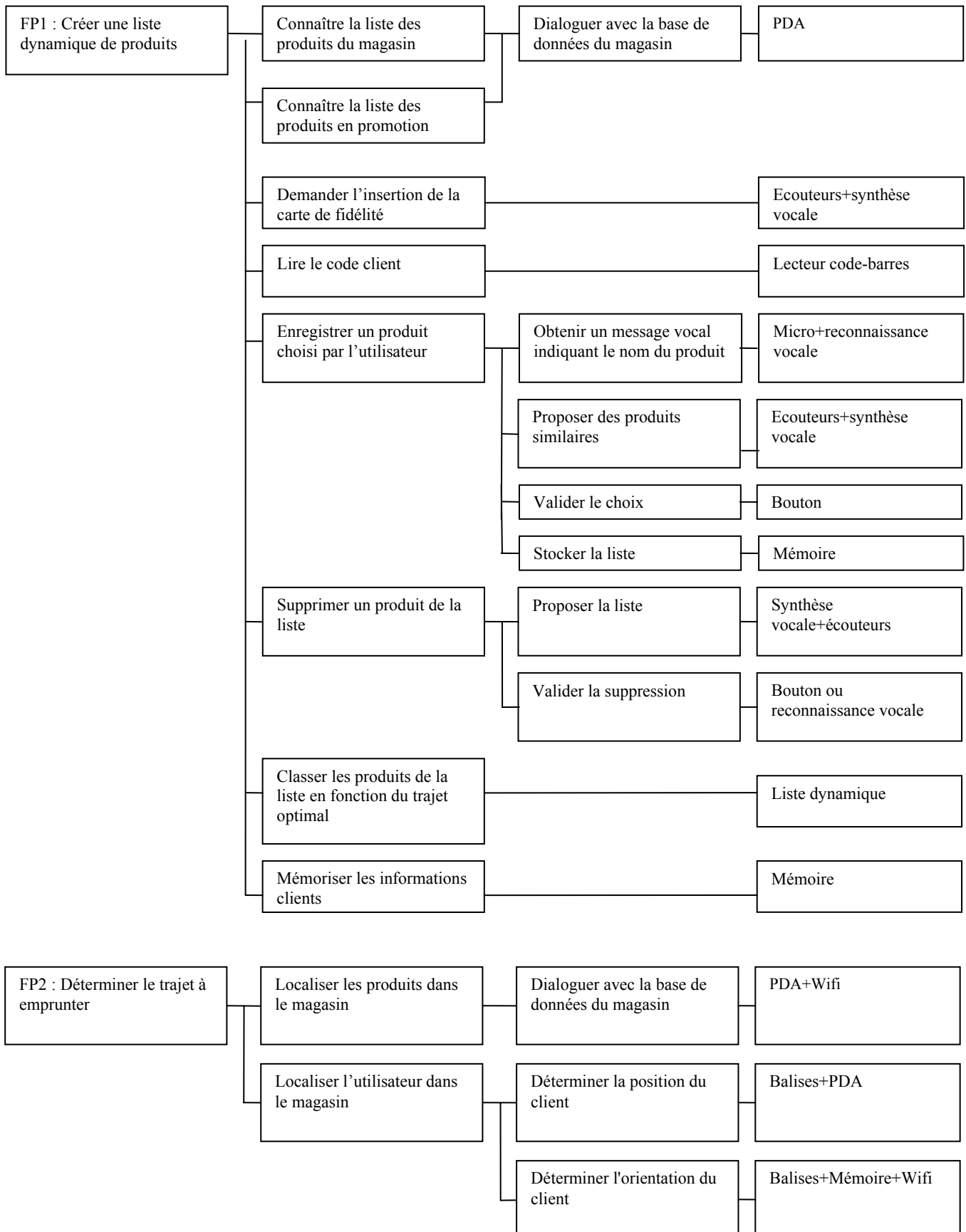
1.3.3.2 Personnes à mobilité réduite

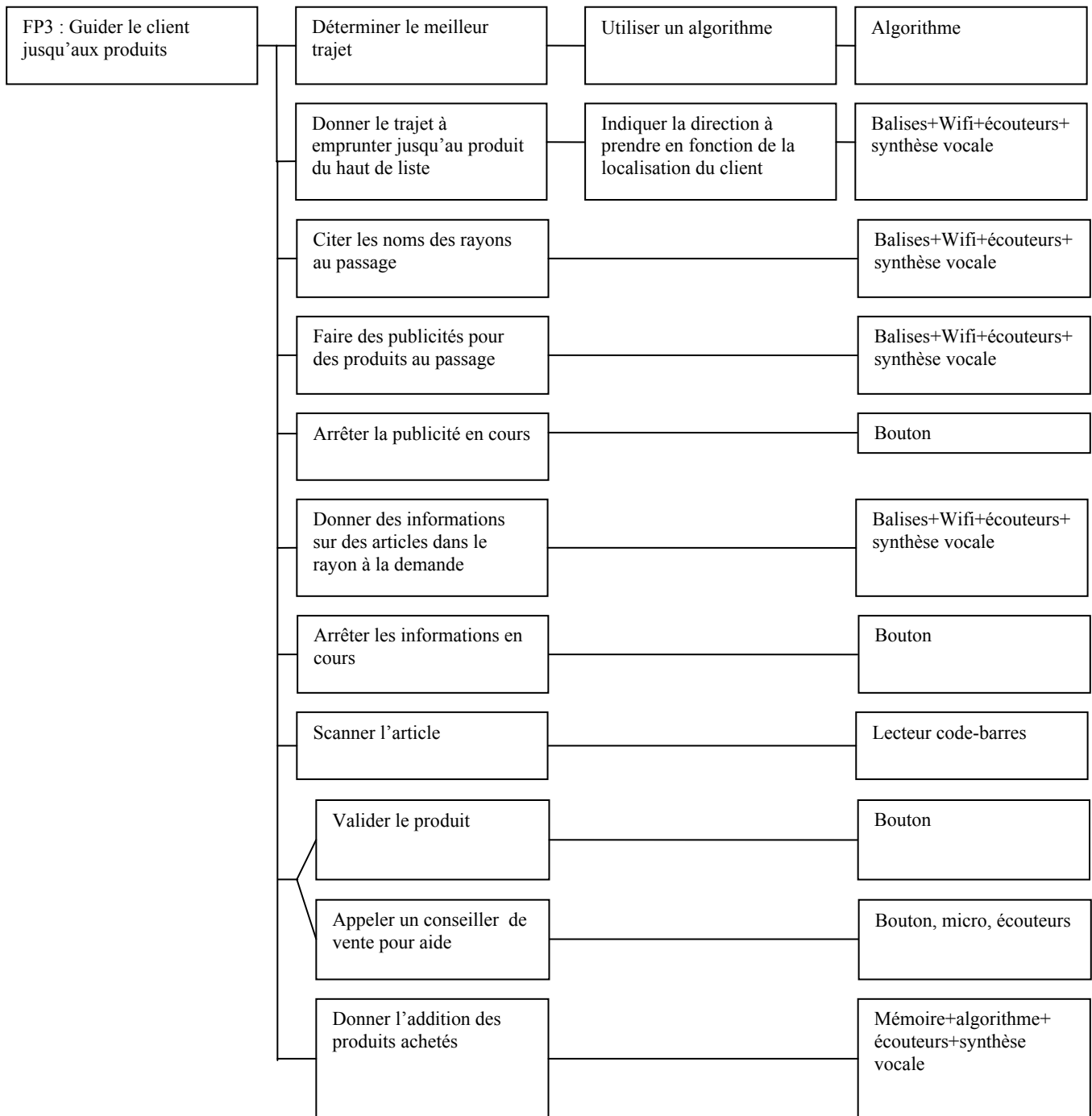




1.3.3.3 Système global

Pour adapter le système aux personnes à mobilité réduite ainsi qu'aux personnes avec un handicap visuel, nous proposons d'utiliser des écouteurs et un micro. Nous utilisons les technologies de synthèse et reconnaissance vocale afin de faciliter la communication entre l'utilisateur et le système.





Ce système permettra :

- une meilleure intégration des personnes handicapées dans la société ;
- une attraction et fidélisation de cette catégorie de clientèle ayant trouvé le confort et le plaisir de faire ses courses ;
- une attraction des fournisseurs pour intégration de leurs publicités dans le système ;

- une capitalisation des connaissances sur le comportement des clients avec ou sans handicap (produits achetés, périodes de fréquentation, degré de fidélité...) pour une amélioration du système et du service.

1.4 Conclusion

Dans ce chapitre, nous avons présenté le système i-GUIDE qui représente un assistant personnel de guidage des personnes dans les hypermarchés. Nous avons commencé par l'étude d'antériorité des projets qui a permis d'avoir une idée sur les nouvelles technologies employées pour l'aide au déplacement des personnes telles que les étiquettes RFID pour la localisation des objets, le Bluetooth et même le corps humain pour le transfert des informations et les satellites GPS pour le positionnement des utilisateurs. Nous nous concentrons sur ces technologies dans le chapitre 3 pour en choisir la technologie la plus adaptée à notre problème. Ensuite, nous avons défini le cahier des charges du système en utilisant la méthode APTE. Après avoir étudié les besoins, les attentes et les contraintes des personnes ayant un handicap visuel et ceux des handicapés moteurs, nous avons proposé un système global répondant à leurs besoins, en particulier, et aux besoins de toute personne en général. Par ailleurs, une veille technologique sera faite tout le long de l'élaboration de ce travail de recherche.

Avant la réalisation de notre système, nous étudions, dans le chapitre 2, l'interaction entre notre système et son environnement ainsi que les objets nécessaires pour assurer ses différentes fonctionnalités.

CHAPITRE 2 : CONCEPTION DU SYSTEME I-GUIDE

2.1 Introduction

Le but de notre étude est de concevoir un système de navigation intelligent et générique pour aider les personnes à se déplacer dans les grandes surfaces et ce dans la logique de développement de l'intelligence ambiante et du u-commerce (commerce ubiquitaire) préconisé par le PICOM. Ce système sera l'assistant personnel de l'utilisateur qui l'aidera à préparer ses achats, choisir ses produits, les localiser dans le magasin, obtenir des conseils et suggestions...

Dans ce chapitre, nous présentons une revue de la littérature sur les méthodes de modélisation des processus de l'entreprise pour schématiser les différents processus de cette chaîne logistique. Ensuite, nous prenons, comme exemple, les magasins Leroy Merlin de notre partenaire R&D, pour étudier la faisabilité technique du système. Une analyse de l'existant fera l'objet de la deuxième partie du chapitre en étudiant les activités des différents acteurs participant dans la chaîne logistique d'achat/vente des produits dans les magasins et les interactions entre eux. Enfin, nous optons pour l'approche UML pour étudier les interactions entre notre système autonome et temps réel et son environnement extérieur ce qui présente la base de l'analyse de besoin.

2.2 Modélisation de la chaîne logistique de vente/achat des produits dans le magasin

Dans la suite, nous présentons une revue de la littérature sur les méthodes de modélisation des processus de l'entreprise. Ensuite, nous choisissons une méthode pour modéliser le processus d'achat et de vente des produits dans un magasin.

2.2.1 Etat de l'art sur les méthodes de modélisation des processus de l'entreprise

La modélisation des processus métiers de l'entreprise et de ses activités devient, aujourd'hui, une étape indispensable pour mieux comprendre la situation actuelle, détecter les

dysfonctionnements et en déduire les causes dans le but d'une amélioration future des performances [HEB10a], [HEYDG09].

Dans la littérature, on trouve diverses méthodes de modélisation des processus de l'entreprise dont on peut citer : GRAI, UML, ACTIF, BPMN... Ces méthodes seront présentées dans la suite et on retiendra une d'entre-elles pour modéliser le processus d'achat et de vente des produits dans un magasin.

2.2.2 Modélisation avec GRAI-GIM

GRAI (Graphe à Résultats et Activités Inter-reliés) est une méthode basée sur la modélisation d'entreprise dans un but de conception ou de re-conception des systèmes de production (industriels ou de services) en se focalisant sur la partie décisionnelle de l'entreprise et en visant une amélioration globale des performances [DVZC92], [Rob93]. Cette méthode peut être appliquée dans divers domaines dont on peut citer : l'ingénierie des systèmes de production, le développement de la stratégie industrielle, le management des connaissances... En 1990, une méthodologie issue d'une extension de GRAI a été proposée pour les systèmes de production dans le cadre du projet ESPRIT de la Communauté Economique Européenne. Cette méthodologie est appelée GIM (GRAI Integrated Method) [DVZC92]. Elle s'appuie sur le modèle conceptuel de référence qui décompose l'entreprise en trois sous-systèmes : le système physique, le système d'information et le système décisionnel. Le cadre de modélisation de GIM est constitué de l'axe des vues de modélisation et de l'axe du cycle de vie. En outre, elle considère quatre vues de modélisation que sont : information (données/connaissances), décision (chaîne d'activités et centres de décision), physique (ressources) et fonction (décomposition fonctionnelle).

Par ailleurs, la méthodologie GIM propose de nouveaux concepts de description des systèmes décisionnels que sont : la grille GRAI et le réseau GRAI [DVZC92], [DV04]. La grille GRAI, comme le montre la figure 8, est un tableau dont les colonnes représentent l'ensemble des fonctions de l'entreprise. Les lignes représentent les niveaux décisionnels de l'entreprise, se rapportant aux différents types de décision en l'occurrence, les décisions stratégiques, tactiques et opérationnelles avec leurs horizons de temps respectifs. A l'intérieur de la grille, on trouve les centres de décision, les flux d'information (représentés par les flèches bleues) et les cadres de décision (représentés par les flèches noires). Un centre de décision regroupe un ensemble d'activités décisionnelles appartenant à un même niveau horizon-période et remplissant une même fonction. Chaque centre de la grille reçoit un cadre

de décision d'un niveau supérieur ou d'un même niveau, et définit des cadres de décision pour les centres de décision de niveau inférieur ou de même niveau [Pie90], [Rob93], [Lat06].

Le réseau GRAI est une représentation locale dédiée au centre de décision. Il consiste à bien définir les activités, les états initiaux et les événements déclencheurs du processus décisionnel.

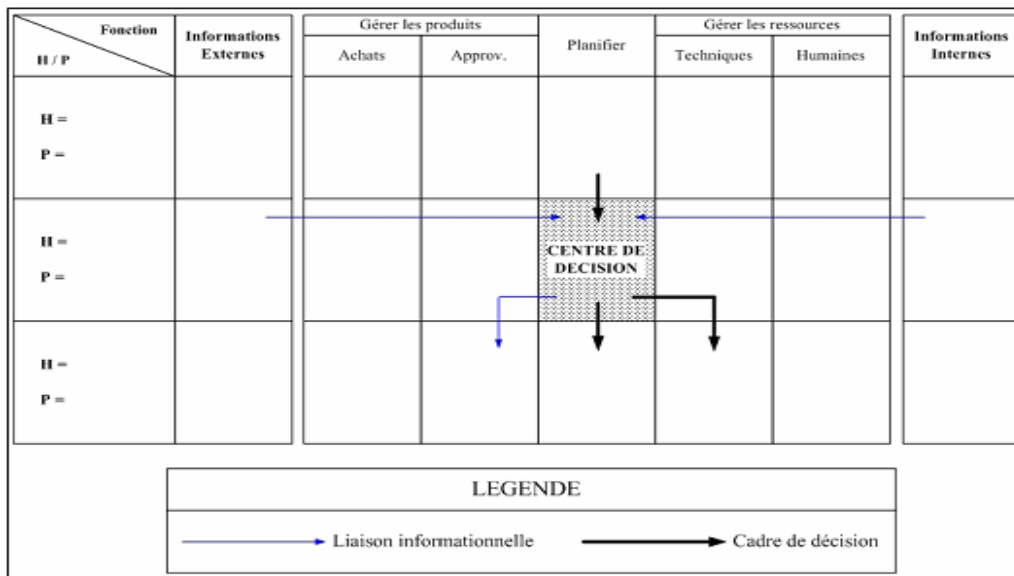


Figure 8 : Exemple de grille GRAI [Lat06]

En conclusion, certes la modélisation avec la grille GRAI ou le réseau GRAI est claire et facile à lire mais elle ne présente pas un intérêt pour notre étude car nous ne nous intéressons pas, à ce stade, à l'aspect décisionnel au niveau de notre modélisation.

2.2.2.1 Modélisation avec UML

UML (Unified Modeling Language) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant [BH96] : OMT (Object Modeling Technique) [RBPEL91], Booch [Boo91] et OOSE (Object Oriented Software Engineering) [JCJO92].

UML n'est pas une méthode dans la mesure où elle ne présente aucune démarche. A ce titre, c'est un formalisme de modélisation objet. A partir de 1997, il est devenu une norme de l'OMG (Object Management Group), ce qui lui a permis de s'imposer en tant qu'approche de développement objet et être reconnue et utilisée par de nombreuses entreprises. UML 2.0 définit treize types de diagrammes divisés en trois catégories (figure 9) :

- Diagramme de structure
- Diagramme d'interaction

- Diagramme de comportements

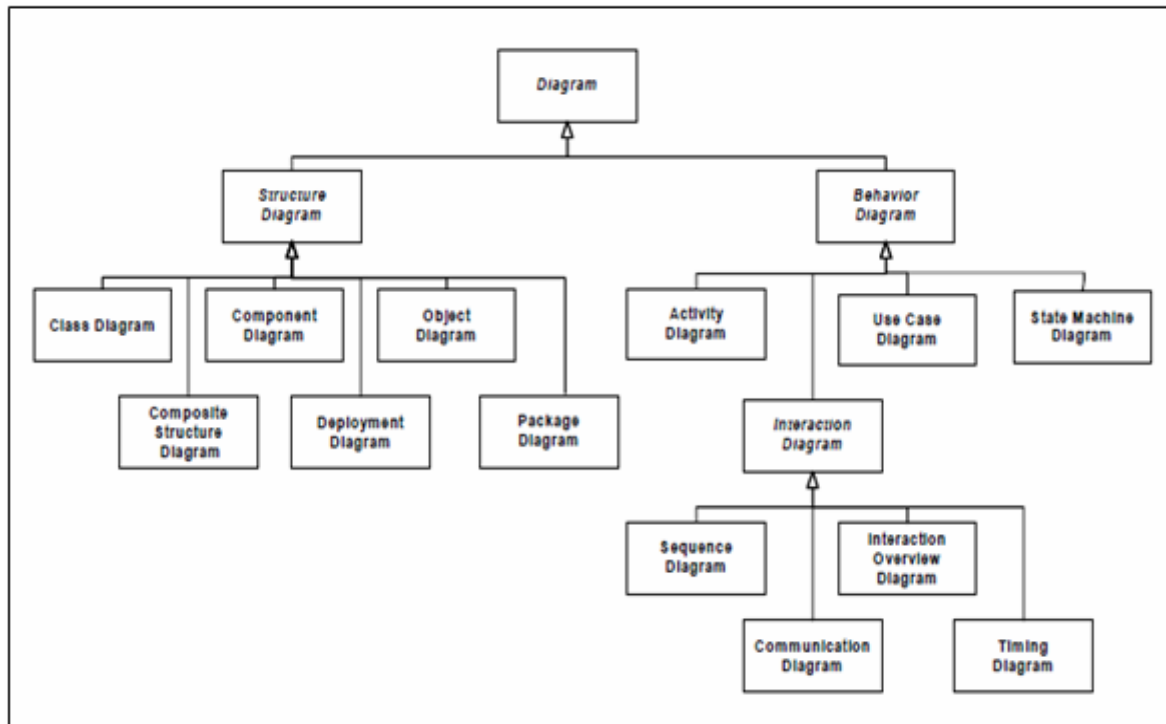


Figure 9 : Les diagrammes dans UML 2.0 [OMG05]

Dans le tableau suivant, nous présentons brièvement chacun de ces diagrammes.

DIAGRAMMES DE STRUCTURES	
Diagramme de classes	c'est un diagramme qui permet de présenter les classes et les interfaces d'un système ainsi que les différentes relations entre celles-ci. Chaque classe contient un nom, des attributs et des méthodes. Ce diagramme est statique car il fait abstraction des aspects temporels et dynamiques.
Diagramme d'objets	c'est un diagramme qui permet de représenter les instances des classes et les relations entre elles.
Diagramme de déploiement	c'est un diagramme de vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que la relation entre eux.
Diagramme de composants	ce diagramme permet de décrire l'architecture statique et physique d'une application en termes de modules (fichiers sources, bibliothèques, exécutables,...).

Diagrammes de paquetages	ce diagramme sert à représenter le regroupement des classes.
Diagramme de structures composites	ce diagramme représente un lien entre le diagramme de classes et le diagramme de composants. Il sert à montrer comment plusieurs éléments interconnectés, représentant des instances exécutables, collaborent à travers des liens de communication (connecteurs) pour atteindre des objectifs communs.
DIAGRAMMES DE COMPORTEMENTS	
Diagramme des cas d'utilisation	ce diagramme permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Ainsi, les cas d'utilisations permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. L'ensemble de cas d'utilisations décrit le comportement du système du point de vue de l'utilisateur.
Diagramme d'états-transitions	ce diagramme sert à représenter des automates d'états finis, sous forme de graphes d'états reliés par des arcs orientés qui décrivent les transitions. Il permet de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets, composants ou acteurs.
Diagramme d'activités	ce diagramme sert à décrire un cas d'utilisation. Il permet également de comprendre les liens entre les différentes fonctions.
DIAGRAMMES D'INTERACTION	
Diagramme de séquences	ce diagramme est une représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.
Diagramme de communication	ce diagramme est une représentation spatiale des objets, des liens et des interactions. Il ressemble au diagramme de séquence.
Diagramme de chronométrage	ce diagramme sert à montrer des interactions en tenant compte des aspects temporels. Il est adapté aux systèmes temps réel.
Diagramme d'interaction d'ensemble	ce diagramme est une version simplifiée des diagrammes d'activités qui se focalise sur les éléments qui interviennent dans la réalisation d'une activité.

Tableau 1 : Diagrammes UML 2.0 [OMG05]

Pour conclure, nous pensons que, parmi tous ces diagrammes, le diagramme d'activités semble le plus approprié à notre problème de modélisation du processus d'achat et de vente des produits dans le magasin car il tient compte de la synchronisation des activités et est riche en informations.

2.2.2.2 Modélisation avec BPMN

En 2004, le BPMI (Business Process Management Initiative) a publié la première version des spécifications de BPMN [BPMN06]. Ensuite en 2005, il a allié ses forces avec l'OMG (Object Management Group) pour créer une « Task Force » concentrée sur le BMI (Business Modeling & Integration) qu'ils ont appelée BPMN (Business Process Modeling Notation). Il s'agit d'un diagramme qui permet de représenter un processus métier en séparant les informations métiers des informations techniques. Ce diagramme semble très proche du diagramme d'activités d'UML mais l'atout de BPMN est de permettre de passer rapidement de la théorie à la pratique. Ainsi, une fois les processus modélisés par les utilisateurs métiers, il est possible de générer automatiquement, et de manière standard, le processus BPEL (Business Process Execution Language) à exécuter par le moteur du processus [Whi09], [OMG09]. Pour cela, il faut se renseigner sur un certain nombre d'informations techniques : le format des messages échangés, les protocoles de transport utilisés, les applications impliquées dans le processus par le biais de leurs connecteurs, les transformations de données effectuées, l'intégration des utilisateurs comme participants au processus... Un processus exécutable, une fois modélisé, peut être déployé sur un moteur BPMS (Business Process Management System) qui gère alors le contrôle de son exécution [BPMN06], [OMG09]. L'aspect d'exécution du processus ne sera pas développé dans notre travail puisque l'objectif de notre travail est de modéliser le processus d'achat et de vente des produits dans un magasin pour mettre en place l'application correspondante plus tard.

La nouvelle version des spécifications de BPMN, apparue en 2009, définit les éléments de base composant un diagramme de processus métiers [OMG09]. Ces éléments sont au nombre de quatre :

✓ *Flow Objects* :

Ce sont les principaux éléments qui permettent de décrire le comportement du processus métier. Il existe trois types de « Flow Objects » représentés dans le tableau 2.



Eléments	Description	Notation																																								
Activity	Une activité représente une action à réaliser. Elle peut être simple et donc représentée comme une tâche (Task) ou complexe et donc représentée comme sous-processus (Sub-process).	<p>Task</p>  <p>Sub-Process</p> 																																								
Event	<p>Un événement est ce qui peut arriver au cours de l'exécution d'une activité et influencer son déroulement. Il existe 3 catégories d'événement : initial, intermédiaire et final.</p> <p>Pour chaque catégorie, on trouve plusieurs types : un message, un « timer », une exception, une règle...</p>	<table border="0"> <thead> <tr> <th></th> <th>Start</th> <th>Intermediate</th> <th>Final</th> </tr> </thead> <tbody> <tr> <td>Message</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Timer</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Error</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Cancel</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Compensation</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Rule</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Link</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Multiple</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Terminate</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Start	Intermediate	Final	Message				Timer				Error				Cancel				Compensation				Rule				Link				Multiple				Terminate			
	Start	Intermediate	Final																																							
Message																																										
Timer																																										
Error																																										
Cancel																																										
Compensation																																										
Rule																																										
Link																																										
Multiple																																										
Terminate																																										
Gateway	<p>Un « Gateway » est utilisé en cas de synchronisation et/ou de prise de décision dans le processus où les flux convergent ou divergent en une ou plusieurs branches. Il existe plusieurs types de Gateway :</p> <ul style="list-style-type: none"> - XOR : en cas de décision et de fusion exclusive. - OR : en cas de décision et de fusion exclusive - Complexe : en cas de conditions et de situations complexe (plus de 3) - AND : en cas de jonction 	<p>Exclusive (XOR) Data-Based </p> <p>Event-Based </p> <p>Inclusive (OR) </p> <p>Complex </p> <p>Parallel (AND) </p>																																								

Tableau 2 : Différents types de Flow Objects

✓ **Connecting Objects :**

Ils servent à relier les « Flow Objects ». Ils sont de trois types : « Sequence Flow », « Message Flow » et « Association ». Ces éléments sont présentés dans le tableau suivant :




Eléments	Description	Notation
Sequence Flow	Un « Sequence Flow » est utilisé pour montrer l'ordre dans lequel sont accomplies les activités dans un processus.	
Message Flow	Un « Message Flow » est utilisé pour montrer le flux des messages qui circulent entre deux participants. Dans BPMN, ces derniers sont représentés par deux pools dans le diagramme.	
Association	Une association est utilisée pour associer une information à un « Flow Object ».	

Tableau 3 : Différents types de Connecting Objects

✓ **Swimlanes :**

Ils permettent d'organiser les activités dans des catégories séparées afin de distinguer les différentes responsabilités des participants. Il existe deux types de « Swimlane » présentés par le tableau 4.

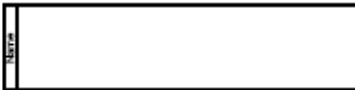
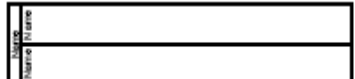
Eléments	Description	Notation
Pool	Un « pool » représente un participant ou une entité métier dans le processus.	
Lane	Une « lane » est une sous-division dans un pool. Elle est utilisée pour organiser les activités.	

Tableau 4 : Différents types de Swimlanes

✓ **Artifacts :**

Ils sont utilisés pour ajouter d'autres informations dans le processus. Trois types d'« Artifacts » sont proposés par la dernière version des spécifications de BPMN : Data Object, Group, Annotation.



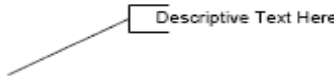
Eléments	Description	Notation
Data Object	Un « Data Object » donne des informations sur les manières dont les activités sont exécutées ou/et produits.	
Group	C'est un groupement d'activités qui sera utilisé pour une documentation ou une analyse.	
Annotation	Les « Text Annotations » permettent d'ajouter des informations pour que le lecteur puisse mieux comprendre le diagramme BPMN.	

Tableau 5 : Différents types d'Artifacts

Nous avons choisi de modéliser le processus d'achat et de vente des produits dans le magasin avec BPMN en utilisant l'outil Intalio Designer car, d'une part, un diagramme BPMN est clair et assure la synchronisation des tâches. D'autre part, c'est un standard qui nous permet de faire un mapping vers BPEL.

2.2.3 Cas Leroy Merlin

Une analyse de l'existant dans un magasin Leroy Merlin fera l'objet de cette partie. Ensuite, nous modélisons le processus d'achat et de vente des produits avec la méthode BPMN.

2.2.3.1 Analyse de l'existant

Nous avons fait des visites à l'un des magasins Leroy Merlin pour étudier son organisation, les activités de chaque responsable, la disposition des rayons, le mouvement des articles... Par ailleurs, nous avons eu l'occasion de tester la borne Keyshopping qui existe dans le magasin Leroy Merlin à Verquin, en 2009. Cette borne permet de rechercher des articles et de les localiser dans le magasin. Nous avons remarqué la lenteur du système, la complexité de son utilisation même pour des personnes sans handicap (noms de produits très techniques sans photos...) ainsi que l'absence d'entretien du système et surtout l'absence de communication et de formation. Le système n'était pas vraiment remarquable et peut être les clients ne savent même pas qu'il existe un système pour les aider à trouver leurs articles et il n'y a personne pour les aider à l'utiliser.

Après avoir fait plusieurs entretiens avec le personnel du magasin pour comprendre le processus d'achat et de vente des produits, nous avons modélisé ce processus dans le but d'analyser ce qui existe [HEB10a].

2.2.3.2 Modélisation du processus d'achat et de vente des produits

Nous avons décidé de représenter les acteurs par des « Pools » dans lesquels nous mettons les différentes activités qu'ils réalisent. La figure 11 représente le diagramme BPMN complet du processus d'achat et de vente des produits dans un magasin Leroy Merlin.

La figure 10 représente une partie de ce diagramme. Comme nous remarquons, notre processus débute par un cercle vide qui présente un événement initial.

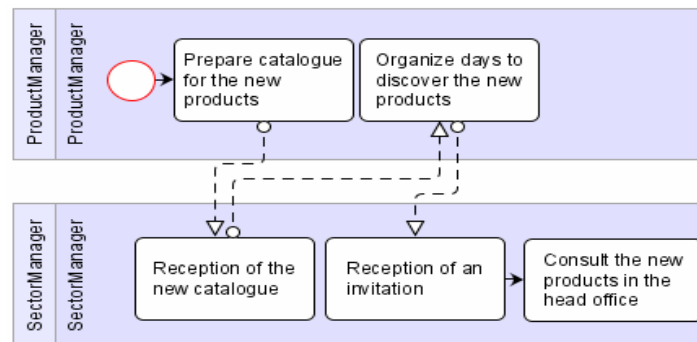


Figure 10 : Extrait du diagramme BPMN

Nous présentons ensuite chaque activité réalisée par un acteur dans le « Pool » correspondant. Les activités qui appartiennent au même « Pool » sont reliées par des « Sequence Flow » alors que celles qui ne se trouvent pas dans le même « Pool » sont reliées par des « Message flow ». Enfin, nous achevons le processus avec un événement final.

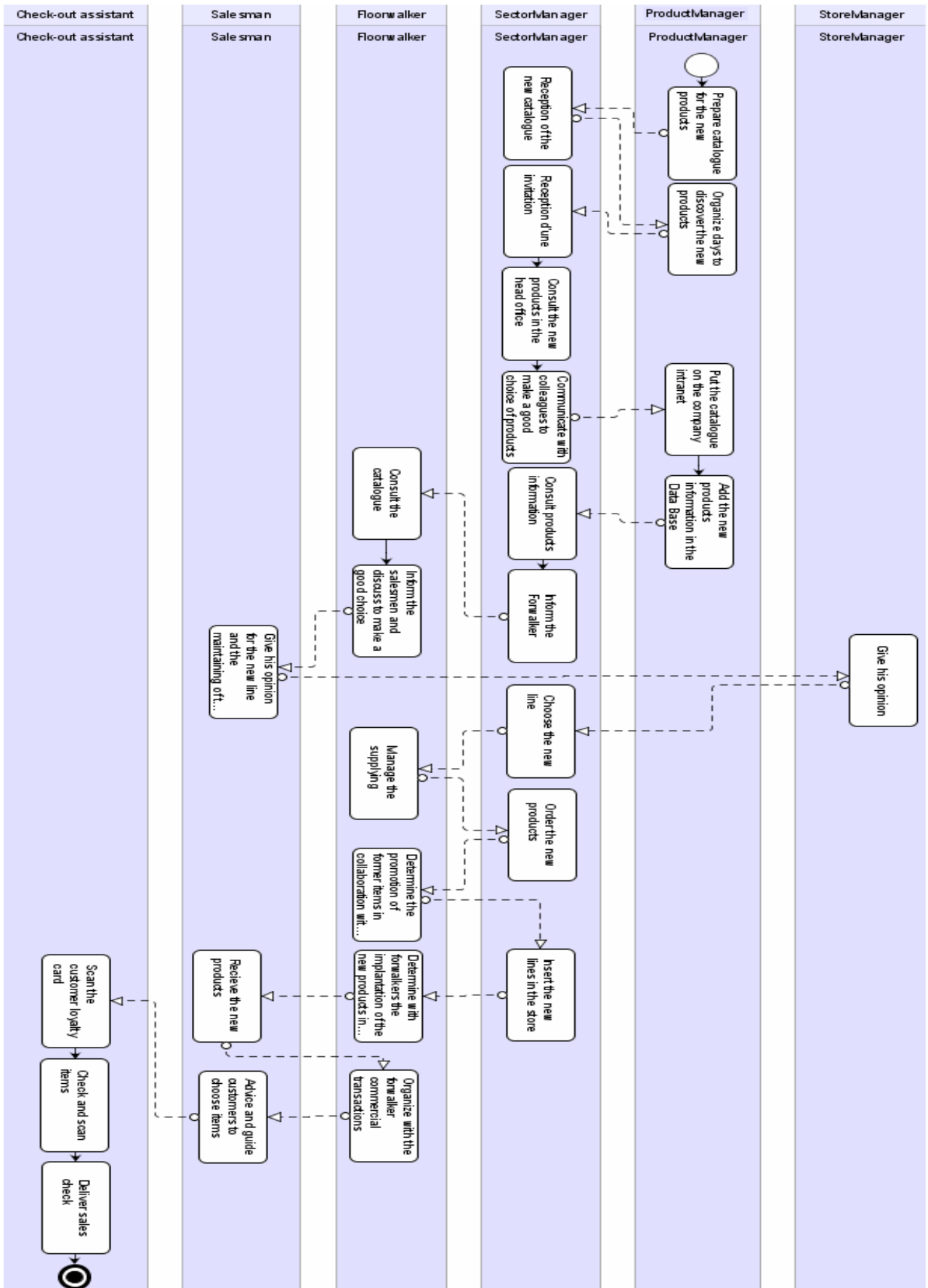


Figure 11 : Modélisation BPMN de la chaîne logistique d'achat et de vente de produits dans un magasin Leroy Merlin [HEB10a]

2.2.3.3 Avantages de la modélisation BPMN

Ce diagramme BPMN nous a permis d'avoir une idée claire sur les activités de chaque acteur ainsi que leurs interactions avec notre système. De plus, nous avons pu dégager des informations qu'on doit prendre en compte dans la conception de notre système tels que :

- L'absence d'un adressage par produit dans la base de données
- L'application du commerce de changement pour attirer les clients
- La difficulté de faire une mise à jour de l'emplacement des produits surtout pour un grand nombre de références de produits
- Le désordre des rayons causé par les clients.

Par ailleurs, le système i-GUIDE aidera les responsables des rayons à choisir le bon emplacement des produits en analysant les parcours des clients, leurs habitudes d'achat, leur hésitation face à un produit, l'impact d'une publicité poussé par le système sur leur comportement d'achat (mode interactif de type « push »), leur degré de fidélité... Une mine d'informations comportementales en sus des habitudes de consommation utilisée jusqu'alors par les grandes surfaces, pouvant servir dans la politique commerciale du magasin, dans ses décisions tactiques mais aussi stratégiques face à une concurrence de plus en plus féroce.

2.3 Analyse des exigences du système i-GUIDE

L'analyse des exigences d'un nouveau système connue, en anglais, sous le nom de « Requirements Analysis » est une étape très importante dans le cycle de vie d'un projet puisqu'elle représente le niveau de compréhension des besoins et des attentes du client par rapport au système proposé. Les exigences représentent une description du comportement d'un système, ses propriétés ou ses attributs [LK95], [Dou98]. En pratique, la méthode d'Analyse Orientée Objet est relativement la plus utilisée pour décrire les fonctionnalités demandées du système sous forme de modèle conceptuel [DP03], [LK95], [RBPEL91], [RJB05]. Dans [HEB10a], nous avons opté pour l'approche UML pour présenter les fonctionnalités du système. Nous avons utilisé le diagramme de cas d'utilisation UML pour présenter les fonctionnalités du système et les acteurs qui participent à son fonctionnement. Ensuite, nous avons détaillé la principale fonction de notre système et présenté son diagramme d'activité UML. Enfin, nous avons utilisé le diagramme de séquence UML pour montrer les objets nécessaires pour accomplir cette fonction.

2.3.1 Diagramme de Cas d'utilisation UML

L'objectif des cas d'utilisation UML est l'expression des besoins en termes de services que doit assurer le système. Les diagrammes de cas d'utilisation définissent deux concepts principaux : les acteurs et les cas d'utilisation [OMG05], [RJB05].

- Un acteur est une entité extérieure du système qui peut initier l'un de ses cas d'utilisation.
- Un cas d'utilisation est une fonctionnalité offerte par le système.

Dans la figure 12, nous présentons le diagramme de cas d'utilisation de notre système. La personne handicapée ou sans handicap représente l'acteur principal et les autres acteurs représentent les acteurs secondaires qui interagissent avec notre système intelligent de navigation. Les acteurs secondaires sont :

- L'automate qui comprend le système de localisation et l'application de guidage.
- Le responsable Marketing qui s'occupe de l'insertion des publicités et des promotions dans notre système.
- Le conseiller de vente pour aider le client à trouver son article recherché en cas de besoin.

De plus, à travers ce diagramme nous avons déterminé les fonctions de notre système que sont :

- Entrer le numéro de la carte de fidélité de l'utilisateur pour l'identifier
- Guider l'utilisateur en lui indiquant les directions et ceci nécessite la détermination de sa position actuelle ainsi que l'emplacement des produits dans le magasin
- Gérer les promotions et les publicités
- Donner des informations sur les articles recherchés (prix, ingrédients, guide d'utilisation...) et appeler un conseiller de vente pour des informations supplémentaires
- Donner l'addition pour éviter la queue à la caisse
- Ajouter des suggestions des utilisateurs pour une amélioration du service.

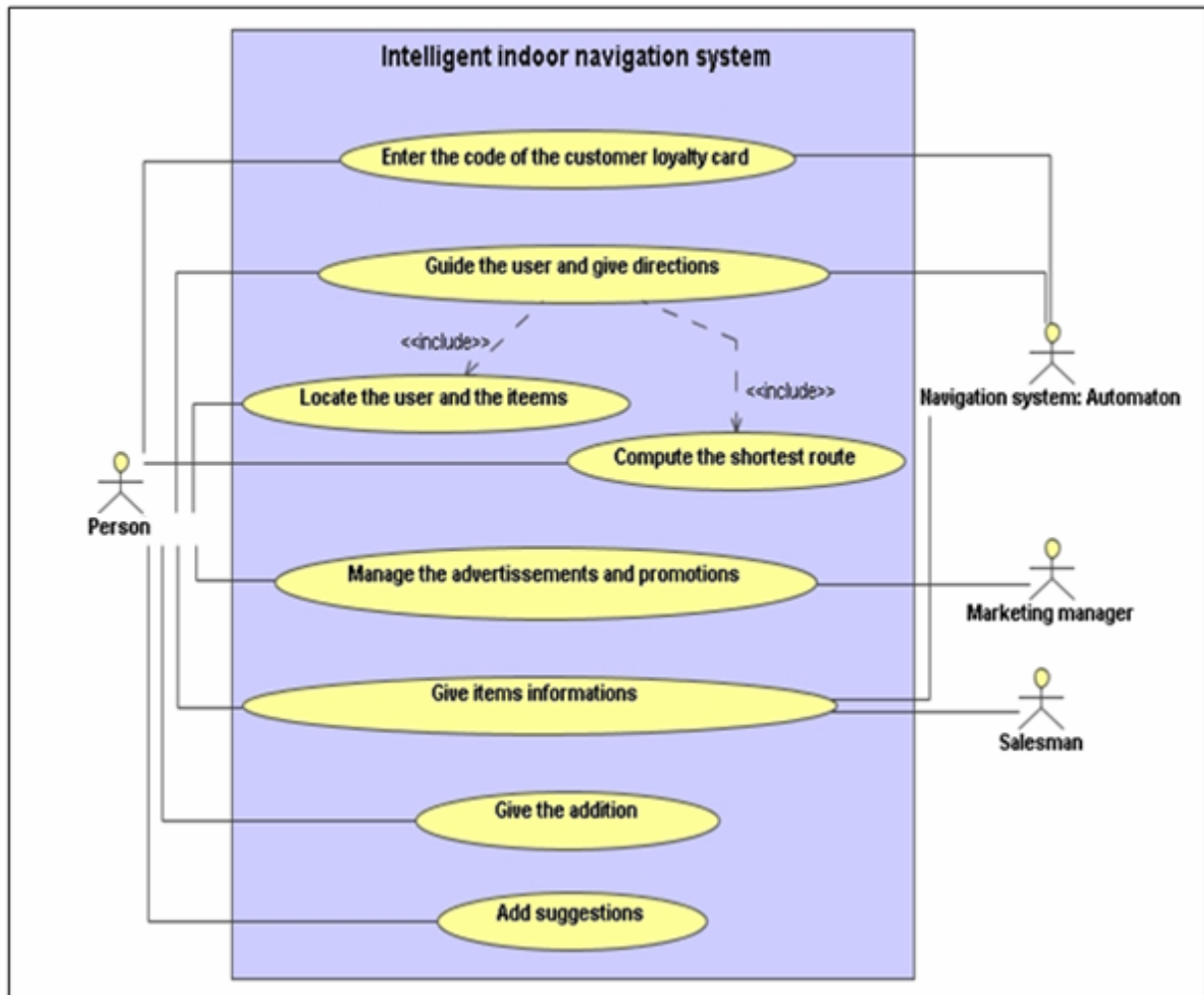


Figure 12 : Diagramme de Cas d'utilisation UML de notre système de guidage [HEB10a]

Dans la section suivante, nous détaillons la fonction principale « Guider l'utilisateur et lui donner des instructions » en utilisant le diagramme d'activités UML.

2.3.2 Diagramme d'activités UML

Les diagrammes d'activité permettent de représenter le déroulement d'un cas d'utilisation [OMG05], [RJB05].

La figure 13 présente le diagramme d'activités de la fonction « Guider l'utilisateur et lui donner des instructions ». Ce diagramme décrit les différentes étapes pour assister une personne, handicapée en particulier, en lui donnant l'opportunité d'être autonome et appeler un conseiller en cas de besoin. Pour cela, nous utilisons les technologies de synthèse et reconnaissance vocale afin de faciliter la communication entre l'utilisateur et le système.

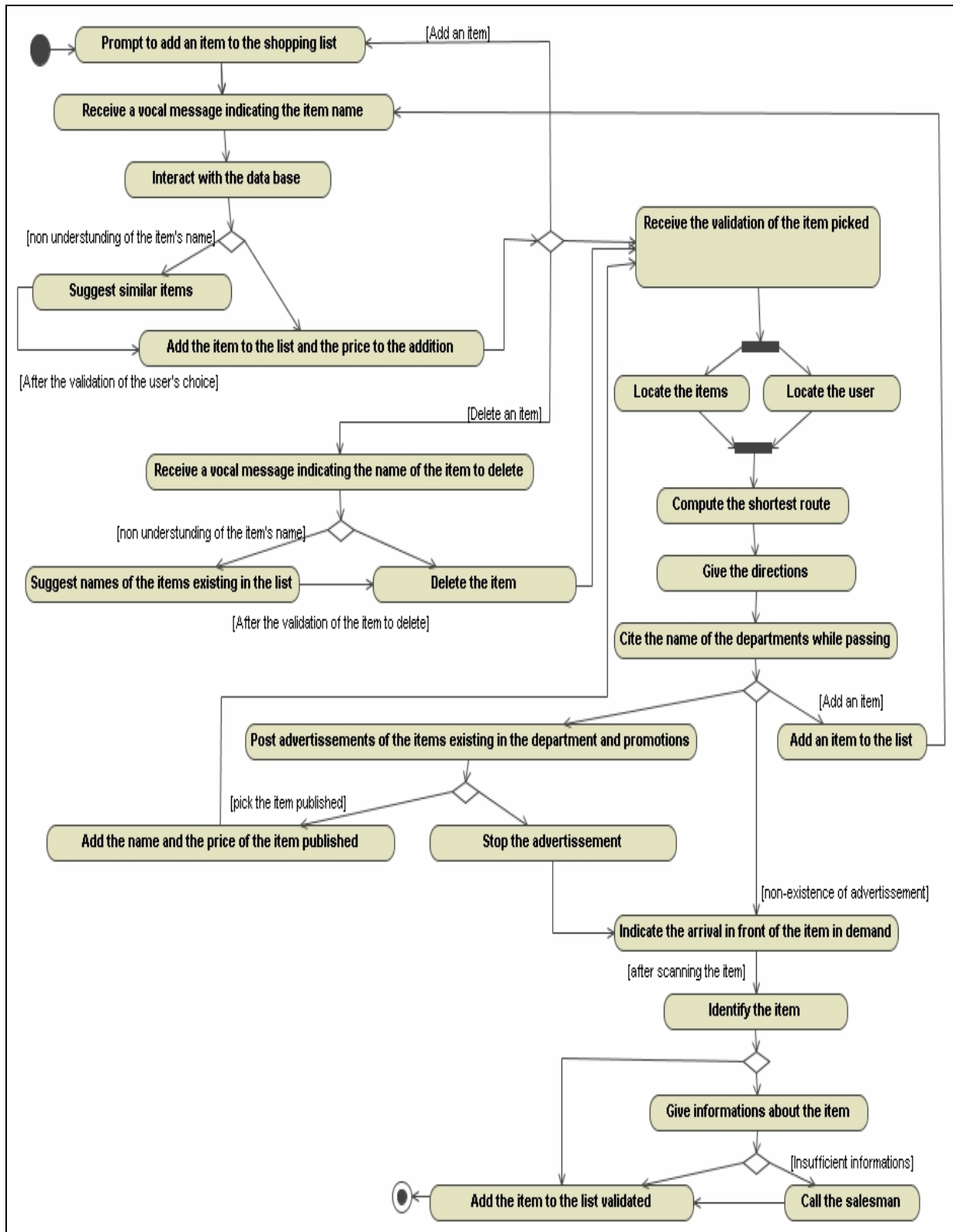


Figure 13 : Diagramme d'activité UML de la fonction « Guider l'utilisateur et lui donner des instructions » [HEB10a]

Le système demande à l'utilisateur d'ajouter un article à sa liste de courses. Ensuite, il reçoit le nom de l'article à ajouter à travers un message vocal. Il demande à la base de

données si son nom existe. Si c'est le cas, il l'ajoute à la liste de courses sinon il lui propose un article ayant un nom similaire et ainsi de suite jusqu'à finaliser sa liste de courses. L'utilisateur a aussi la possibilité de supprimer des articles ajoutés. Une fois la liste validée, le système peut commencer le calcul de l'itinéraire le plus court passant par tous les articles de la liste. Pour cela, il faut tout d'abord déterminer la position actuelle de l'utilisateur ainsi que l'emplacement des articles recherchés dans le magasin. Ensuite, le système commence à donner des indications à l'utilisateur via des écouteurs, par exemple, pour lui montrer le chemin à prendre.

Le système recalcule l'itinéraire suite à toute modification de la liste de courses ou à tout changement de direction. En passant par chaque rayon, le système indique son nom et poste les promotions en cours et des publicités sur des produits du rayon. Si l'utilisateur est intéressé par ces dernières ou s'il se rappelle d'un nouveau produit, il a la possibilité de l'ajouter à sa liste. L'utilisateur peut arrêter la publicité à tout moment. Le système informe l'utilisateur de son arrivée devant l'article recherché et lui demande de le prendre et le scanner. Une fois l'article vérifié, l'utilisateur peut demander au système de lui donner des informations sur le produit telles que son prix, ses ingrédients, son guide d'utilisation... En cas où il trouve que ces informations ne sont pas suffisantes ou il ne trouve pas le bon produit ou encore il n'arrive pas à ramasser l'article (cas des handicapés moteurs), il peut appeler un conseiller qui viendra à son aide. Si l'article est validé, il l'ajoute à la liste des articles validés et son prix sera ajouté à l'addition ce qui lui permettra de payer ses produits directement dans une caisse rapide sans les décharger et les recharger.

Dans la section suivante, nous nous concentrons sur la fonction de calcul du chemin le plus court à parcourir pour trouver tous les articles existants dans la liste des courses.

2.3.3 Diagramme de séquences UML

Nous utilisons le diagramme de séquence UML pour montrer les objets nécessaires pour accomplir la fonction de calcul du chemin le plus court (figure 14).

Ces objets sont :

- Les capteurs pour déterminer la position de l'utilisateur à un instant donné
- La mémoire pour enregistrer les noms et références des articles choisis par l'utilisateur
- La base de données pour donner la localisation des articles dans le magasin et les distances entre eux

- L'algorithme d'optimisation qui nécessite toutes les informations précédentes pour calculer le chemin le plus court.

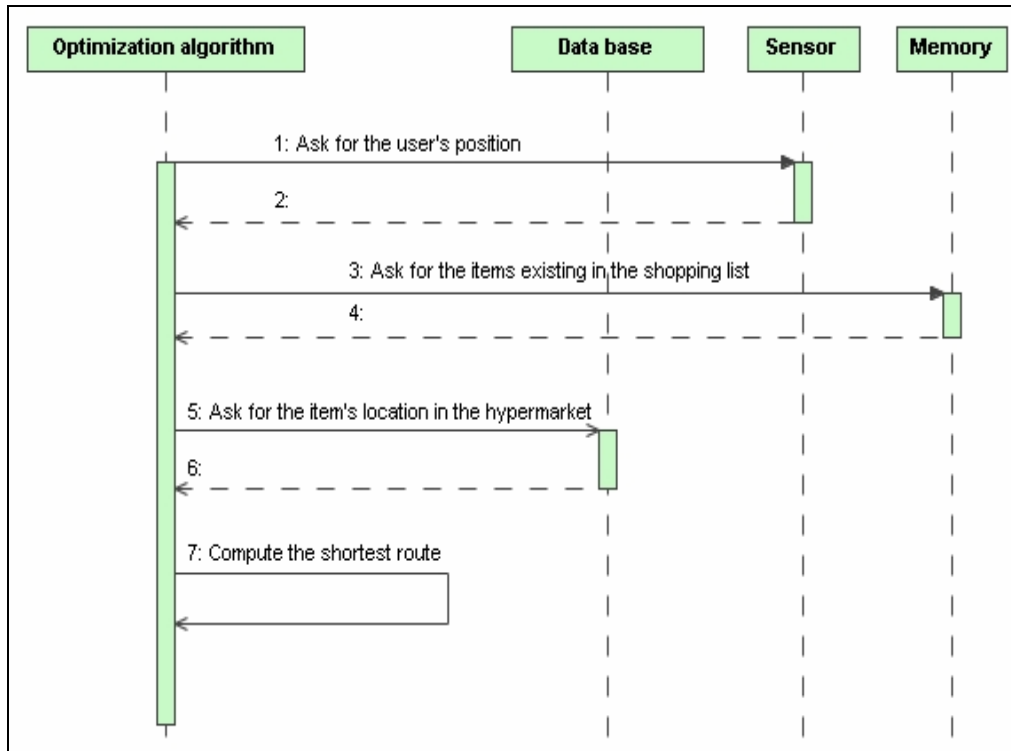


Figure 14 : Diagramme de séquence de la fonction calcul du chemin le plus court [HEB10a]

La mémoire doit être suffisante pour stocker les noms/références et les prix des articles et la base de données doit être mise à jour suite à toute modification pour donner la bonne position. En plus, la technologie de localisation doit être la plus précise possible pour localiser le client et l'assister tout au long de son parcours et c'est sur cette partie que nous allons nous concentrer dans le chapitre 3.

2.3.4 Avantages du système i-GUIDE

Nous présentons, dans la suite, les avantages du système i-GUIDE pour le client et pour le magasin.

2.3.4.1 Avantages pour le client

Le système i-GUIDE est un système intelligent, autonome et innovant dans le contexte du u-commerce représentant un assistant personnel au client dans une démarche citoyenne et de nouveaux services à la personne, lui permettant de :

- préparer sa liste de courses en magasin ou chez lui (via le scan 2D qui se généralise de plus en plus sur le nouveau type de prospectus), la sauvegarder sur son compte et le cas échéant l'éditer à tout moment
- gagner du temps sur les achats besoins pour dégager du temps pour les achats plaisirs
- suivre le client et le guider pour ne pas se perdre dans le magasin grâce à sa technologie de localisation (géo-localisation) surtout si la personne n'est pas habituée au magasin
- déterminer le chemin le plus court à parcourir grâce à son calculateur d'itinéraire
- éditer sa liste de courses à tout moment et lui indiquez le nouveau chemin à prendre
- ajouter le prix du produit à l'addition à chaque fois où il le scanne ceci l'évitera de faire la queue en caisse...

De plus, c'est un système de guidage multi-canal. Ainsi, nous envisageons, dans le cadre des travaux en cours de réalisation au sein de notre équipe de recherche, de concevoir et développer un robot de type humanoïde qui ferait les courses à la place du client qui peut se décharger de cette corvée et rester à la cafète prendre une boisson et lire un journal. De cette manière, l'hypermarché devient un espace de détente contrairement à ce qu'il l'est actuellement surtout pour les « hommes ».

2.3.4.2 Avantages pour le magasin

Le système i-GUIDE est une source d'informations et un moyen de communication entre le magasin et ses clients. C'est une nouvelle forme de dématérialisation de la relation client puisqu'elle permet de :

- suivre l'évolution de ses achats à travers les chemins parcourus
- booster ses achats en proposant des produits complémentaires aux produits déjà ajoutés à la liste et en diffusant des promotions personnalisées.
- analyser ses habitudes d'achats et surtout tracer son comportement d'achat (produit supprimé, produit devant lequel il s'est arrêté longtemps pour comprendre et chercher les raisons de son hésitation voire lui suggérer un crédit en mode « push » ou tout autre type de financement échelonné ...)

Par ailleurs, le système i-GUIDE représente un outil de multi-merchandising pour segmenter l'offre par besoin client (par univers, événements, profils clients,...). Il peut même représenter un guide d'achats interactifs pour guider les clients dans leurs achats à travers quelques questions simples. De plus, avoir un système innovant de ce type dans le magasin va

attirer des clients pour le découvrir et même s’amuser en l’essayant. Sans parler de l’impact de ce genre de service sur l’image de marque de l’enseigne du fait qu’il s’inscrit dans une démarche citoyenne et de nouveaux services d’aide à la personne : la clientèle appréciera cette démarche altruiste qui, tout en rompant avec l’image habituelle qui colle aux grandes surfaces comme « des machines à faire du fric », permettrait néanmoins à l’enseigne de trouver de nouveaux modes d’optimisation de sa chaîne logistique et par conséquent de ses profits tout en mettant le client au centre de ses préoccupations.

2.4. Conclusion

Dans ce chapitre, nous avons modélisé la chaîne logistique du magasin Leroy Merlin en amont et en aval en utilisant la méthode BPMN, un standard de l’OMG. Le diagramme BPMN nous a permis d’avoir une idée claire sur les activités de chaque acteur ainsi que leurs interactions avec notre système et de dégager ainsi les informations importantes à prendre en compte dans la conception du système. Par ailleurs, des diagrammes UML ont été utilisés pour étudier les interactions entre notre système temps réel et son environnement extérieur. Ainsi, nous avons utilisé le diagramme des cas d’utilisation UML pour représenter les principales fonctionnalités du système et les acteurs qui participent à son fonctionnement. Ensuite, nous avons énuméré les étapes pour accomplir la fonction principale du système « Guider l’utilisateur et lui donner des instructions » à travers le diagramme d’activités UML. Enfin, nous avons utilisé le diagramme de séquence UML pour présenter les objets nécessaires (technologie de localisation, base de données facile à mettre à jour, algorithme d’optimisation...) pour accomplir la fonction de calcul du chemin le plus court. Enfin, nous avons décrit les avantages d’un système nomade de guidage pour le client et le magasin.

Le système i-GUIDE ne peut être fonctionnel sans une technologie de localisation et un programme d’optimisation de parcours. Nous étudions, dans le chapitre 3, les nouvelles technologies de localisation et proposons la technologie la plus appropriée à notre problème. Des approches d’optimisation de parcours feront l’objet du chapitre 4.

CHAPITRE 3 : TECHNOLOGIES DE LOCALISATION DANS UN ESPACE FERME

3.1. Introduction

De nos jours, les personnes demandent de plus en plus de services offrant des informations de localisation pour faciliter leurs déplacements à l'extérieur et à l'intérieur des bâtiments. Le système GPS est, généralement, le plus utilisé pour se déplacer dans les rues. Cependant, ce système permet d'atteindre des précisions de l'ordre de quelques mètres et reste peu fiable voire non fonctionnel à l'intérieur des bâtiments (centre commercial, aéroport, métro, musée, magasin...) à cause des obstacles pour détecter les signaux d'où l'apparition d'autres technologies de localisation. Ces dernières seront présentées dans la première section de ce chapitre. Ensuite, nous choisissons la technologie qui permettra de déterminer la position actuelle de l'utilisateur dans le magasin. Ce choix dépendra, essentiellement, de l'infrastructure existante, de la précision nécessaire et du coût de la solution. A la fin du chapitre, nous proposons une application mobile sous Android pour le guidage des personnes dans un hypermarché.

3.2. Technologies de localisation dans un espace fermé

Il existe plusieurs technologies de localisation à l'intérieur des bâtiments. Parmi ces technologies, on peut citer le Bluetooth, le Wifi, le GPS indoor et la RFID. Une description brève de ces technologies de localisation fera l'objet de cette section.

3.2.1 La technologie Bluetooth

Le Bluetooth est une technologie qui permet la communication sans fil entre différents appareils tels que les téléphones portables, les oreillettes, les casques sans fil, les ordinateurs portables... Il assure l'envoi de tout type de données et en particulier des communications de la voix en temps réel. Le Bluetooth utilise des ondes radio avec une bande de fréquence de 2.4 GHz et sa portée est limitée à 100 mètres et diminue suivant les obstacles rencontrés.

Par ailleurs, la technologie Bluetooth a été utilisée pour la localisation d'un mobile dans un espace fermé à travers un réseau de balises Bluetooth. C'est cette technologie que le système BlueEyes, cité au premier chapitre, a utilisée pour la localisation d'une personne malvoyante dans les couloirs du métro. Ainsi, l'utilisateur du système est détecté à chaque fois qu'il passe dans la zone de rayonnement (entre 3 et 10 mètres) des balises. Ce système permet à l'utilisateur de se géolocaliser à l'aide de son appareil mobile et lui indique les directions à prendre.

Il est vrai que cette technologie paraît peu coûteuse et pratique puisque la plupart voire tous les téléphones portables existants actuellement sur le marché possèdent la technologie Bluetooth. Cependant, elle n'est pas intéressante pour notre cas d'étude car elle peut détecter plusieurs balises Bluetooth en même temps puisque dans une gondole les repères qui se trouvent en parallèles sont espacés de 1,5 mètres au minimum. De plus, des interférences avec les gondoles métalliques sont possibles et conduiront à une distorsion voire brouillage du signal Bluetooth émis.

3.2.2 La technologie RFID

La technologie RFID ou Radio Frequency IDentification est une technologie qui permet d'identifier un objet, suivre son cheminement et connaître ses caractéristiques à distance grâce à une étiquette émettant des ondes radio, attachée ou incorporée à l'objet. Cette technologie permet la lecture des étiquettes et peut traverser de fines couches de matériaux. Un tag ou étiquette RFID se compose d'une puce stockant les données reliées à une antenne. Cette étiquette est lue par un lecteur qui permet de capter et de transmettre les informations.

Il existe deux grandes familles d'étiquettes RFID :

- **Les étiquettes actives** : elles sont équipées d'une batterie leur permettant d'émettre un signal. Les étiquettes actives possèdent une meilleure portée mais à un coût plus élevé et avec une durée de vie plus restreinte.
- **Les étiquettes passives** : elles utilisent l'énergie propagée à courte distance par le signal radio de l'émetteur. Ces étiquettes à moindre coût sont généralement plus petites et possèdent une durée de vie quasi-illimitée. En contrepartie, elles nécessitent une quantité d'énergie non négligeable de la part du lecteur pour pouvoir fonctionner.

La technologie RFID est beaucoup utilisée, en particulier en logistique, dans plusieurs applications dont on peut citer : le suivi des colis, l'identification des conteneurs, la gestion des stocks en particulier en e-commerce, la gestion de production [PKPLC10], [RPAVTDD06]...

3.2.3 La technologie WiFi

Le WiFi (Wireless Fidelity) est une technologie de communication sans fil permettant de relier des ordinateurs portables, des ordinateurs de bureau, des assistants personnels (PDA) ou tout type de périphérique à une liaison haut débit sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) à plusieurs centaines de mètres en environnement ouvert.

Par ailleurs, le WiFi a été utilisé pour localiser des personnes ou des objets. La géolocalisation par WiFi est réalisée par triangulation, en corrélant la force ou le délai de transit des signaux entre l'objet à localiser et les bornes WiFi aux alentours. Des applications ont été réalisées par la société Ekahau utilisant des tags Wifi, représentés sous forme d'un bracelet, pour localiser les malades dans un hôpital, les enfants dans un parc d'attraction. Cependant, la précision de la localisation des personnes, à l'aide du Wifi, reste de l'ordre d'un mètre. De plus, du fait que les grandes surfaces, en particulier Leroy Merlin, ont leur propre réseau WiFi pour la gestion du backoffice, une contrainte imposée par l'entreprise partenaire est de ne pas installer de nouveaux réseaux WiFi pour éviter toute interférence ou malveillance avec leur réseau maître. Ceci explique, entre-autres, l'abandon de cette solution.

3.2.4 Le système de GPS indoor

Le système de GPS indoor a été utilisée par Insiteo pour la localisation des personnes dans un espace fermé tels que les musées, les centres commerciaux, les aéroports... Cette technologie combine le Wifi et le GPS pour donner l'itinéraire entre deux points. Insiteo a développé pour l'opérateur Orange une application permettant aux visiteurs du Salon de l'Automobile de géopositionner les stands et d'afficher l'itinéraire entre deux points.

Certes, cette technologie semble intéressante mais elle sert à la localisation point à point et n'est pas assez précise et donc elle ne répond pas à notre besoin.

3.2.5 La technologie NFC

La technologie NFC (Near Field Communication) ou encore technologie de communication de proximité, lancée par Sony et Philips, permet d'échanger des données entre un lecteur et n'importe quel terminal mobile ou entre les terminaux eux-mêmes et ce, à un débit maximum de 424 Kbits/s.

A l'instar de la technologie RFID qui utilise des marqueurs passifs, les communications NFC s'appuient sur le lecteur pour transmettre la puissance à la puce NFC par radiofréquence.

Les utilisations de cette technologie sont multiples. En effet, elle est utilisée pour : connecter un téléphone portable à un ordinateur pour télécharger un fichier, envoyer des images d'un appareil photo à un PDA, échanger des données entre un téléviseur et un smartphone... Au-delà, ce sont les marchés du contrôle d'accès, de la billetterie ou des bornes interactives de tout type qui sont concernés, mais aussi les achats par simple approche d'un terminal pourront avoir lieu...

Il est vrai que la technologie NFC est la technologie de l'avenir qui facilitera la vie des consommateurs mais la distance de communication maximale de cette technologie (maximum 10 cm) reste faible pour la localisation de l'utilisateur. Cependant, elle pourrait être intéressante dans le futur si cette distance est nettement améliorée.

3.3. Proposition d'un système de localisation dans un hypermarché

Après une revue des technologies de localisation indoor, nous présentons, dans cette section, l'architecture du système i-GUIDE proposé ainsi que le prototype réalisé.

3.3.1 Présentation de l'architecture du système i-GUIDE

Compte tenue du fait que la localisation d'une personne dans un bâtiment s'effectue dans des environnements moins favorables que le GPS, les réseaux locaux, basés sur par exemples les tags RFID, les tags Bluetooth ou encore le WiFi, peuvent être exploités pour prendre le relais du système GPS et permettent ainsi la localisation d'une personne dans un rayon limité pouvant aller jusqu'à quelques dizaines de mètres. Cependant certaines de ces technologies imposent des contraintes quant à leur déploiement. Une grande densité

d'équipements est nécessaire pour fournir en permanence la localisation tout en conservant des performances raisonnables de l'ordre du mètre.

Etant donné que la plupart des magasins et en particulier les magasins Leroy Merlin disposent du réseau WiFi, nous avons décidé de profiter de ce réseau existant et donc de proposer une solution qui lui est compatible mais sans l'altérer ni lui apporter aucune modification et/ou ajout comme l'exige le partenaire. Ainsi, nous proposons de faire un couplage du réseau WiFi existant et de la technologie RFID. Nous avons choisi la technologie RFID (Radio Frequency IDentification) pour localiser l'utilisateur dans l'hypermarché et détecter sa position quand il change de direction. Nous utilisons le Wifi installé dans les magasins Leroy Merlin pour assurer la communication entre la base de données et la technologie RFID et le standard ODBC (Open Database Connectivity) pour la communication entre la base de données et l'algorithme d'optimisation. La figure 15 présente l'architecture globale du système.

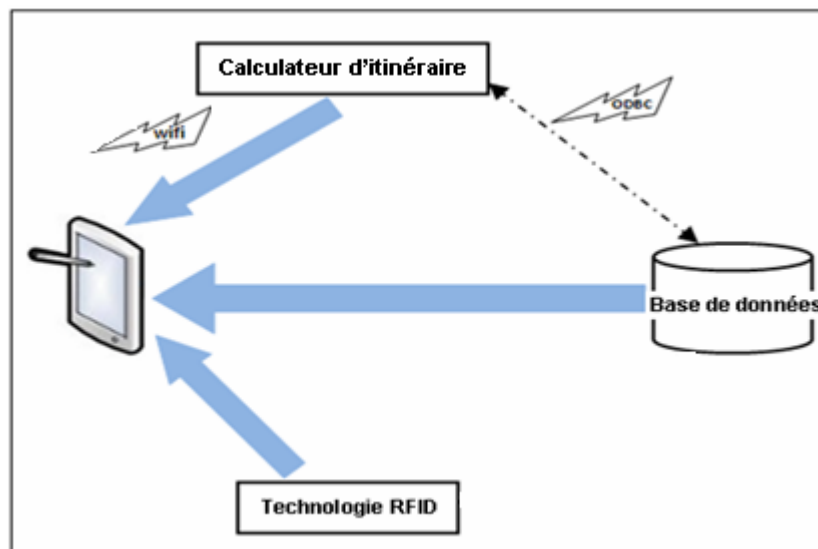


Figure 15 : Architecture globale du système de navigation temps réel

3.3.2 Présentation du prototype réalisé

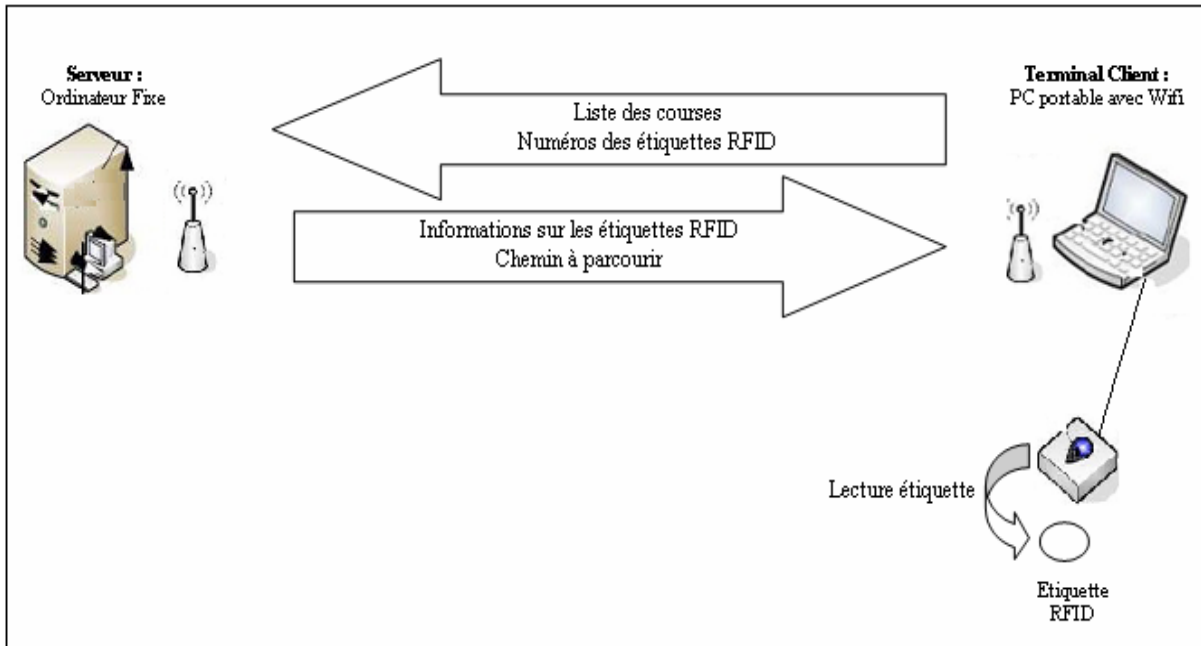


Figure 16 : Prototype de la solution

La figure 16 représente le prototype réalisé dans le laboratoire. Il est composé de trois parties :

- **les étiquettes RFID** : Nous utilisons des étiquettes passives. Chaque étiquette RFID contient une information de 12 octets: un octet de début de tag, un identifiant ou tag constitué de 10 octets et donc de 10 nombres codés en hexadécimal et un octet marquant la fin du tag. Comme le montre la figure 17, ces étiquettes sont posées sur les repères existants sur chaque gondole. Nous avons associé chaque étiquette RFID à un repère du magasin au sein de la base de données.



Figure 17 : Disposition des repères sur les gondoles

Le magasin est ainsi cartographié par les étiquettes RFID ce qui permet de localiser le client lors de son déplacement et lui envoyer des messages. Le choix du maillage à mettre en place dépendra essentiellement de la précision recherchée et du budget de l'entreprise. Dans notre cas, nous avons choisi le critère de précision et nous avons proposé une distance qui varie entre 2 et 3 mètres selon les dimensions des produits.

- **une partie « client » embarquée** : Elle représente le terminal utilisateur. Nous avons utilisé un ordinateur portable pour représenter cette partie. Cet ordinateur est équipé d'une connexion WiFi et d'un lecteur RFID. Sur l'ordinateur portable, un programme réalisé en Python permet la gestion des informations arrivant sur le port USB et l'extraction du tag lorsque le lecteur RFID passe au dessus d'une étiquette. Celui-ci envoie le tag détecté vers le serveur. Tant que l'identifiant détecté ne change pas, aucune information n'est envoyée. Chaque client est identifié par un nom d'utilisateur.
- **une partie « serveur »** : Elle est fixe et permet de recevoir et traiter des informations. Elle contient également la base de données relative au magasin et l'algorithme d'optimisation. Cette partie est représentée par un ordinateur de bureau équipé d'une connexion WiFi. Le serveur a été également développé en Python. Il communique avec la base de données Access à l'aide des requêtes SQL. Ainsi il connaît, pour un client donné, le chemin que celui-ci doit emprunter. Autrement dit, il doit vérifier que la liste des tags détectés suit l'ordre donné par l'algorithme d'optimisation. Ainsi, il compare les changements de tags avec ceux du chemin optimal théorique et envoie, au client, vers quel tag il doit se déplacer. Le serveur communique également au client qu'il a des articles à prendre quand il arrive dans une zone contenant des produits demandés par le client. Le client et le serveur communiquent à travers le Wi-Fi installé dans le magasin.

Par ailleurs, dans le cas où l'utilisateur ne trouve pas d'étiquette RFID, le système lui demande d'avancer et de s'approcher plus des étagères existantes dans les rayons.

Après avoir testé ce prototype et dans une politique de u-commerce, nous avons décidé d'embarquer notre application sur un terminal portable plus pratique et adapté tel que les Smartphones, les iPhone, les PDA... et adapter alors l'architecture de notre système. Une étude de faisabilité de l'application et l'interopérabilité entre les différentes parties du système font l'objet de la section suivante.

3.4. Présentation de l'application mobile

Dans cette partie, nous proposons une application sous Android pour le guidage des personnes dans un hypermarché. Pour ce faire, nous commençons par la présentation du système d'exploitation Android. Ensuite, nous testons deux architectures différentes pour le système de guidage. Enfin, sur la base des résultats de ces tests, nous proposons une architecture que nous envisageons d'implémenter ultérieurement. Pour la localisation du client, nous proposons, d'inclure dans le futur, dans le Smartphone, une carte SD de lecture de puces RFID. Cependant, actuellement, la portée de cette carte ne dépasse pas 10 centimètres [HKB11c], ce qui explique les contraintes et limitations auxquelles nous sommes encore confrontés, le domaine de l'intelligence ambiante étant en ébullition et en constante évolution chaque jour ce qui justifie que nous avons opté pour une veille technologique permanente tout au cours de nos travaux de recherche.

3.4.1 Présentation du système d'exploitation Android

Android est un système d'exploitation open-source développé par Open Handset Alliance pour Google. Ce système est installé aujourd'hui sur les Smartphones, mais aussi les tablettes tactiles ou d'autres systèmes tels que les téléviseurs [Gar09].

Android a été conçu pour intégrer au mieux des applications existantes de Google comme le service de courrier Gmail, ou celui de cartographie, Google Maps, ou encore Google Agenda, Google Talk, ... Un accent particulier est mis sur la géolocalisation avec Google Latitude et la météo de la ville la plus proche disponible sur le menu principal.

Le langage de programmation associé est le JAVA. Les mises à jour de ce système d'exploitation sont assez fréquentes. La version la plus récente est la version 3.0, sortie en février 2011, mais la version la plus courante sur le marché actuellement est la version 2.2 pour Smartphones [Gar09], [GCR10].

Outre sa portabilité et sa proximité avec les applications Google, Android prend en charge les capteurs et modules dont sont maintenant équipés les téléphones le plus récents tels que le système GPS, les accéléromètres, la technologie NFC... qui peuvent être très utiles pour plusieurs applications. De plus, il se positionne comme un système d'exploitation et un environnement de développement dédié aux appareils mobiles et indépendant de toute plateforme matérielle spécifique.

La programmation d'une application sous Android peut s'effectuer de deux manières :

- via « Android SDK », kit de développement Java utilisable sous l'IDE Eclipse ;
- via « App Inventor », une interface de développement par prototypes.

Il faut noter qu'un programme développé avec l'App Inventor est plus rapide que sous le SDK car il permet de s'affranchir de certaines contraintes de programmation. En revanche, il est un peu plus limité (pas de traitements complexes), mais son développement continu par Google et les développeurs amateurs devraient étendre son spectre de possibilités. En effet, Google compte beaucoup sur les communautés de développeurs Open Source pour améliorer le système existant, ainsi que pour développer les meilleures applications pour Smartphones. Des primes sont offertes aux développeurs des meilleures applications dans le cadre de « Android Developer Challenge » [GCR10].

Les programmes développés pour Android peuvent être émulés sur une console virtuelle avant le transfert de l'exécutable sur le téléphone pour un test in situ.

Le choix du système d'exploitation Android permet de couvrir la plupart des Smartphones du marché actuel. Comme le montre la figure 18, Android domine actuellement le marché des smartphone et l'évolution de ses ventes est importante en une année grâce à ses innovations et l'intérêt que portent les développeurs pour le faire évoluer et ainsi dépasser ses concurrents [Alt11].

Worldwide smart phone market					
Market shares Q4 2010, Q4 2009					
OS vendor	Q4 2010		Q4 2009		Growth Q4'10/Q4'09
	shipments (millions)	% share	shipments (millions)	% share	
Total	101.2	100.0%	53.7	100.0%	88.6%
Google	33.3	32.9%	4.7	8.7%	615.1%
Nokia	31.0	30.6%	23.9	44.4%	30.0%
Apple	16.2	16.0%	8.7	16.3%	85.9%
RIM	14.6	14.4%	10.7	20.0%	36.0%
Microsoft	3.1	3.1%	3.9	7.2%	-20.3%
Others	3.0	2.9%	1.8	3.4%	64.8%

Source: Canalys estimates, © Canalys 2011

Figure 18 : Comparaison entre les parts de marché des ventes de Smartphones par systèmes d'exploitation en 2009 et 2010 [Alt11]

Deux applications différentes ont été développées sous Android : une version avec un calculateur d'itinéraire et une base de données embarqués sur le smartphone et une autre avec

un calculateur d'itinéraire et une base de données externes. Ces deux applications ont été développées en collaboration avec une équipe de deux étudiants que j'ai encadrés. Elles seront présentées dans la suite.

3.4.2 Application sous Android avec calculateur et base de données embarqués

La figure 19 illustre l'architecture de cette application. Contrairement au prototype présenté précédemment, nous supposons, que la base de données avec les articles du magasin ainsi que le calculateur de l'itinéraire sont embarqués sur le Smartphone pour diminuer le risque de perte d'informations.

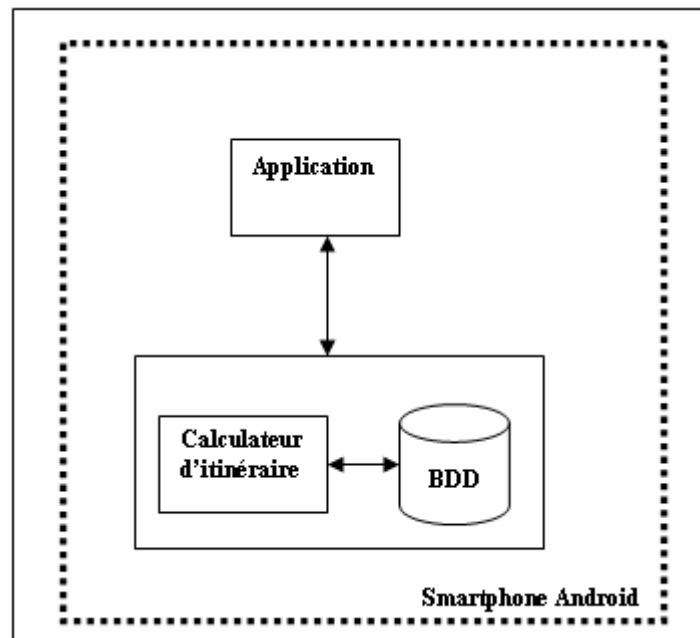


Figure 19 : Architecture du système de guidage avec un calculateur et une BDD embarqués

Compte tenu des limitations techniques d'un Smartphone (mémoire, rapidité, ports...), la base de données doit être légère. En effet, nous avons construit une base de données contenant seulement les rayons et les distances entre eux. Cette base a été intégrée dans le programme de calcul de l'itinéraire. L'utilisateur entrera le nom du rayon recherché et l'application calculera le chemin le plus court pour aller les récupérer.

Cette application a été développée avec Android SDK sous Eclipse. Nous avons fait des tests pour étudier la performance du système en fonction du nombre des rayons ajoutés par

l'utilisateur. Nous avons remarqué qu'en ajoutant seulement 4 rayons, le temps de calcul devient égal à 8 secondes ce qui est trop lent et inacceptable pour une application pareille. Le détail du calcul de l'itinéraire sera donné dans le chapitre 4. Par ailleurs, pour des raisons de sécurité, la base de données du magasin est souvent confidentielle. Pour toutes ces raisons, nous avons décidé, de délocaliser le calculateur et la base de données sur un serveur pour alléger l'application [HKB11c].

3.4.3 Application sous Android calculateur base de données externes

Comme le montre la figure 20, dans le système de guidage, nous avons hébergé le calculateur d'itinéraire ainsi que la base de données intégrée dans un ordinateur portable équipé du Bluetooth. Cet ordinateur représente le serveur et communique avec le Smartphone via Bluetooth. La portée de ce dernier est de 100 mètres. Cependant, son efficacité maximale est dans un périmètre de 30 mètres.

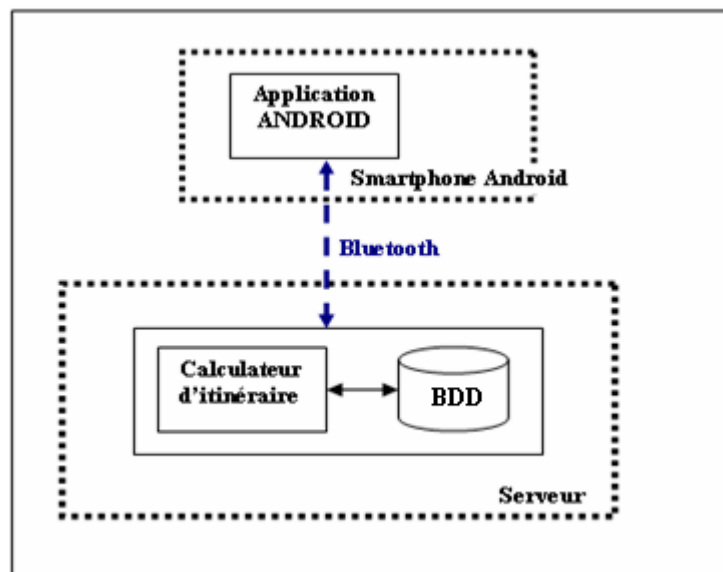


Figure 20 : Architecture du système de guidage avec un calculateur et une base de données externes

Cette application a été développée à l'aide de l'App Inventor, l'outil permettant de générer des applications Android en imbriquant des blocs entre eux. Nous avons construit 3 blocs :

- Un bloc de démarrage qui s'occupe de ce qui se produit lorsque l'application est lancée à partir du bureau : définition de la fréquence d'horloge, chargement des

périphériques Bluetooth déjà rencontrés, les sauvegarder dans une base de données,...

- Un bloc pour la gestion de la connexion Bluetooth
- Un bloc pour la sélection et la validation des articles entrés

La figure 21 illustre le bloc dédié à la gestion de la connexion Bluetooth.

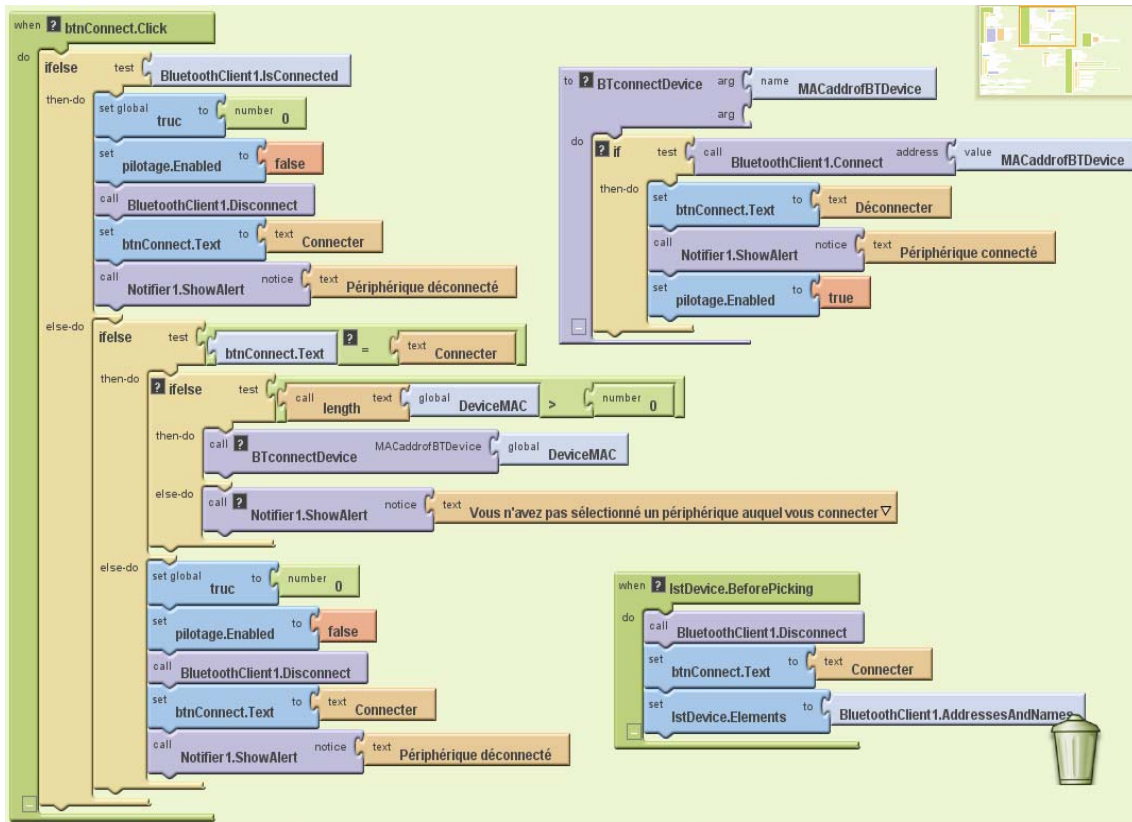


Figure 21: Exemple de bloc : Gestion de la connexion Bluetooth



Figure 22 : Ecran d'accueil pour Android

La figure 22 représente l'écran d'accueil de l'application Android.

Nous avons testé cette application avec une base de données de 50 articles. Par exemple, pour 45 articles entrés et même plus, le temps de calcul est moins de 1 seconde ce qui est raisonnable [HKB11c].

Il est vrai que cette application a donné des résultats raisonnables mais avec une base de données de taille réduite et sous forme d'un fichier texte pour faciliter l'implémentation. Pour cette raison, nous avons décidé de créer une base de données externe contenant les articles du magasin. Nous présentons, dans la suite, l'architecture de l'application sous Android en cours.

3.4.4 Application sous Android

Nous proposons de créer une application mobile légère où le calculateur et la base de données ne sont pas intégrés. Ce choix permet de soulager l'application mobile et rendre son utilisation plus pratique. Ainsi, le client aura la possibilité de la télécharger sur son smartphone et l'utiliser à l'entrée du magasin. Par contre, cela oblige l'application mobile à se connecter au serveur pour calculer un itinéraire.

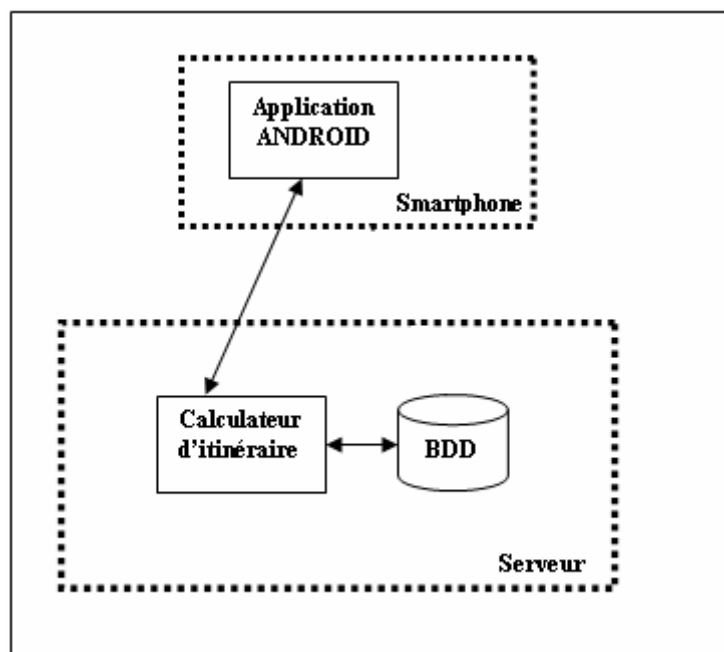


Figure 23 : Architecture de l'application sous Android en cours

La figure 23 représente la nouvelle architecture du système i-GUIDE qui est une architecture 3-tiers avec 3 couches logicielles :

- *La couche présentation* qui correspond à l’affichage et le dialogue avec l’utilisateur. Elle est représentée par l’application sous Android
- *La couche métier* qui correspond à la mise en œuvre de l’ensemble des règles de gestion et de la logique applicative. Elle est représentée par le calculateur de l’itinéraire
- *La couche accès aux données* qui correspond aux données destinées à être conservées sur une durée. Elle est représentée par la base de données.

La communication entre le serveur et l’application Android peut se faire de différentes manières : via le Bluetooth, via le WiFi ou encore à travers une connexion 3G.

Dans l’application précédente, afin de faciliter la communication entre le serveur et l’application mobile, nous avons choisi le Bluetooth. Cependant, cette technologie demande un investissement et une large infrastructure pour son implémentation dans un magasin. Contrairement au Wifi qui existe déjà dans le magasin mais reste à vérifier la possibilité de son utilisation par les clients du magasin pour des raisons de sécurité. Le moyen de communication le plus utilisé dans les applications mobiles est la connexion 3G surtout que les Smartphones possèdent cette fonctionnalité et la plupart des fournisseurs de téléphonie mobile en France incluent une connexion 3G illimitée dans leurs abonnements.

3.5. Conclusion

Dans ce chapitre, nous avons présenté les nouvelles technologies de localisation dans un espace fermé. Parmi ces technologies, nous avons cité : le Bluetooth, le WiFi, le GPS indoor, la RFID et le NFC. Etant donné que la plupart des magasins et en particulier les magasins Leroy Merlin disposent du réseau WiFi, nous avons décidé de profiter de l’existence de ce réseau et donc de proposer une solution qui lui est compatible tout en veillant à l’interopérabilité des technologies hétérogènes utilisées. Ainsi, nous avons proposé le couplage du réseau WiFi et la technologie RFID. Nous avons choisi la technologie RFID pour localiser l’utilisateur dans l’hypermarché et détecter sa position en temps réel. Par ailleurs, nous avons proposé une architecture du système de guidage ainsi que le prototype réalisé. Ce dernier est composé d’une partie client embarquée contenant un lecteur RFID, une partie serveur et des étiquettes RFID. Le Wifi a été utilisé pour assurer la communication entre le serveur et le terminal client. L’inconvénient de ce système est que quand on perd le signal du

WiFi, le système devient inactif. Pour cela, nous avons proposé d'embarquer les différentes parties du système sur un Smartphone.

De plus du prototype, nous avons proposé de développer une application sous Android de guidage sur un Smartphone. Le choix de ce système d'exploitation permet de couvrir la plupart des Smartphones du marché actuel. Pour rendre le système plus performant, nous avons proposé de créer une application mobile légère où le calculateur et la base de données ne sont pas intégrés. Ceci permettra de soulager l'application mobile et rendre son utilisation plus pratique. Ainsi, le client aura la possibilité de la télécharger sur son smartphone et de l'utiliser à l'entrée du magasin.

Par ailleurs, le système i-GUIDE pourra devenir un système de géolocalisation de produits multi-canal. Ainsi, nous envisageons de créer une application web permettant aux clients de préparer leur liste de courses chez eux et de déterminer le chemin à parcourir. L'entrée des articles pourra se faire en scannant le code 2D à partir du catalogue ou sur l'écran. Ceci permettra à l'utilisateur d'avoir des informations sur les articles. Les articles sélectionnés ainsi que le chemin à parcourir seront enregistrés sur le compte de l'utilisateur. Ensuite, il pourra suivre ce chemin en magasin. Par ailleurs, il aura la possibilité d'éditer sa liste de courses et le nouveau chemin sera recalculé dynamiquement en temps réel en magasin. Les approches d'optimisation de parcours seront traitées dans le chapitre suivant.

CHAPITRE 4 : APPROCHES D'OPTIMISATION DE PARCOURS EN TEMPS REEL DANS UN HYPERMARCHÉ

4.1. Introduction

Le problème d'optimisation de parcours en temps réel dans un hypermarché (POPHTR) consiste à trouver le chemin le plus court pour récupérer tous les articles existants dans une liste de courses. Ainsi, ce problème peut être considéré comme un problème de routage. Pour cette raison, nous proposons dans ce chapitre de présenter, tout d'abord, des définitions générales sur les parcours. Ensuite, nous exposons l'état de l'art sur les méthodes d'optimisation de parcours, notamment, dans le domaine de transport et du guidage par GPS.

Enfin, nous présentons la position du problème d'optimisation de parcours dans un hypermarché et les approches de résolution proposées.

4.2. Définitions générales

Les problèmes d'optimisation sont des problèmes très utiles pour l'ensemble de la communauté scientifique. En effet, ils fournissent des techniques génériques qui peuvent être utilisées dans différents domaines. Parmi les problèmes d'optimisation, nous nous intéressons au problème d'optimisation de parcours qui a été traité dans plusieurs domaines.

Dans cette section, nous rappelons quelques définitions utiles que nous utilisons dans la suite du chapitre.

4.2.1 Optimisation combinatoire et théorie de la complexité

Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais compte un très grand nombre d'éléments, et il est décrit de manière implicite, c'est-à-dire par une liste de contraintes que doivent satisfaire les solutions réalisables. Pour définir la notion de meilleure solution, une fonction, dite fonction objectif, est introduite. Pour chaque solution, elle renvoie un réel et la meilleure solution (ou solution optimale) est celle qui minimise ou maximise la fonction objectif. Un problème d'optimisation combinatoire peut avoir plusieurs solutions possibles.

Le problème du plus court chemin entre deux sommets A et B d'un graphe est un exemple classique de problème d'optimisation combinatoire. L'ensemble des solutions réalisables est l'ensemble des chemins entre A et B tandis que la fonction objectif est la longueur du chemin.

Trouver une solution optimale dans un ensemble discret et fini est un problème facile en théorie : il suffit d'essayer toutes les solutions et comparer leurs qualités pour distinguer la meilleure. Cependant, en pratique, l'énumération de toutes les solutions peut prendre beaucoup de temps alors que le temps de recherche de la solution optimale est un facteur très important et c'est à cause de lui que les problèmes d'optimisation combinatoire sont réputés difficiles. Ainsi, la théorie de la complexité s'intéresse à l'étude formelle de la difficulté des problèmes en informatique. Elle se distingue de la théorie de la calculabilité qui s'attache à savoir si un problème peut être résolu par un ordinateur. La théorie de la complexité consiste à estimer le temps de calcul d'un algorithme (complexité temporelle) et les besoins en mémoire informatique (complexité spatiale). Elle repose sur différentes classes de complexité permettant de créer un ordre sur les algorithmes proposés pour résoudre un problème. Parmi les classes de complexité existantes, citons la classe P. Un problème de décision est dans la classe P s'il peut être décidé par un algorithme déterministe en un temps polynomial par rapport à la taille de l'instance. On qualifie alors le problème de polynomial.

La plupart des problèmes d'optimisation combinatoire appartiennent à la classe des problèmes NP-difficile. Cela signifie qu'il est peu probable qu'il existe un algorithme permettant de le résoudre en temps polynomial [GJ79].

4.2.2 Définitions utiles

Dans cette section, nous présentons quelques définitions utiles sur les parcours [LPS03].

Définition 1 : Un graphe orienté G est un couple $(N;A)$ avec N un ensemble dont les éléments sont appelés nœuds et A un ensemble dont les éléments sont des couples ordonnés de nœuds appelés arcs.

Définition 2 : Un graphe non orienté G est un couple $(N;A)$ avec N un ensemble dont les éléments sont appelés nœuds et A un ensemble dont les éléments sont des paires de nœuds appelées arrêtes.

Définition 3 : Un parcours d'un graphe G est hamiltonien s'il passe exactement une fois par tout sommet.

Définition 4 : Un parcours d'un graphe G est eulérien s'il passe exactement une fois par chaque arc ou arrête de G .

Définition 5 : Un chemin de longueur q est une séquence de q arcs. Un chemin fermé est un circuit. Un arc peut être vu comme un chemin de longueur 1 et une boucle comme un circuit de longueur 1.

Définition 6 : Une chaîne de longueur q est une séquence de q arêtes. Une chaîne fermée est un cycle. Une arrête peut être vue comme une chaîne de longueur 1 et une boucle comme un cycle de longueur 1.

Le terme parcours regroupe les chemins, les circuits, les chaînes et les cycles.

4.3. Etat de l'art sur les méthodes d'optimisation de parcours

Dans la littérature, le problème d'optimisation de parcours en temps réel a été évoqué dans plusieurs domaines. Parmi ces domaines, nous décrivons, dans cette section, quelques méthodes de résolutions utilisées, essentiellement, dans le domaine de transport et le guidage par GPS.

4.3.1 Méthodes d'optimisation liées aux problèmes de tournées

Dans le domaine de transport, nous évoquons les problèmes de routage ou encore problèmes de tournées à travers, essentiellement, trois grandes familles de problèmes :

- *le problème de voyageur de commerce* ou *Traveling Salesman Problem (TSP)* qui consiste à trouver la plus courte tournée permettant de visiter n villes et de revenir au point de départ en ne visitant chaque ville qu'une seule fois [ABCC07], [JM02], [LPS03].
- *le problème de tournées de clients* ou *Vehicle Routing Problem (VRP)* qui est une extension du TSP. Il vise à déterminer les tournées permettant de servir une liste de clients en minimisant le coût de livraison [Hou05].
- *le problème de ramassage et de livraison* ou *Pickup and Delivery Problem (PDP)* qui représente une variante du VRP. Il ajoute aux propriétés du VRP le fait que chaque client introduit deux positions géographiquement différentes : la première pour le ramassage du produit et la deuxième pour la livraison de ce dernier. Ceci induit directement une contrainte de précédence à ajouter au problème du VRP

classique sachant que, dans une tournée, chaque opération de livraison doit être précédée par l'opération de ramassage respective [Mech95]

Le tableau 6 représente les différentes caractéristiques de ces trois problèmes.

Paramètres	TSP	VRP	PDP
Ensemble des nœuds	Villes	Clients	Clients et Fournisseurs
Contraintes	<ul style="list-style-type: none"> - Passer par toutes les villes une et une seule fois - Contrainte de temps - Contrainte de capacité 	<ul style="list-style-type: none"> - Passer par tous les clients une et une seule fois - Contrainte de temps - Contrainte de capacité 	<ul style="list-style-type: none"> - Passer par tous les clients une et une seule fois - Passer par tous les fournisseurs une et une seule fois - Passer par le fournisseur avant le client - Contrainte de temps - Contrainte de capacité
Critères	Distance, temps, capacité...	Distance, temps, nombre de clients...	Distance, temps, capacité, nombre de clients...

Tableau 6 : Caractéristiques des trois grandes familles du problème de transport

Pour résoudre ces problèmes d'optimisation, il existe plusieurs types d'approche.

Lorsque l'algorithme utilisé ne permet pas nécessairement d'obtenir une solution optimale du problème, la méthode est appelée méthode approchée ou heuristique. Ce genre d'approche est souvent adapté au problème à résoudre, et elle permet généralement la génération d'une solution réalisable de bonne qualité en des temps de calcul peu importants et en nécessitant également une charge mémoire limitée. L'expression « de bonne qualité » signifie ici que l'algorithme permet de trouver une solution de valeur assez proche de la valeur optimale du problème. Les heuristiques sont très utiles pour traiter les problèmes de grande taille, ou encore pour générer des solutions rapidement dans un contexte spécifique tel que les applications en temps réel.

Pour résoudre le problème de tournée de clients dynamique ou DVRP (Dynamic Vehicle Routing Problem) qui représente l'un des problèmes de tournées de clients où toutes les demandes pour tous les clients ne sont pas connues d'avance. Ainsi, on peut avoir, par exemple, l'apparition d'un nouveau client en cours de journée. Dans ce cas, le décideur doit changer la planification des clients en réponse aux nouvelles demandes qui arrivent au cours du temps [KPS98], [GGPS98].

Bianchi [Bia00] ainsi que Larsen [LAR01] ont présenté plusieurs méthodes de résolution du problème DTRP (Dynamic Traveling Repairman Problem), un cas du DVRP (Dynamic Vehicle Routing Problem). Parmi ces méthodes, nous citons quelques heuristiques simples utilisées :

- FCFS (First Come First Serve) : Les demandes sont servies dans l'ordre dans lequel elles ont été reçues par le planificateur.
- SQM (Stochastic Queue Median) : C'est une modification de la méthode FCFS où le réparateur part directement de la médiane de la région de service au prochain client. Après avoir accompli le service, le réparateur revient à la médiane et attend la prochaine demande.
- NN (Nearest Neighbor) : Après avoir servi un client, le réparateur se dirige vers le client le plus proche non encore servi.
- PART (PARTitioning policy) : La région de service est divisée en sous-régions dans lesquelles les demandes sont servies en utilisant l'heuristique FCFS.
- TSP (Traveling Salesman Problem strategy) : Au fur et à mesure de l'arrivée des demandes, des ensembles composés de n clients sont formés et mis dans une file d'attente. Après, chaque ensemble est traité séparément comme étant un TSP.

Ces heuristiques simples ont été appliquées sur de petites instances de DTRP et elles deviennent instables dès que le nombre de demandes dynamiques augmente.

La méthode de colonies de fourmis a été introduite par Colormi et al en 1992 [CDM92] pour résoudre le problème du TSP et par Gambardella et al. [GROCDML03] pour résoudre le problème de VRP. La méthode de recherche tabou a été utilisée, pour la résolution du VRP, dans [GHL94], [Tai93], [XK96] et [Tai98]. Par ailleurs, nous citons les travaux de Taillard et al. [TBGGP97] et Cordeau et Laporte [CDDSS01] qui utilisent une méthode tabou également pour résoudre un problème avec des fenêtres de temps larges pour le problème de VRPTW (Vehicle Routing Problem With Time Windows). Par ailleurs, dans [DBDG10], les auteurs ont utilisé la méthode tabou pour l'optimisation des tournées de techniciens dans un environnement dynamique avec des temps de service stochastique. Bräysy et al. ont présenté dans [BG05] une structure de voisinage basée sur le « CROSS exchange » exploité par une heuristique de recherche tabou pour résoudre le problème de tournée de véhicules avec des fenêtres de temps souples. Dans [Els10], El-Sherbeny donne une revue de la littérature sur les méthodes de résolution du VRPTW. En outre, Dridi et al. ont proposé un algorithme génétique pour la minimisation du coût total de transport dans le cas de PDPTW [HKBK08].

4.3.2 Méthodes de résolution des problèmes de guidage des utilisateurs par GPS

Pour des raisons commerciales, peu de travaux ont été publiés sur les méthodes d'optimisation d'itinéraires dans un GPS. Cependant, des travaux ont été réalisés pour la conception d'un système de navigation GPS pour le guidage de l'utilisateur dans son déplacement en ville. Ces travaux ont choisi l'algorithme de Dijkstra pour l'optimisation de l'itinéraire. Dans [PH04], les auteurs ont conçu un système de guidage des personnes ayant une déficience visuelle pour se déplacer dans la ville sans avoir besoin d'une aide extérieure. Le système est composé d'un GPS, d'un PDA et d'un kit main-libre Bluetooth. Le GPS est utilisé pour recueillir les informations sur la position de l'utilisateur (latitude, longitude). L'application utilise l'algorithme de Dijkstra pour le calcul du chemin le plus court et le moins risqué. Il compare la trajectoire que fait l'utilisateur avec celle indiquée par le système pour le prévenir en cas de dépassement. Quand il arrive au coin de la rue, une alerte est déclenchée pour l'avertir. Par ailleurs, Abboud et al [AAK04] ont créé un système de navigation GPS en temps réel. Il détermine la position de l'utilisateur sur la carte de Beirut en utilisant un récepteur GPS. Cette carte est une carte GIS (Geographic Information System) qui contient des données spatiales et des attributs [TF09]. Ces attributs contiennent des informations sur les données spatiales telles que le nom de la route, le nœud de départ et le nœud d'arrivée, la longueur de la route, la durée de parcours de la route, le nom de la rue... Le chemin le plus court est défini comme étant le chemin ayant le coût le plus faible. Le coût peut être soit la somme des distances entre les rues soit la somme des temps nécessaires pour parcourir chaque rue. Il existe plusieurs algorithmes utilisés pour le calcul du chemin le plus court tels que : l'algorithme de Dijkstra, l'algorithme de Bellman Ford, l'algorithme de tri topologique et l'algorithme A Star (A*). Le choix de l'algorithme dépend de la taille de la carte. L'algorithme de Bellman Ford peut prendre beaucoup plus de temps pour trouver la solution mais ses résultats peuvent être similaires ou même meilleures que celles utilisant l'algorithme Dijkstra. Quant à l'algorithme de tri topologique, il peut trouver le chemin le plus court mais il peut ne pas être le meilleur dans cette étude. L'algorithme A* peut trouver des résultats similaires à celles de l'algorithme Dijkstra avec moins de mémoire mais son implémentation est beaucoup plus difficile pour trouver le chemin le plus court. Pour toutes ces raisons, l'algorithme Dijkstra a été retenu pour l'implémentation de l'application surtout qu'il est le plus utilisé dans les systèmes GIS. Le temps de réponse dépend du nombre de

nœuds entre le point de départ et le point d'arrivée. L'utilisateur entre la destination et l'algorithme Dijkstra calcule le chemin le plus court. Cette application guide l'utilisateur en lui indiquant les directions et recalcule l'itinéraire en temps réel à chaque fois où il change de direction. Par ailleurs, Sauvanet et al. [Sau11] ont proposé des méthodes permettant de calculer des itinéraires adaptés aux cyclistes à l'échelle d'une agglomération dans un contexte multiobjectif en tenant en compte plusieurs critères tels que la distance, l'insécurité et l'effort. Deux approches ont été abordées pour résoudre ce problème : l'approche a posteriori dans laquelle l'ensemble des solutions de compromis est calculé et l'approche a priori dans laquelle les préférences de l'utilisateur sont prises en compte et permettent d'orienter la recherche pour privilégier les chemins les plus prometteurs. Ces deux approches sont basées sur des méthodes de labelling.

L'algorithme de Dijkstra ainsi que l'algorithme A* permettent de calculer le chemin le plus court entre deux nœuds uniquement. C'est la raison pour laquelle on ne peut pas les utiliser pour résoudre notre problème puisque l'algorithme doit optimiser le chemin passant par tous les articles existants dans la liste de courses.

4.4. Position du problème

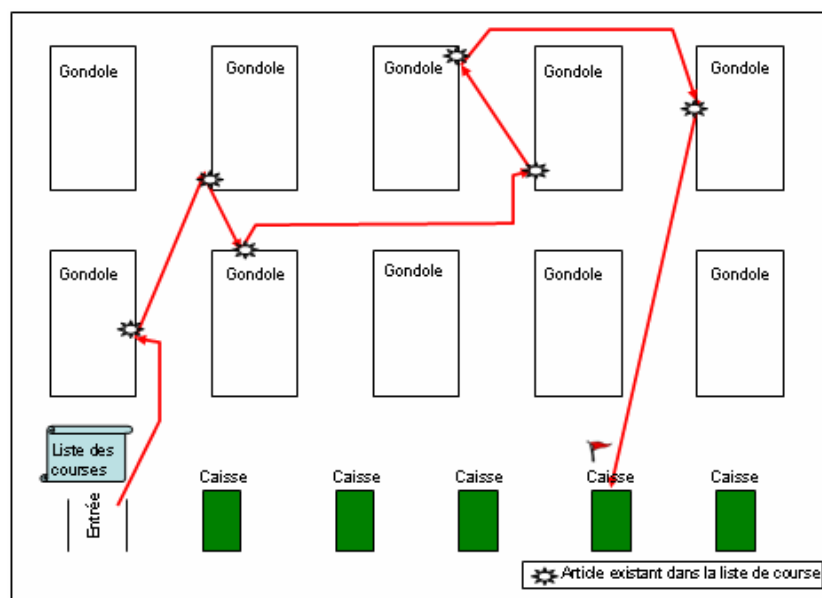


Figure 24 : Présentation du problème d'optimisation de l'itinéraire dans un hypermarché

Le système de guidage des personnes dans un hypermarché vise à les aider à trouver facilement les articles souhaités dans le magasin. Ce qui leur permettra de gagner du temps sur les achats besoins pour dégager du temps pour les achats plaisirs. Ainsi, comme le montre

la figure 24, à l'entrée du magasin, l'utilisateur constitue sa liste d'articles soit en tapant leurs noms soit en les récupérant à partir du catalogue en cours. Ensuite, une application cherche le chemin le plus court à parcourir à partir de la position actuelle de l'utilisateur et passant, une seule fois, par tous les articles existants dans la liste de courses.

Dans la littérature, ce problème d'optimisation est similaire au problème de voyageur de commerce (PVC ou TSP) défini au début du chapitre. Etant donné que le nœud de départ est la position actuelle de l'utilisateur alors que le nœud d'arrivée est la caisse, le résultat de notre problème sera un chemin hamiltonien alors que celui du PVC est un circuit hamiltonien. Le tableau 7 représente une comparaison entre les deux problèmes.

Paramètres	TSP	Notre problème
Ensemble des nœuds	Les villes	Les articles entrés par l'utilisateur
Contraintes	<ul style="list-style-type: none"> - Le nœud de départ est le même nœud d'arrivée - Passer par toutes les villes une et une seule fois 	<ul style="list-style-type: none"> - Le nœud de départ n'est pas le même nœud d'arrivée - Le point initial dépend de la position donnée par le système de localisation - Passer par tous les articles une et une seule fois
Critères	Distance, temps, coût...	Distance, passage par les rayons
Solution	Circuit hamiltonien	Chemin hamiltonien

Tableau 7 : Analogie entre le problème du voyageur de commerce et notre problème

Dans le problème d'optimisation de parcours dans un hypermarché, on trouve une partie statique qui consiste à trouver le chemin le plus court pour récupérer les articles souhaités et on trouve aussi une partie dynamique qui consiste à recalculer l'itinéraire en cas de détection d'un changement de direction grâce à la technologie de localisation. Des heuristiques constructives sont utilisées et adaptées aux caractéristiques du problème ensuite améliorées par l'algorithme de recherche tabou. Par ailleurs, les hypermarchés offrent souvent des promotions sur des articles bien déterminés et ces promotions peuvent être annoncées avant quelques jours, pour attirer des clients, ou bien le jour même. Nous nous intéressons, dans la deuxième partie d'optimisation de parcours, aux produits en promotion qui se passe pendant un intervalle de temps dans lequel le client doit récupérer l'article. Une heuristique d'insertion est utilisée pour insérer dynamiquement les articles avec fenêtre de

temps dans parcours. Ensuite, une recherche tabou adaptée à cette optimisation permet d'avoir des résultats intéressants. Dans la section suivante, nous donnons une description de la méthodologie suivie pour résoudre le problème d'optimisation de parcours dans un hypermarché.

4.5. Méthodologie d'optimisation de parcours dans les hypermarchés

Le principe de notre système de navigation dans les hypermarchés est le suivant : l'utilisateur entre sa liste d'articles et le système calcule le chemin le plus court. En cas d'ajout ou de suppression d'un article ou encore si l'utilisateur change de direction, un nouveau itinéraire sera recalculé dynamiquement. Pour cela, nous avons besoin d'une base de données facile à mettre à jour.

Nous avons proposé dans [HEB10b] et [HEB10c] une méthodologie pour résoudre ce problème d'optimisation. Les principales étapes de cette méthodologie sont :

1. Création de la base de données du magasin
2. Détermination des distances entre les articles deux à deux et leur insertion dans la base de données
 - i. Division du magasin en zones selon la disposition des gondoles
 - ii. Attribution de repères dans chaque zone
 - iii. Proposition d'une méthode de calcul des distances entre les articles deux à deux
 - iv. Insertion des distances dans la base de données
3. Tests des méthodes d'optimisation pour déterminer le chemin le plus court
4. Choix de la méthode la plus appropriée à notre problème
5. Proposition d'une méthode pour recalculer le chemin le plus court en cas de modification de la liste des courses ou changement de direction

Nous détaillons, dans la suite, la méthodologie de résolution du problème d'optimisation de parcours dans un hypermarché et ses différentes étapes.

4.6. Création de la base de données

La complexité de notre problème d'optimisation réside dans son aspect dynamique (ajout ou suppression d'articles dans la liste, changement de direction, très grand nombre d'articles dans l'hypermarché qui peuvent changer de place d'une période à une autre...).

Pour cela, nous avons créé une base de données qui détaille les différents éléments d'un magasin ce qui facilitera sa mise à jour plus tard.

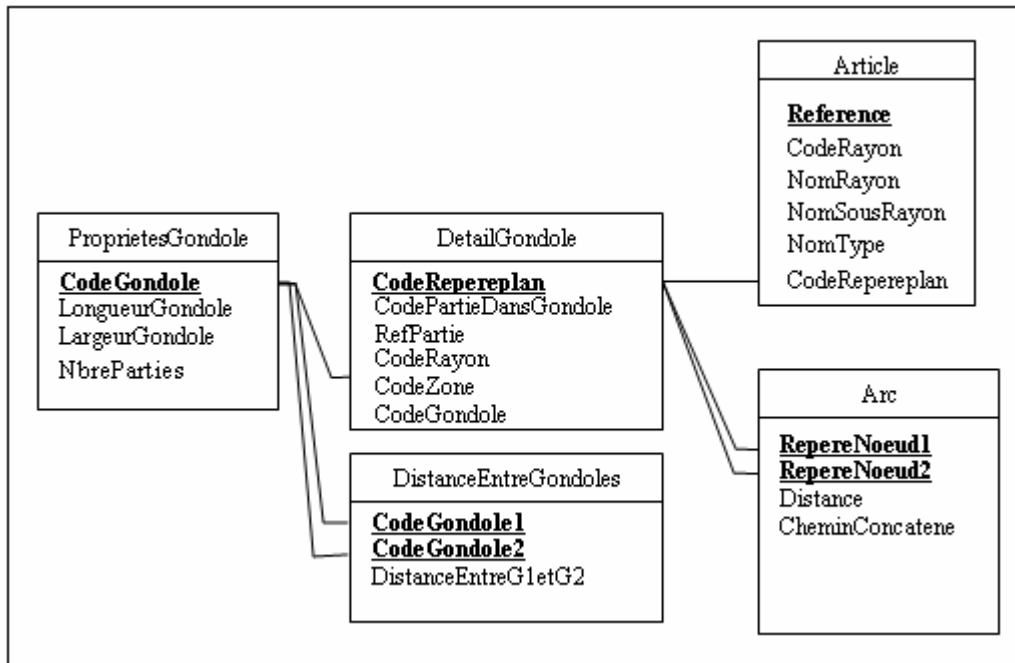


Figure 25 : Base de données d'un magasin

Nous rappelons qu'un magasin contient des gondoles et chaque gondole peut appartenir à un ou deux rayons. Chaque rayon est divisé en sous-rayons qui contiennent différents types de produits. Afin de faciliter le remplissage de la base de données, nous avons défini les valeurs entrées suivantes :

Informations sur les gondoles : la longueur de la gondole, sa largeur, la distance entre les gondoles, la zone et le/les rayon(s) auxquels elle appartient, le nombre de parties qu'elle contient et qui dépend du maillage choisi.

Informations sur les articles : la référence de l'article, le nom du sous-rayon, le nom du type de produits ainsi que le nom du sous-type de produits auxquels il appartient.

A partir de ces informations, nous avons effectué des programmes permettant de remplir le reste des champs. La figure 25 représente une base de données générique dont la structure est valable à n'importe quel magasin et les valeurs lui seront personnalisées.

Dans la section suivante, nous nous intéressons à la détermination des distances entre des articles pris deux à deux.

4.7. Détermination des distances entre les articles deux à deux

Avant de chercher le chemin le plus court à parcourir pour trouver les articles existants dans la liste de courses, il faut déterminer les distances entre articles pris deux à deux. Dans la suite de ce rapport de recherche, nous considérons un magasin Leroy Merlin qui nous a fourni des données réelles pour valider nos applications et tests. Ainsi, nous avons construit sa base de données sur laquelle nous avons travaillé.

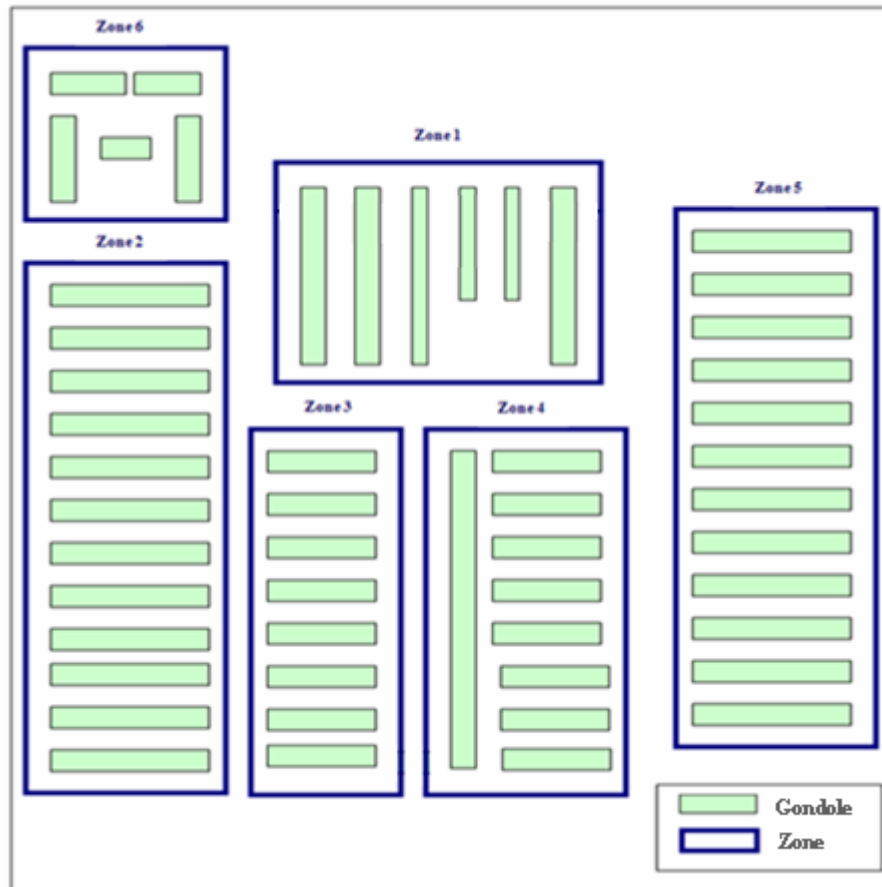


Figure 26: Division d'un magasin Leroy Merlin en 6 zones

Compte tenu du grand nombre d'articles et pour faciliter le calcul des distances, nous proposons de diviser le magasin retenu en 6 zones selon la disposition des gondoles dans le magasin comme le montre la figure 26. De plus, nous considérons les types de produits et non pas les articles. En d'autres termes, nous allons rassembler les articles voisins appartenant au même type de produits et calculer seulement la distance entre les différents types de produits. En cas où un type de produit contient plusieurs articles distants, on considérera le sous-type de produits.

Ensuite, nous mettons des repères sur chaque gondole. L'attribution des repères dépend de la longueur de la gondole, la disposition des produits et le maillage choisi. Comme le montre la figure 25, ces informations sont détaillées dans la base de données du magasin pour faciliter la mise à jour suite à tout changement. La table « DetailGondole » détaille l'emplacement de la gondole dans le magasin : la zone à laquelle elle appartient, le rayon duquel elle fait partie ainsi que les repères qu'elle contient.

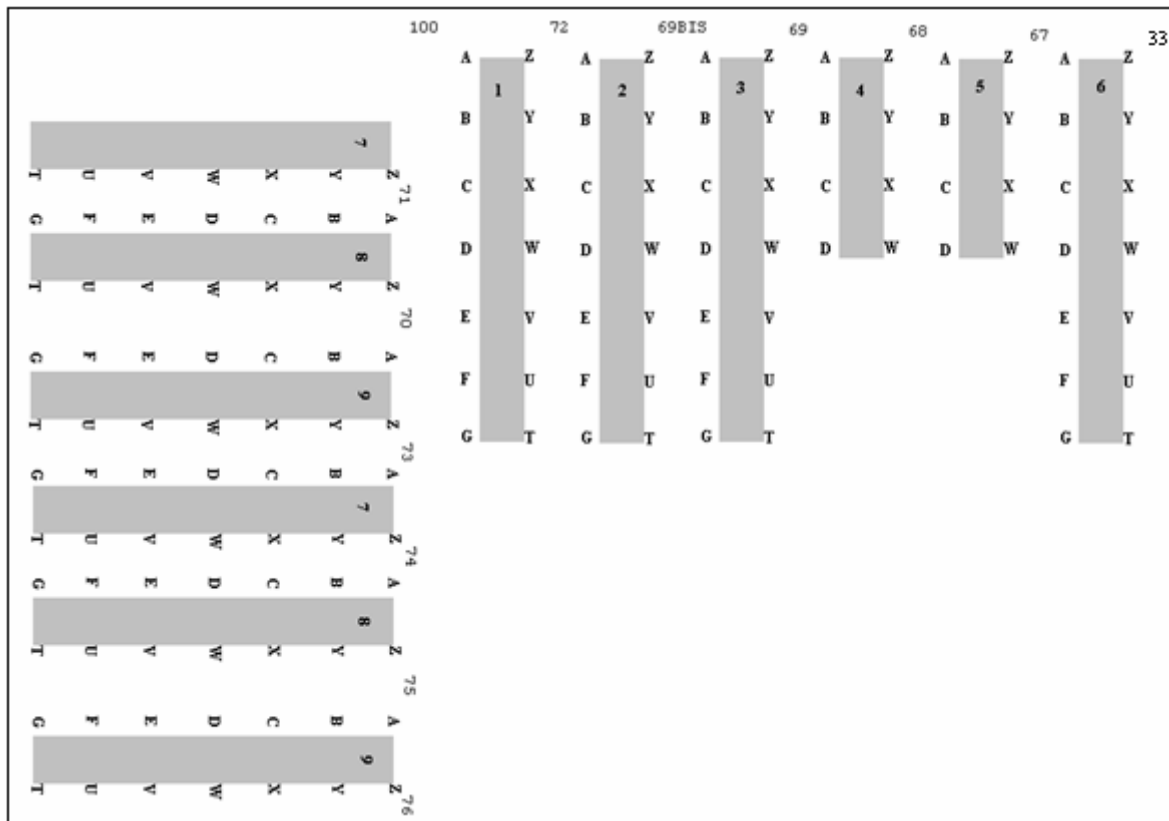


Figure 27 : Attribution de repères sur les gondoles

Compte tenu du grand nombre d'articles existants dans le magasin, nous avons travaillé sur la Zone 1 et la Zone 2 pour tester nos algorithmes. La figure 27 représente ces deux zones avec des repères. Nous avons choisi de diviser le repère en deux parties afin de faciliter le calcul des distances :

- une partie qui représente la référence de l'allée où se trouve l'article. Cette partie est appelée RefPartie dans la base de données,
- une partie qui peut prendre des valeurs sous forme de lettres différentes de chaque côté de la gondole. Cette partie est appelée CodePartieDansGondole dans la base de données.

Les types de produits sont rassemblés et chaque ensemble de types de produits est attribué à un repère sur le plan (CodeReprerePlan).

Dans la suite, nous déterminons les distances entre deux articles de la même zone ensuite entre deux articles existants dans deux zones différentes.

4.7.1 Détermination des distances entre deux articles appartenant à la même zone

La distance entre deux articles de la même zone est fonction de la longueur et de la largeur de la gondole à laquelle appartient chaque article, la distance entre ces gondoles, le nombre de parties de chaque gondole... Ainsi, notre algorithme a besoin de ces variables qu'il récupère de la base de données.

Cependant, il existe plusieurs cas à traiter. Ces cas dépendent essentiellement de l'emplacement de :

- la gondole contenant l'article 1 (Gondole1) par rapport à la gondole contenant l'article 2 (Gondole2) ;
- la RefPartie de l'article 1 (A1) par rapport à la RefPartie de l'article 2 (A2) ;
- le CodePartieDansGondole de A1 et A2 par rapport à l'extrémité des gondoles sur lesquelles ils se trouvent.

La figure 28 illustre un exemple de calcul de distances dans le cas où les deux articles n'appartiennent pas à la même gondole et ne se trouvent pas sur la même allée. Dans ce cas, on doit différencier 4 éventualités selon l'existence du repère sur la partie à droite ou la partie à gauche de sa gondole. La figure 28 traite une de ces éventualités dans le cas où le repère de l'article 1 se trouve dans la partie à gauche de la gondole 1 et le repère de l'article 2 se trouve dans la partie à droite de la gondole 2. Comme il peut exister deux chemins pour aller du repère 1 vers le repère 2, on doit choisir la distance du chemin le plus court.

```

Gondole1 : code de la gondole qui contient l'article A1
Gondole2 : code de la gondole qui contient l'article A2
cod1 : CodePartieDansGondole de l'article A1
cod2 : CodePartieDansGondole de l'article A2
ref1 : RefPartie de l'article A1
ref2 : RefPartie de l'article A2
LongG1 : longueur de la gondole 1
LongG2 : longueur de la gondole 2
LargG1 : largeur de la gondole 1
LargG2 : largeur de la gondole 2
NbrPartG1 : nombre de parties dans la gondole 1
NbrPartG2 : nombre de parties dans la gondole 2
DistGond[G1,G2] : distance entre la gondole 1 et la gondole 2
EH1 : Extrémité en haut de la gondole 1 du côté de A1
EB1 : Extrémité en bas de la gondole 1 du côté de A1
dist[A1,EH1]: distance entre A1 et EH1
dist[A1,EB1]: distance entre A1 et EB1

Si (Gondole1≠Gondole2) alors
  Si (ref1≠ref2)
    Si (A1 se trouve à gauche de Gondole1) et (A2 se trouve à droite de
      Gondole2)
      Distance1=(LongG1/NbrPartG1)*dist[A1,EH1]+(LargG1/2)+(LargG2/2)+
        DistGond[G1,G2]+(LongG2/NbrPartG2)* dist[A2,EH2]

      Distance2=(LongG1/NbrPartG1)*dist[A1,EB1]+(LargG1/2)+(LargG2/2)+
        DistGond[G1,G2]+(LongG2/NbrPartG2)* dist[A2,EB2]

      Si (Distance1-Distance2<=0) alors
        Distance = Distance1
      Sinon
        Distance = Distance2

```

Figure 28 : Extrait de l'algorithme de calcul des distances

4.7.2 Détermination des distances entre deux articles appartenant à deux zones différentes

Pour déterminer la distance entre les articles appartenant à deux zones différentes, nous avons eu l'idée de représenter les repères selon des coordonnées cartésiennes et nous avons défini un point à l'entrée du magasin comme le point d'origine (0,0). Cette méthode facilite le calcul des distances pour ce cas puisqu'on ajoute la distance euclidienne entre les extrémités de chaque zone.

Soient deux articles A1 et A2 appartenant respectivement à la Zone 1 et à la Zone 2.

Soit E1 l'extrémité de la Zone 1 appartenant à l'allée commune entre la Zone 1 et la Zone 2 et E2 l'extrémité de la Zone 2 appartenant à l'allée commune entre la Zone 1 et la Zone 2.

La distance $d[A1,A2]$ entre A1 et A2 est égale à :

$$d[A1,A2]= d[A1,E1] + d[E1,E2] + d[E2,A2]$$

avec $d[E1,E2]$ la distance euclidienne calculée à partir des coordonnées de E1 et E2.

Comme le montre la figure 29, pour aller à l'allée commune, il faut passer par l'extrémité EH1, en haut, ou par l'extrémité EB1, repère en bas.

La distance $d[A1,A2]$ est égale au Minimum($d1$, $d2$) tels que :

$$d1= d[A1,EH1] + d[EH1,E2] + d[E2,A2]$$

$$d2= d[A1,EB1] + d[EB1,E2] + d[E2,A2]$$

L'extrémité E2 appartient à la même gondole et à la même allée que A1.

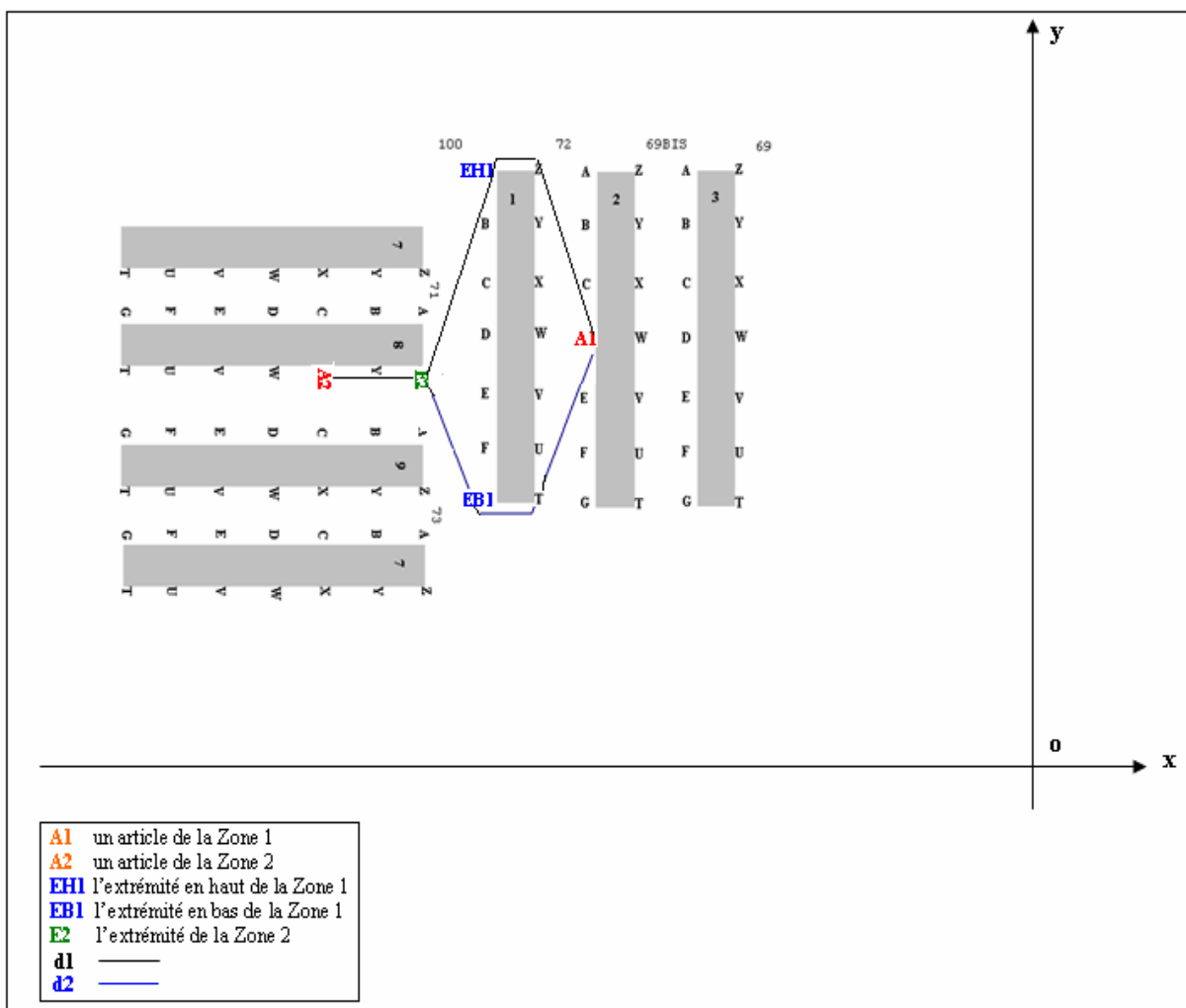


Figure 29 : Détermination des distances entre deux articles appartenant à deux zones différentes

Enfin, nous insérons toutes ces distances dans la base de données, pour les utiliser plus tard dans la partie optimisation. Par ailleurs, nous déterminons le chemin entre deux articles.

Ce chemin représente l'ensemble des nœuds par lesquels passe l'utilisateur pour aller d'un article à un autre.

Dans ce travail de recherche, tous nos tests ont porté sur la Zone 1 et la Zone 2 seulement. Cependant, nous envisageons agrandir notre domaine d'application dans des travaux futurs. Pour le calcul des distance entre des articles appartenant à deux zones différentes, il faut penser à construire un algorithme de calcul plus générique se basant sur les méthodes de recherche du chemin le plus court entre deux points telles que la méthode Dijkstra, la méthode A*...

Après avoir déterminé et inséré les distances dans la base de données, nous nous concentrons dans la section suivante sur les méthodes d'optimisation de parcours dans un hypermarché.

4.8. Approches d'optimisation de parcours dans un hypermarché

Dans cette section, nous proposons deux approches d'optimisation du parcours de l'utilisateur dans un hypermarché. La première permet de donner le chemin le plus court à parcourir pour récupérer les articles existants dans la liste de course et la deuxième ajoute à ce problème d'optimisation, une contrainte de temps pour certains articles de la liste dans le cas d'une promotion qui dure pendant un certain temps et dont la quantité est limitée.

4.8.1 Problème d'optimisation de parcours dans un hypermarché (POPH)

L'objectif de notre problème est de guider les clients dans un magasin en leur montrant le chemin pour trouver tous les articles ajoutés à leur liste de courses en minimisant la distance à parcourir. Pour ce faire, nous proposons, tout d'abord, une formulation mathématique du problème. Ensuite, nous exposons les différentes méthodes d'optimisation que nous avons adaptées à notre problème. Enfin, nous présentons les tests et les résultats obtenus.

4.8.1.1 Formulation mathématique du problème

Nous présentons, dans la suite, une formulation mathématique du problème par analogie avec le PVC.

Notre problème est caractérisé par les paramètres suivants :

- N : Nombre des nœuds représentant les articles existants dans la liste de courses du client,
- Ni : Article au nœud i,
- c : Le nœud final qui représente la caisse
- Dij : distance entre le nœud i et le nœud j, si $D_{ij} = \infty$ alors le chemin entre i et j n'existe pas,
- $i = 0..N$: indice des nœuds prédécesseurs,
- $j = 0..N$: indice des nœuds successeurs.
- Il y a un seul client qui cherche les articles existants dans sa liste de courses,
- Un nœud n'est visité par le client qu'une et une seule fois,
- Il y a un seul nœud de départ qui représente la position actuelle du client,
- Le client commence le trajet à partir de sa position actuelle et se dirige vers la caisse à la fin.

$$X_{ij} = \begin{cases} 1 & \text{Si le client part du nœud } i \text{ et se dirige vers le nœud } j \\ 0 & \text{Sinon} \end{cases}$$

La fonction à optimiser est la suivante :

$$\text{Minimiser } F = \sum_{i \in N} \sum_{j \in N} D_{ij} X_{ij} \quad [1]$$

Sous les contraintes :

$$\sum_{j \in N} X_{0j} = 1 \quad [2]$$

$$\sum_{i \in N} X_{ic} = 1 \quad [3]$$

$$\sum_{i \in N} X_{iu} + \sum_{j \in N} X_{uj} = 2 \quad \forall u \in N \quad [4]$$

L'équation [2] assure que chaque sommet n'est visité qu'une seule fois par le client.

L'équation [3] assure que client se dirige vers la caisse une fois.

L'équation [4] assure la continuité d'une tournée par le client : le sommet visité doit impérativement être quitté.

4.8.1.2 Proposition des méthodes de résolution du problème d'optimisation

Nous considérons dans notre problème que le nombre d'articles ajoutés par le client à sa liste de courses ne dépasse pas 100 alors notre problème est similaire au PVC avec 100 villes. Ainsi, le problème est considéré, par analogie, comme un problème NP difficile [LPS03]. Plusieurs recherches ont été faites pour résoudre ce problème. Parmi ces méthodes, on peut citer l'algorithme de Branch and Bound, l'algorithme de Christofides et l'algorithme du Plus Proche Voisin.

Nous avons adapté ces algorithmes à notre problème d'optimisation. Dans la suite, nous décrivons les modifications ajoutées ainsi que les résultats obtenus.

4.8.1.2.1 La méthode de Branch and Bound

La méthode de Branch and Bound, appelée aussi la méthode par séparation et évaluation progressive, se base sur l'énumération et l'évaluation des solutions possibles. Son algorithme construit une arborescence et évalue, pour chacune des branches, la possibilité de trouver la solution optimale. Ensuite, seuls les sommets qui peuvent mener à une solution intéressante sont examinés pour éviter de parcourir entièrement l'arbre des solutions [BT85], [LPS03], [ABCC07].

```

set N ordered; // N l'ensemble des articles
param n := card {N};
set LINKS := {i in N, j in N: ord(i) < ord(j)};
param cost {LINKS} >= 0;
var X {LINKS} in {0,1}, integer;
minimize TotCost: sum {(i,j) in LINKS} cost[i,j] * X[i,j];
subj to Tour {i in N diff {0,n}}:
sum {(i,j) in LINKS} X[i,j] + sum {(j,i) in LINKS} X[j,i]=2;
subj to TourA {i in 0}:
sum {(i,j) in LINKS} X[i,j] + sum {(j,i) in LINKS} X[j,i]=1;
subj to TourK {j in {n}}:
sum {(i,j) in LINKS} X[i,j] + sum {(j,i) in LINKS} X[j,i]=1;

```

Figure 30 : Algorithme de Branch and Bound

Nous avons testé cet algorithme en utilisant AMPL comme langage de modélisation et CPLEX comme solveur. La figure 30 illustre l'algorithme de Branch and Bound implémenté par analogie au problème de TSP. Il est vrai que cette méthode donne des solutions optimales

mais dès que le nombre des articles devient grand, le temps de calcul devient important. C'est pour cette raison, cette méthode n'a pas été retenue.

4.8.1.2.2 La méthode de Christofides

L'algorithme de Christofides [Chr76] est un algorithme basé sur l'Arbre Recouvrant de Poids Minimal (ARPM) appelé en anglais « Minimum Spanning Tree » (MST). Il existe plusieurs algorithmes pour déterminer cet arbre dont on peut citer les plus utilisées [HK69], [LPS03] :

- L'algorithme de Prim qui consiste à choisir, arbitrairement, un sommet et à faire croître un arbre à partir de ce sommet. Chaque augmentation se fait de la manière la plus économique possible.
- L'algorithme de Kruskal qui consiste à ranger, tout d'abord, par ordre de poids croissant les arêtes d'un graphe, puis à retirer une à une les arêtes selon cet ordre et à les ajouter à l'ARPM cherché tant que cet ajout ne fait pas apparaître un cycle dans l'ARPM.

Le principe de l'algorithme de Christofides est le suivant :

1. Construire l'ARPM à partir de l'ensemble des nœuds
2. Déterminer un couplage de coût minimum sur les sommets de degré impair et l'ajouter à l'ARPM.
3. Déterminer un cycle eulérien à partir du graphe obtenu
4. Raccourcir la tournée si elle passe plus d'une fois par un même sommet.

Nous avons utilisé l'algorithme Prim pour déterminer l'ARPM. A partir d'un sommet initial représentant la position actuelle de l'utilisateur, cet algorithme fait croître un arbre de manière que la somme des poids des arcs soit minimale.

La figure 31 illustre un exemple de 7 sommets pour lesquels nous avons appliqué l'algorithme de Christofides.

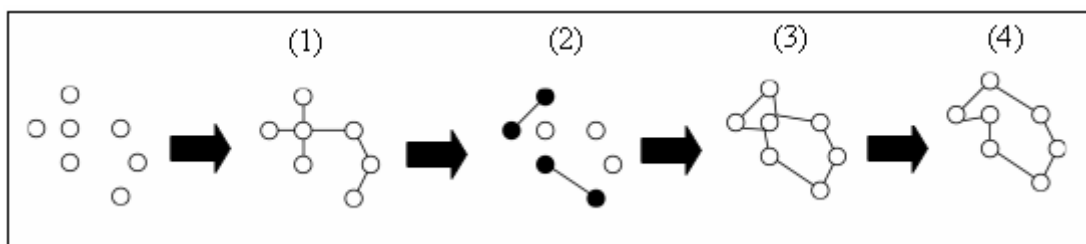


Figure 31 : Exemple de l'algorithme de Christofides

Par ailleurs, puisque notre solution est un chemin hamiltonien et non pas un circuit hamiltonien, nous proposons une modification à la méthode de Christofides qui est la suivante : nous supprimons, de la solution, le dernier arc relié au nœud initial pour obtenir un chemin et on relie le dernier nœud au nœud de la caisse la plus proche.

4.8.1.2.3 La méthode du Plus Proche Voisin

Cet algorithme consiste à construire un cycle en faisant croître une chaîne. On part d'un sommet arbitraire à partir duquel on va au sommet voisin le plus proche, puis de celui-là à son plus proche voisin non visité jusqu'à ce que tous les sommets soient parcourus et on revient au départ. Cependant, les solutions qu'il fournit peuvent être arbitrairement mauvaises. En effet, cette procédure commence généralement par faire de très bons choix en sélectionnant des arêtes de poids faible, mais, vers la fin, la chaîne doit ensuite aller visiter des sommets qui restent, et des distances importantes peuvent alors être rajoutées à la chaîne.

Nous avons testé l'algorithme du Plus Proche Voisin (PPV) ainsi que l'algorithme de Christofides modifié. Nous avons obtenu des solutions réalisables mais pas toujours optimales et des fois même on peut revenir à un rayon déjà visité. Pour améliorer ces solutions et les adapter à la structure d'un magasin, nous avons ajouté, à la méthode du PPV, une règle de passage par tous les articles d'un même rayon avant de passer aux autres.

4.8.1.2.4 La méthode du Plus Proche Voisin Adaptée

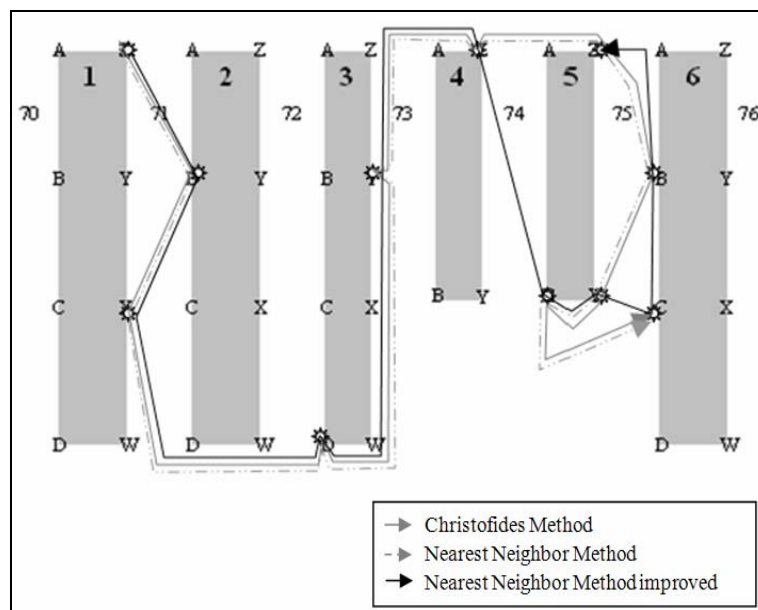


Figure 32 : Amélioration de la solution avec l'algorithme PPVA [HEB10b]

La méthode du Plus Proche Voisin Adaptée est une méthode basée sur l'algorithme du PPV. Nous avons ajouté à ce dernier une règle de passage par tous les articles d'un même rayon avant de passer aux autres. De plus, nous avons proposé de laisser les produits fragiles ainsi que les produits surgelés (cas des magasins agroalimentaires) à la fin du parcours pour ne pas les abîmer. Cette solution peut être, des fois, un peu plus longue que les autres mais elle paraît plus cohérente pour l'utilisateur et répond plus à ses besoins. De plus, à la fin, le client se dirige vers la caisse la plus proche au dernier article visité. La figure 32 illustre les améliorations apportées par cette méthode par rapport aux deux autres méthodes.

Cependant, la méthode du PPV adaptée (PPVA) se base sur une méthode qui commence généralement par faire de très bons choix en sélectionnant des arêtes de poids faible, mais, vers la fin, on doit aller visiter des sommets qui restent, et des distances importantes peuvent alors être rajoutées au chemin. Pour cela, nous proposons d'améliorer la solution en appliquant la méthode de recherche tabou [HKB11a].

4.8.1.2.5 La méthode de Recherche Tabou

La recherche Tabou est une métaheuristique originalement développée par Glover [Glo89], [Glo90]. Elle est basée sur des idées simples, mais reste néanmoins efficace. Cette méthode, développée pour résoudre des problèmes combinatoires, la plupart NP-difficiles, propose de surmonter le problème des optima locaux par l'utilisation d'une mémoire. La méthode tabou est une procédure itérative qui, partant d'une solution initiale, tente de converger vers la solution optimale en exécutant, à chaque pas, un mouvement dans l'espace de recherche. Chaque pas consiste d'abord à engendrer un ensemble de solutions voisines de la solution courante pour ensuite en choisir la meilleure, même si ce choix entraîne une augmentation de la fonction objectif à minimiser. En acceptant de détériorer la valeur de la solution courante, le minimum local peut être évité mais, en contre partie, des parcours répétitifs sont déplorés. Aussi, pour palier l'inconvénient majeur des méthodes de recherche locale, la recherche tabou a pour but d'améliorer à chaque étape, la valeur de la fonction objectif, en utilisant une mémoire afin de conserver les informations sur les solutions déjà visitées. Cette mémoire constitue la liste Tabou qui va servir à interdire l'accès aux dernières solutions visitées. Lorsqu'un optimum local est atteint, il y a interdiction de revenir sur le même chemin.

Un critère d'aspiration, est également utilisé pour lever l'interdiction d'utilisation d'un mouvement si ce dernier conduit à une meilleure solution.

Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles [MC03]. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire : problèmes de routage de véhicules, problèmes d'affectation quadratique, problèmes d'ordonnancement, ...

Il convient maintenant de décrire plus précisément les différentes composantes de la méthode. Ces paramètres portent sur la solution initiale, la structure du voisinage, la restriction tabou et le critère d'aspiration. Nous allons présentons, dans la suite, la mise en application de la méthode de recherche tabou adaptée pour la résolution du problème d'optimisation du parcours dans un hypermarché.

4.8.1.2.5.1 *Solution initiale*

Dans la phase d'initialisation, nous retenons la solution construite grâce à l'algorithme du PPVA. Cette solution respecte les contraintes de passage par tous les articles existants au même rayon et laisse les produits fragiles à la fin du parcours. Puis le processus itératif de recherche tabou des nouvelles solutions admissibles commence, jusqu'à ce qu'un critère d'arrêt soit satisfait.

4.8.1.2.5.2 *Structure du voisinage*

Il est clair que l'exploration d'un voisinage de grande dimension à chaque nouvelle itération du processus augmente à la fois les chances de converger vers l'optimum global et la complexité de la recherche. Afin d'obtenir un équilibre entre la qualité et la rapidité de la recherche, deux mouvements d'échange sont retenus dans la création du voisinage de la solution courante.

- la permutation des deux éléments i et $i+k$ avec k allant de 2 à $N-2$ et N nombre de nœuds
- la permutation des deux éléments i et $N-i-1$

Dans le cas où $k=1$ cela revient à l'inversion de deux éléments successifs dans la solution initiale. La figure 33 représente le premier mouvement dans le cas où $k=1$ et la figure 34 représente le deuxième mouvement.

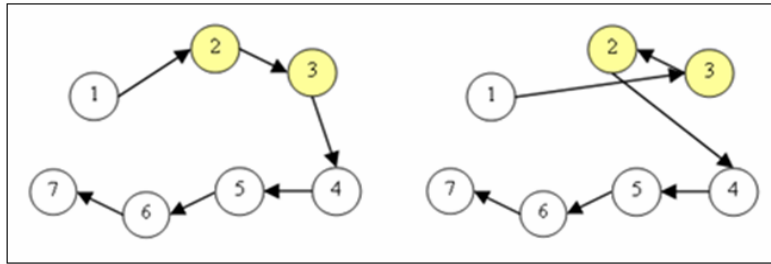


Figure 33 : Mouvement d'échange entre deux éléments successifs

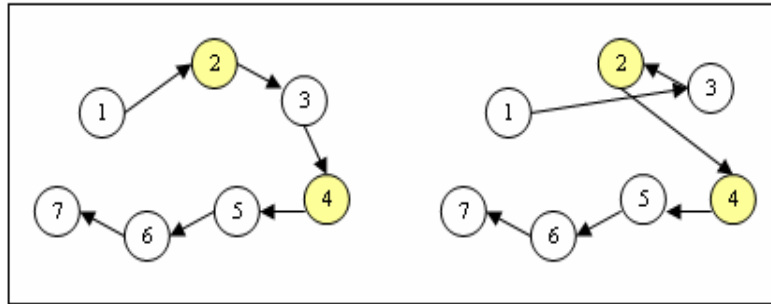


Figure 34 : Mouvement d'échange entre l'élément i et l'élément $N-i-1$

4.8.1.2.5.3 Définition des ensembles

Il faut noter que l'utilisation de ces mouvements fournit un voisinage très peuplé, qui peut contenir des solutions non réalisables qui ne respectent pas la règle de passage aux articles appartenant à la même allée ensuite passer aux autres. Dans le but de diminuer le temps de calcul et pour éviter d'avoir ce type de solutions, nous proposons de vérifier cette règle avant de faire un mouvement d'échange. Il est vrai que cela permettra à l'algorithme d'avoir des solutions réalisables mais cela diminuera l'intérêt des mouvements et par la suite on risque de tomber rapidement dans un minimum local. Pour cela, nous proposons de constituer des ensembles de nœuds. Chaque ensemble constitue un élément qui subit un mouvement. Cet ensemble contient des nœuds appartenant à la même allée et qui peuvent être permutés ou inversés entre eux. Chaque nœud appartenant à la tête de gondole constitue un singleton à part. Ainsi, les mouvements seront effectués entre ensembles, d'une part, et entre les nœuds appartenant à un même ensemble d'autre part.

La figure 35 illustre un exemple de parcours résolu en utilisant la méthode de recherche Tabou. Le coût de cette solution est égal à 132. Nous pouvons remarquer, à travers la figure 26, l'amélioration de cette solution en permutant Les ensembles suivants: $\{Z72\}$, $\{X72, B72\}$, $\{D69BIS\}$, $\{B69\}$, $\{X69, Y69\}$, $\{B68\}$, $\{C101\}$, $\{A67\}$. Le nouveau coût de la solution devient 105.

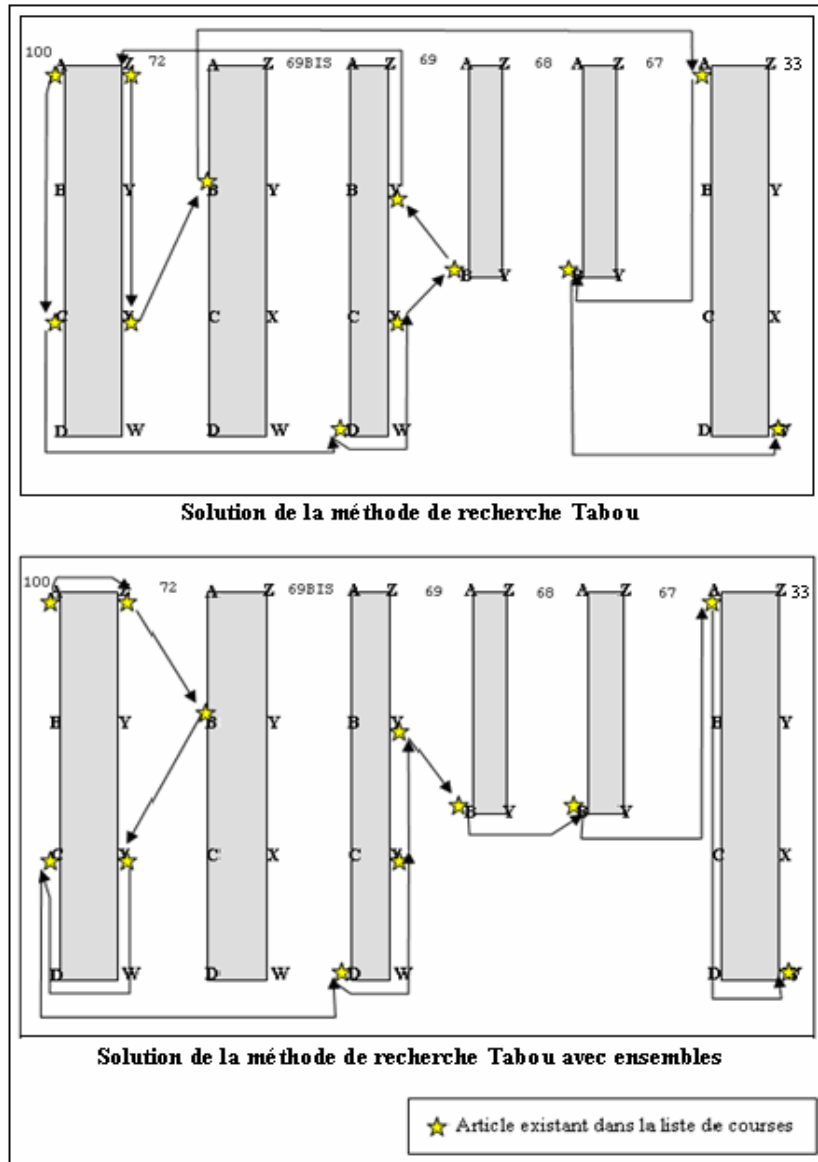


Figure 35 : Amélioration de la solution Tabou après définition des ensembles

4.8.1.2.5.4 Restriction tabou

La restriction tabou que nous avons choisi d'utiliser est compatible avec la construction du voisinage que nous avons retenue. En effet, lorsque l'exécution d'un mouvement est réalisée à l'itération t , un sommet ne peut plus être soumis à un autre mouvement jusqu'à l'expiration de sa tenure. Ainsi les sommets du graphe qui sont retenus pour composer la restriction tabou et qui enregistrent dans la liste tabou sont considérés comme attributs tabou d'une solution.

4.8.1.2.5.5 Critère d'aspiration

Nous rappelons que le critère d'aspiration permet d'éliminer le statut tabou d'une solution. Pour notre algorithme, nous considérons que si une solution tabou est la meilleure

solution rencontrée depuis le début de la recherche, cette solution devient la solution initiale suivante dans le processus itératif de recherche.

La figure 36 représente l'algorithme de recherche tabou adapté au problème d'optimisation de parcours dans un hypermarché.

```

1. Définir un critère d'arrêt de la recherche
2. Prendre pour solution initiale ( $S_0=s$ ) la solution donnée par l'algorithme
   du PPV modifié
3. Initialiser la liste Tabou T (vide)
4. Tant que le critère d'arrêt n'est pas satisfait :
   - Mettre à jour la liste Tabou (s'il y des solutions à libérer)
   - Prendre la meilleure solution  $S_0$  comme solution courante
   - Chercher le voisinage de la solution courante en appliquant les 3
     mouvements d'échange
   - Choisir le mouvement le plus intéressant (celui qui donne le moindre
     coût)
   - Ajouter les solutions Tabou à la liste Tabou T
   - Evaluer l'ensemble des solutions et définir la meilleure solution  $s'$ 
     (avec  $F(s') \leq F(s)$ ) comme nouvelle solution courante
5. Retourner la meilleure solution

```

Figure 36 : Algorithme de recherche Tabou adapté [HKB11a]

4.8.1.3 Etude expérimentale et résultats

Pour le développement des algorithmes utilisés dans nos approches d'optimisation, nous avons utilisé le langage JAVA pour ses divers avantages par rapport aux autres langues de programmation. C'est un langage orienté objet qui permet de créer des programmes avec un code réutilisable. Par ailleurs, l'un des avantages les plus importants de JAVA est son indépendance de la plateforme et sa capacité de se déplacer facilement d'un système informatique à un autre. C'est la garantie de portabilité qui a fait la réussite de JAVA dans les architectures client-serveur, en facilitant la migration entre serveurs. Cette portabilité est assurée par la machine virtuelle JRE (Java Runtime Environment) qui effectue la traduction et l'exécution du bytecode en code natif. Par ailleurs, le langage JAVA a été conçu pour être exploité dans des environnements serveur et distribués. Pour cela, la sécurité a été tenue en compte ce qui permet à JAVA la construction de systèmes inaltérables et sans virus. Le langage JAVA est multithread puisqu'il offre, à un programme, la possibilité d'exécuter plusieurs tâches simultanément. Ce dernier avantage est très important pour nos algorithmes d'optimisation car il permet de diminuer le temps de calcul.

Pour la création de la base de données, nous avons utilisé Microsoft Access qui est un système de gestion de base de données relationnelle. Pour la conception de l'interface

utilisateur et la compilation de l'application, nous avons utilisé l'outil gratuit Eclipse auquel nous avons ajouté Apache Tomcat, un conteneur libre de servlets J2EE (JAVA 2 Enterprise Edition). Par ailleurs, nous avons utilisé le gestionnaire ODBC (Open DataBase Connectivity) pour assurer la communication entre la base de données et l'application.

Les tests ont été effectués sur un ordinateur portable avec un processeur Intel Pentium M à 2,00 Ghz et 1 Go de mémoire vive.

Dans cette section, nous présentons les tests que nous avons effectués afin de bien choisir la tenure et le nombre des itérations nécessaires. Nous avons généré deux types de problèmes classés selon le critère de l'appartenance aux zones du magasin. Nous avons traité des problèmes pouvant avoir 5, 10, 15, 20, 30, 40, 50, 70 et 100 articles à récupérer.

Les caractéristiques de ces problèmes sont les suivantes :

Type P1 : les sommets appartiennent à la même zone.

Type P2 : les sommets appartiennent à des zones différentes

Le choix de P1 et P2 est lié à l'étude de l'impact de la taille du problème (maillage) et de la dispersion des repères sur la solution.

Les tableaux 8 et 9 illustrent les résultats donnés par la méthode de recherche Tabou adaptée (RTA) et la méthode du PPVA pour les problèmes P1 et P2.

Les tests de P1 sont effectués sur la Zone 1, contenant 72 nœuds, et les tests de P2 sur la Zone 1 et Zone 2 qui contiennent au total 149 nœuds. Par ailleurs, nous avons utilisé des threads pour exécuter des mouvements simultanément pour diminuer le temps de calcul.

Pour déterminer le nombre des itérations (NI) ainsi que la valeur de la tenure (LT), nous avons effectué plusieurs tests pour chaque instance afin de trouver les valeurs qui trouvent la bonne solution avec un temps raisonnable. Si cette tenure est trop petite, le risque d'avoir un comportement cyclique du processus de recherche est grand. Si LT est trop grande, il y a un risque d'avoir de nombreux mouvements non évalués, rendant la recherche de l'optimum global excessivement coûteuse en temps de calcul [GHL94], [Gol90]. Les valeurs de NI et LT qui se trouvent dans les tableaux 8 et 9 représentent les valeurs moyennes calculées sur la base des tests effectués pour chaque instance.

P1 : Articles appartenant à la même zone						
N	RTA				PPVA	
	NI moyen	LT moyenne	CT (mètres)	TT (ms)	CPPVA (mètres)	TPPVA (ms)
5	9	4	77	31	85	15
10	10	5	103	85	111	15
15	36	27	121	316	135	16
20	47	28	130	570	159	16
30	42	24	140	688	182	31
40	35	20	164	782	194	46
50	30	16	187	973	209	57
70	10	6	232	1140	236	69

Tableau 8 : Résultats de la méthode de recherche Tabou et la méthode du PPV pour P1

P2 : Articles appartenant à deux zones différentes						
N	RTA				PPVA	
	NI moyen	LT moyenne	CT (mètres)	TT (ms)	CPPVA (mètres)	TPPVA (ms)
5	6	4	83	32	99	15
10	9	4	152	91	256,5	15
15	30	17	165	236	215,5	16
20	34	20	190	357	276,416	16
30	39	23	195	697	263,75	31
40	43	24	196	789	247	37
50	48	27	268	946	298,75	42
70	56	31	263	1232	299	50
100	72	34	362	1941	406,5	56

Tableau 9 : Résultats de la méthode de recherche Tabou et la méthode du PPV pour P2

Avec :

N : le nombre des articles à récupérer ;

NI : le nombre des itérations ;

LT : la tenure ;

CT : le coût de la solution Tabou en mètres ;

TT : le temps de calcul de la solution tabou en millisecondes ;

CPPVA : le coût de la solution PPVA en mètres ;

TPPVA : le temps de calcul de la solution PPVA en millisecondes.

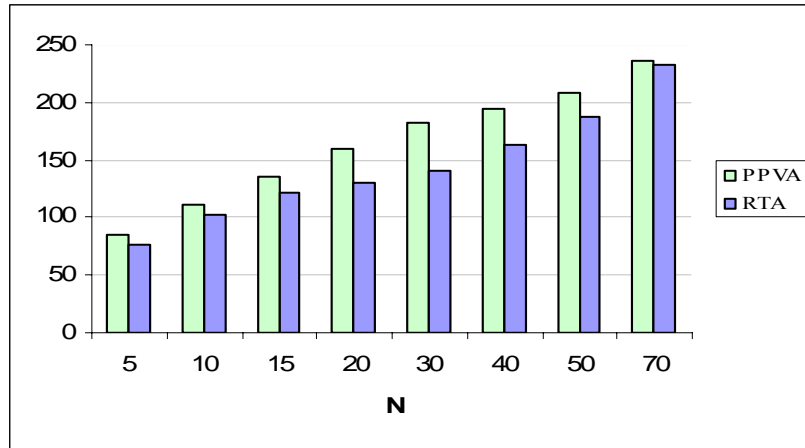


Figure 37 : Comparaison entre la solution tabou et la solution du PPVA pour P1

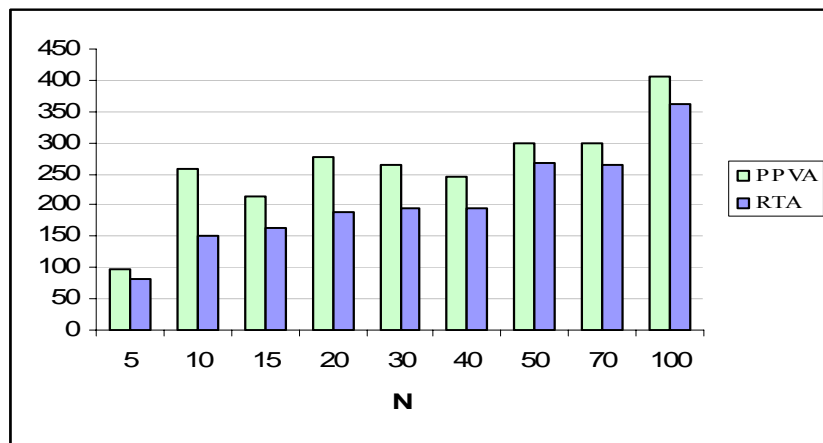


Figure 38 : Comparaison entre la solution tabou et la solution du PPVA pour P2

Impact de la dispersion sur la solution :

Les figures 37 et 38 représentent une comparaison entre la solution tabou et la solution de l'algorithme PPVA. Nous remarquons que pour des instances (70) de taille proche du nombre total des nœuds (72), la solution du PPVA est très proche voire dans la plupart des cas égale à la solution tabou. Cependant, cette instance est très peu réaliste car il est très peu probable que l'utilisateur ajoute tous les produits d'une zone d'où l'intérêt de l'utilisation de la méthode de recherche tabou.

Impact du maillage sur la solution

Le choix du maillage à mettre en place dépend essentiellement de la précision recherchée et du budget de l'entreprise. Dans notre cas, nous avons proposé une distance qui varie entre 2 et 3 mètres selon les dimensions des produits et la longueur de la gondole. Avec ce maillage, nous avons défini 149 repères sur les zones de test de la Zone 1 et la Zone 2.

Plus le maillage est fin plus la taille du problème augmente. D'après les figures 37 et 38, nous remarquons que l'amélioration de la solution apportée par la méthode tabou est plus importante dans le cas où les articles appartiennent à deux zones différentes (taille du problème plus grande). Ainsi, l'application de la méthode de recherche tabou est de plus en plus intéressante quand la taille du problème est importante. D'ailleurs, au départ, nous avons appliqué un maillage plus grossier avec des distances allant de 4 à 6 mètres sur les mêmes zones de test. Les améliorations apportées par la méthode tabou étaient moins importantes.

Convergence de la solution Tabou

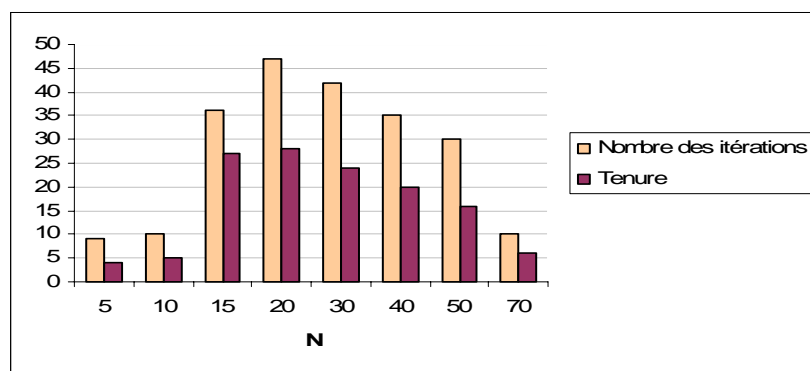


Figure 39 : Convergence de la solution pour P1

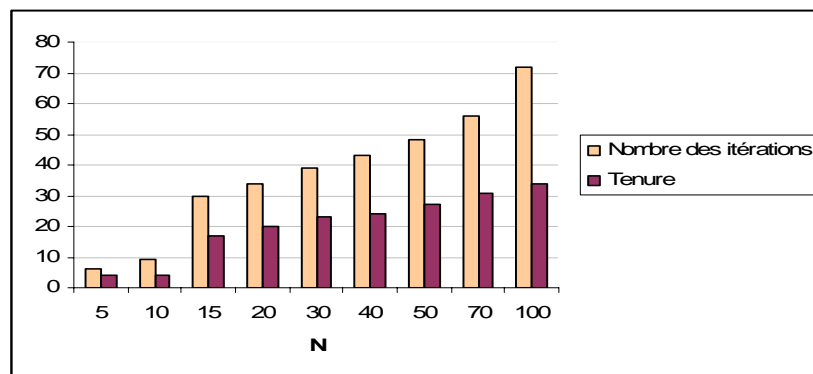


Figure 40 : Convergence de la solution pour P2

D'après les figures 39 et 40, nous remarquons que, pour le problème P1, ou encore dans le cas de la même zone, l'algorithme converge plus rapidement quand les instances sont de petite taille (<10) et quand les instances ou encore le nombre des articles à récupérer s'approche du nombre total des repères de la zone. Dans le cas de la même zone, c'est à partir de 70 articles. Pour les autres instances, la convergence est moins rapide. Etant donné que le temps de calcul reste raisonnable, nous pouvons affirmer que les solutions de la méthode de recherche tabou sont de bonne qualité. Par ailleurs, à travers les tests que nous avons

effectués pour chaque instance, la convergence dépend également de l'emplacement des nœuds par rapport au point initial et au point final. Ainsi, il suffit de changer un nœud de la solution de départ et la solution peut converger moins ou plus vite selon son emplacement.

Détermination du nombre des itérations et de la tenure à appliquer au problème

En se basant sur l'étude de convergence décrite ci-dessus, nous avons fixé pour chaque plage le nombre des itérations et la tenure nécessaires. Ces dernières sont représentées dans les tableaux 10 et 11 en fonction du nombre des articles entrés pour trouver la solution tabou avec ensembles.

P1		
Nombre des articles entrés par l'utilisateur	NI	LT
$N \leq 5$	9	4
$5 < N \leq 10$	10	5
$10 < N \leq 15$	36	27
$15 < N \leq 20$	47	28
$20 < N \leq 30$	42	24
$30 < N \leq 40$	35	20
$40 < N \leq 50$	30	16
$50 < N \leq 70$	2	1

Tableau 10 : Tableau récapitulatif des tenures et du nombre des itérations pour P1

P2		
Nombre des articles entrés par l'utilisateur	NI	LT
$N \leq 5$	6	4
$5 < N \leq 10$	9	4
$10 < N \leq 15$	30	17
$15 < N \leq 20$	34	20
$20 < N \leq 30$	39	23
$30 < N \leq 40$	43	24
$40 < N \leq 50$	48	27
$50 < N \leq 70$	56	31
$70 < N \leq 100$	72	34

Tableau 11 : Tableau récapitulatif des tenures et du nombre des itérations pour P2

4.8.2 Problème d'optimisation de parcours dans un hypermarché pour des articles en promotion avec fenêtre de temps (POPHFT)

Dans cette section, nous avons ajouté, au problème d'optimisation, une contrainte de temps pour certains articles de la liste de courses dans le cas d'une « Vente flash » ou « Promotions pour ouverture exceptionnelle du magasin » qui durent pendant un certain temps et dont la quantité est limitée. Dans ce cas, le système propose à l'utilisateur de choisir les articles concernés qui l'intéressent et les ajouter à sa liste de courses. Par conséquent, l'algorithme d'optimisation contiendra des articles avec des fenêtres de temps et d'autres sans contraintes de temps.

Dans la littérature, plusieurs problèmes de routage avec contraintes de temps ont été évoqués tels que le TSPTW (Traveling Salesman Problem Time Windows), le VRPTW (Vehicle Routing Problem Time Windows), le PDPTW (Pickup and Delivery Problem with Time Windows) et leurs variantes. Dans ces problèmes, les contraintes de temps, appelées fenêtres de temps, sont dues à la disponibilité limitée des clients. Elles peuvent être impossibles à violer, et la contrainte de temps sera une contrainte rigide, comme elles peuvent être violées tout en ayant une pénalité à assumer.

4.8.2.1 Détermination des fenêtres de temps

Une fenêtre de temps correspond à chaque article en promotion représente l'intervalle de temps $[a_i, b_i]$ dans lequel le client doit être présent et même, des fois, il doit être présent pour faire la queue devant cet article pour garantir de le trouver encore disponible. Il faut noter que cet intervalle n'est pas nécessairement le même intervalle de validité de la promotion $[T_i, B_i]$ annoncée par le magasin où T_i représente le début effectif de la promotion et B_i sa fin effective. Si on considère l'intervalle $[T_i, B_i]$ comme fenêtre de temps pour un article en promotion, le risque d'arriver « en retard » est assez grand puisque, dans la plupart du temps, quand l'article en promotion est intéressant et donc très demandé, on trouve une longue queue avant même la date T_i , date de début effective de la promotion. Pour cette raison, nous avons proposé de déterminer un autre intervalle de temps $[a_i, b_i]$ qui augmente la chance d'arriver avant expiration de la quantité limitée. La détermination de cet intervalle dépend, essentiellement, du degré de l'importance de l'article, le besoin du client par rapport à cet article et l'importance de la réduction ainsi que sa quantité disponible. Ainsi, si c'est un article cher et pour lequel la réduction est exceptionnelle et importante, il attirera de

nombreux clients qui peuvent commencer à arriver devant le rayon, dédié à la promotion, même une heure à l'avance. La détermination de cet intervalle peut se baser, aussi, sur l'historique d'achat de cet article et les avis des consommateurs par rapport à cet article qu'on trouve gratuitement sur les forums de consommation.

Généralement, le magasin informe ses clients de la quantité disponible pour chaque article en promotion. La quantité limitée fait, aussi, partie de la stratégie de marketing pour ne pas diminuer son importance et attirer des clients. La distribution de ces articles se base sur le principe FIFO : le premier arrivé sera servi le premier.

Nous supposons que :

- Chaque client n'a le droit qu'à un seul article en promotion avec fenêtre de temps.
- Le client est rationnel. Dès que le nombre des clients qui font la queue est égal à la quantité disponible de l'article en promotion, il saura que faire la queue, en ce moment, est inutile car quand son tour arrivera, le produit sera épuisé. Il doit passer alors à la recherche des autres articles de sa liste.

Les dates a_i et b_i restent des dates approximatives et peuvent s'ajuster au fur et à mesure en se basant sur l'historique de distribution des produits en promotion.

La figure 41 représente les différentes dates de l'article i en promotion avec fenêtre de temps.

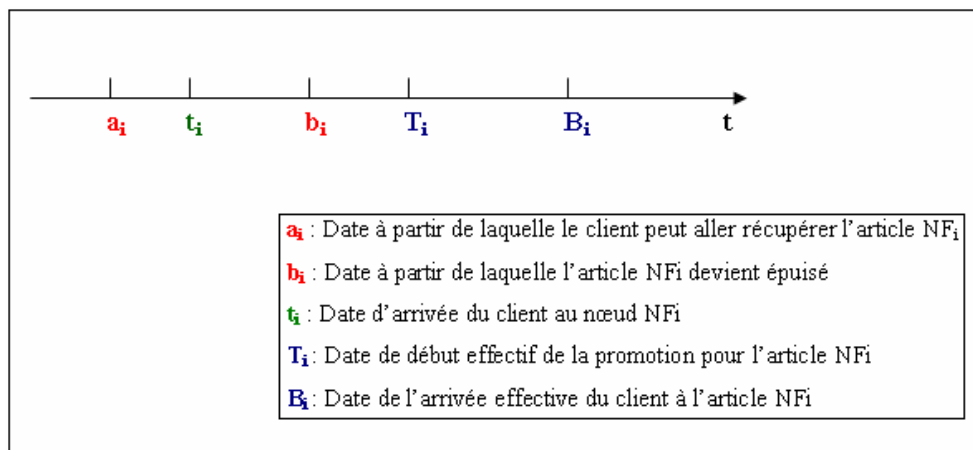


Figure 41 : Différentes dates de l'article i en promotion avec fenêtre de temps

Il faut noter que dans le cas d'une « vente flash » ou une « promotion exceptionnelle » dont la durée et la quantité disponible sont limitées, on trouve un vendeur qui commence à distribuer les articles à la date effective de début de la promotion. On suppose que le temps de service est égal à 2 secondes par client. Ainsi, la date effective de récupération (DE_i) de l'article au nœud i dépend du rang du client dans la queue.

$$Rang = \begin{cases} \frac{t_i - a_i}{b_i - a_i} * Q_{\max} & a_i \leq t_i \\ 0 & \text{Sinon} \end{cases}$$

$$DE_i = T_i + Rang * \text{temps de service}$$

Dans la suite, nous supposons que les valeurs a_i et b_i sont données.

4.8.2.2 Formulation mathématique du problème :

Chaque nœud i , représentant un article en promotion avec fenêtre de temps, doit être visité durant une fenêtre de temps $[a_i, b_i]$ et peut avoir un temps d'attente pour la récupération de l'article à T_i .

Le nœud j représente un article quelconque appartenant à la liste de courses. A chaque arc (i,j) , on associe une distance D_{ij} et un temps de parcours T_{ij} positifs. On fait l'hypothèse que les temps de parcours T_{ij} sont égaux aux distances D_{ij} . Notre problème d'optimisation consiste à trouver le chemin de coût minimum qui commence à la position actuelle de l'utilisateur et finit à la caisse la plus proche et qui passe par chaque nœud de l'ensemble N exactement une seule fois. Sur ce chemin, le nœud i doit être visité au plus tard au temps b_i et ne peut pas être quitté avant le temps DE_i .

La fonction objective consiste à minimiser le temps de parcours.

Le POPHFT est caractérisé par les paramètres suivants :

- N : Nombre des nœuds représentant les articles existants dans la liste de courses du client,
- c : nœud final qui représente la caisse,
- NL : Ensemble des nœuds représentant les articles existants dans sa liste de courses et ne contenant pas une contrainte de temps ($NL \subset N$),
- NL_i : Article au nœud i et ne contenant pas une fenêtre de temps ($NL_i \in NL$),
- NF : Ensemble des nœuds représentant les articles existants dans sa liste de courses et contenant une contrainte de temps ($NF \subset N$),
- NF_i : Article au nœud i et contenant une fenêtre de temps ($NF_i \in NF$),
- T_{ij} : temps mis par l'utilisateur pour aller du nœud i au nœud j ,
- a_i : Date à partir de laquelle le client peut aller récupérer l'article NF_i
- b_i : Date à partir de laquelle l'article NF_i devient épuisé

- t_i : Date d'arrivée du client au nœud NF_i
- ts_i : temps de service au nœud NF_i
- $[a_i, b_i]$: fenêtre de temps de l'article NF_i ,
- DE_i : date effective de récupération de l'article NF_i
- $Q_{\max i}$: quantité maximale de l'article NF_i disponible au départ
- TA_i : temps d'attente au nœud i contenant un article avec fenêtre de temps
- $i = 0..N$: indice des nœuds prédécesseurs,
- $j = 0..N$: indice des nœuds successeurs,
- On suppose que $T_{ij} = D_{ij}$ et T_{ij} inclut le temps de service de l'article j
- Il y a un seul client qui cherche les articles existants dans sa liste de courses,
- Un nœud n'est visité par le client qu'une et une seule fois,
- Il y a un seul nœud de départ qui représente la position actuelle du client,
- Le client commence le trajet à partir de sa position actuelle et se dirige vers la caisse à la fin,

Chaque client a le droit à un seul article par promotion.

$$X_{ij} = \begin{cases} 1 & \text{Si le client part du nœud } i \text{ et se dirige vers le nœud } j \\ 0 & \text{Sinon} \end{cases}$$

$$Y_i = \begin{cases} 1 & \text{Si il y a un temps d'attente au nœud } i \\ 0 & \text{Sinon} \end{cases}$$

La fonction à optimiser est la suivante :

$$\text{Minimiser } F = \sum_{i \in N} \sum_{j \in N} T_{ij} X_{ij} + \sum_{j \in N} TA_j Y_j \quad [1]$$

$$TA_j = DE_j - t_j = (T_j + \frac{t_j - a_j}{b_j - a_j} * Q_{\max j} * ts_j) - t_j$$

Sous les contraintes :

$$\sum_{i \in N} X_{ij} = 1 \quad \forall j \in N \quad [2]$$

$$\sum_{j \in N} X_{0j} = 1 \quad [3]$$

$$\sum_{i \in N} X_{ic} = 1 \quad [4]$$

$$\sum_{i \in N} X_{iu} + \sum_{j \in N} X_{uj} = 2 \quad \forall u \in N \quad [5]$$

La fonction objective [1] consiste à minimiser le temps de parcours. Ce dernier est égal à la somme des coûts des arcs par lesquels passe le client et la somme des temps d'attente devant les articles en promotion.

La contrainte [2] assure que chaque sommet n'est visité qu'une seule fois par le client.

La contrainte [3] assure que le client sort du dépôt une fois.

La contrainte [4] assure que le client se dirige vers la caisse une fois.

La contrainte [5] assure la continuité de la tournée du client : le sommet visité doit impérativement être quitté.

4.8.2.3 Proposition d'une approche de résolution du problème

Pour la résolution de notre problème, nous avons proposé, au départ, un algorithme d'insertion qui part d'une solution qui optimise le parcours, uniquement, pour des articles sans contraintes de temps ensuite insère les autres articles en fonction de leur fenêtre de temps. Ensuite, nous avons proposé d'étudier le problème global en utilisant la méthode de recherche tabou. Dans la suite, nous présentons les détails de ces méthodes ainsi que les résultats obtenus.

4.8.2.3.1 La méthode d'insertion

Cet algorithme fonctionne de la manière suivante : on part d'un tour réduit à quelques sommets, puis on sélectionne un sommet hors du tour que l'on insère entre deux sommets du tour de manière à augmenter le moins possible la longueur totale de celui-ci. On insère ainsi tous les sommets jusqu'à obtenir un circuit.

Le principe de cet algorithme est le suivant : on fixe le point initial qui correspond à la position actuelle du client et le point final qui correspond à la caisse. Ensuite, on part d'un tour réduit aux sommets représentant les articles sans contrainte de temps. Ce tour est déterminé en utilisant la méthode du Plus Proche Voisin Adapté (PPVA) citée dans la section précédente. Puis, on sélectionne un sommet représentant un des articles avec fenêtre de temps que l'on insère entre deux sommets du tour de manière à respecter la fenêtre de temps correspondante et augmenter le moins possible le coût total du parcours. On insère ainsi tous

les sommets jusqu'à obtenir un circuit. Le choix du sommet à insérer se fait en ordonnant les intervalles $[a_i, b_i]$ de manière à ne pas dépasser la date b_i en arrivant devant l'article en question.

La figure 42 représente l'algorithme d'insertion adapté (AIA) au POPHFT.

Dans le cas où deux ou plusieurs articles en promotion ont des fenêtres de temps superposées, le système demande à l'utilisateur de les ordonner en fonction de ses préférences et l'algorithme d'optimisation se basera sur cet ordre pour aller chercher l'article choisi.

Vu que l'algorithme d'insertion adapté ne permet pas l'optimisation globale du problème, nous avons proposé de lui appliquer l'algorithme de recherche Tabou [HKB11b].

```

1. Prendre pour solution initiale (S0) la solution donnée par
   l'algorithme PPVA pour les articles sans fenêtre de temps
2. Initialiser la liste des articles de la nouvelle solution NS
   (vide)
3. Initialiser HA à l'heure actuelle de début de parcours
3. Parcourir la liste de la solution initiale
4. Si ((HA+coûtArcAB < bi) avec i l'article avec FT pas encore
   inséré et AB un arc de S0)
       Mettre à jour la liste NS (en ajoutant B)
   Sinon
       Ajouter i à la liste NS ensuite insérer B
5. Actualiser HA
6. Retourner la nouvelle solution
    
```

Figure 42: Algorithme d'insertion adapté au POPHFT

4.8.2.3.2 La méthode de Recherche Tabou adaptée au POPHFT

Cette méthode a le même principe que celui décrit dans la section précédente sauf que, après avoir choisi le mouvement intéressant, elle élimine les solutions ne respectant pas les fenêtres de temps [HKB11b].

La figure 43 représente l'algorithme de recherche tabou adapté (ARTA) au problème d'optimisation avec des articles en promotion avec fenêtres de temps (POPHFT).

1. Définir un critère d'arrêt de la recherche
2. Prendre pour solution initiale ($S_0=s$) la solution donnée par l'algorithme d'insertion adapté
3. Initialiser la liste Tabou T (vide)
4. Tant que le critère d'arrêt n'est pas satisfait :
 - Mettre à jour de la liste Tabou (s'il y des solutions à libérer)
 - Prendre la meilleure solution S_n comme solution courante
 - Chercher le voisinage de la solution courante en appliquant les 3 mouvements d'échange
 - Choisir le mouvement le plus intéressant (celui qui donne le moindre coût)
 - Eliminer les solutions ne respectant pas les fenêtres de temps
 - Ajouter les solutions Tabou à la liste Tabou T
 - Evaluer l'ensemble des solutions et définir la meilleure solution s' (avec $F(s') \leq F(s)$) comme nouvelle solution courante
5. Retourner la meilleure solution

Figure 43 : Algorithme de recherche tabou adapté au POPHFT

Quand le client commence son parcours, le calculateur d'itinéraire compare le temps estimé au temps effectif et met à jour toutes les dates. Ainsi, dans le cas où le client passe beaucoup de temps à choisir ou hésiter devant des articles, le système lui indique le temps restant pour aller récupérer l'article en promotion.

4.8.2.4 Etude expérimentale et résultats

Nous avons effectué des tests pour la validation des algorithmes d'optimisation pour le POPHFT. Nous avons créé, dans la base de données, une table pour représenter les données concernant les articles en promotion avec fenêtre de temps.

Le tableau 12 illustre un exemple de 3 articles en promotion dont les références sont : B72, D101 et X33. Nous rappelons que :

a_i : Date à partir de laquelle le client peut aller récupérer l'article i

b_i : Date à partir de laquelle l'article i devient épuisé

$Q_{\max i}$: Quantité maximale de l'article i disponible au départ

T_i : Date du début effectif de la promotion

RepereArticle	a_i	b_i	$Q_{\max i}$	T_i
B72	12:00:00	12:21:00	15	12:30:00
D101	12:20:00	12:30:00	7	12:35:00
X33	12:25:00	12:35:00	10	12:40:00

Tableau 12 : Caractéristiques des articles en promotion avec fenêtre de temps

Nous avons fait une comparaison entre les résultats donnés par l'algorithme d'insertion adapté et ceux de l'algorithme de recherche tabou adapté au problème POPHFT sur 8 instances : 5, 10, 15, 20, 30, 50, 80 et 100 articles.

Pour toutes ces instances, nous avons posé l'heure de début du parcours (HA) égale à 12:20:53.

Le tableau 13 illustre les résultats donnés par la méthode de recherche Tabou adaptée au POPHFT.

N	AIA		ARTA	
	CAIA (s)	TAIA (ms)	CARTA (s)	TARTA (ms)
5	942	44	828	52
10	985	45	904	83
15	1081	47	979	97
20	1093	49	1001	198
30	1189	72	1067	393
50	1269	96	1185	516
80	1297	126	1202	964
100	1352	131	1294	1612

Tableau 13: Tableau récapitulatif des résultats de l'AIA et de l'ARTA

Avec :

N : le nombre des articles à récupérer ;

CARTA : le coût de la solution Tabou en secondes ;

TARTA : le temps de calcul de la solution tabou en millisecondes ;

CAIA : le coût de la solution de l'algorithme d'insertion adapté en secondes ;

TAIA : le temps de calcul de la solution de l'algorithme d'insertion adapté en millisecondes.

Nous remarquons que l'algorithme de recherche tabou a apporté des améliorations à la solution. La figure 44 représente ces améliorations avec un temps de calcul qui reste raisonnable (moins de 2 secondes pour 100 articles).

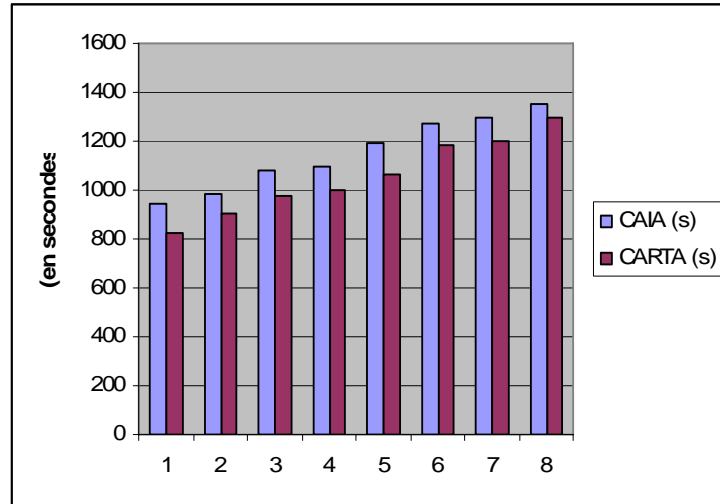


Figure 44 : Amélioration apportée par la méthode de recherche tabou adaptée

4.9. Gestion de la partie dynamique de guidage

Dans cette partie, nous proposons des solutions de re-calculation du chemin en cas de modification de la liste des courses et en cas de changement de direction.

4.9.1 Cas de modification de la liste de courses

La figure 45 illustre le diagramme UML de la dynamique applicative du système i-GUIDE en cas de modification de la liste de courses après optimisation.

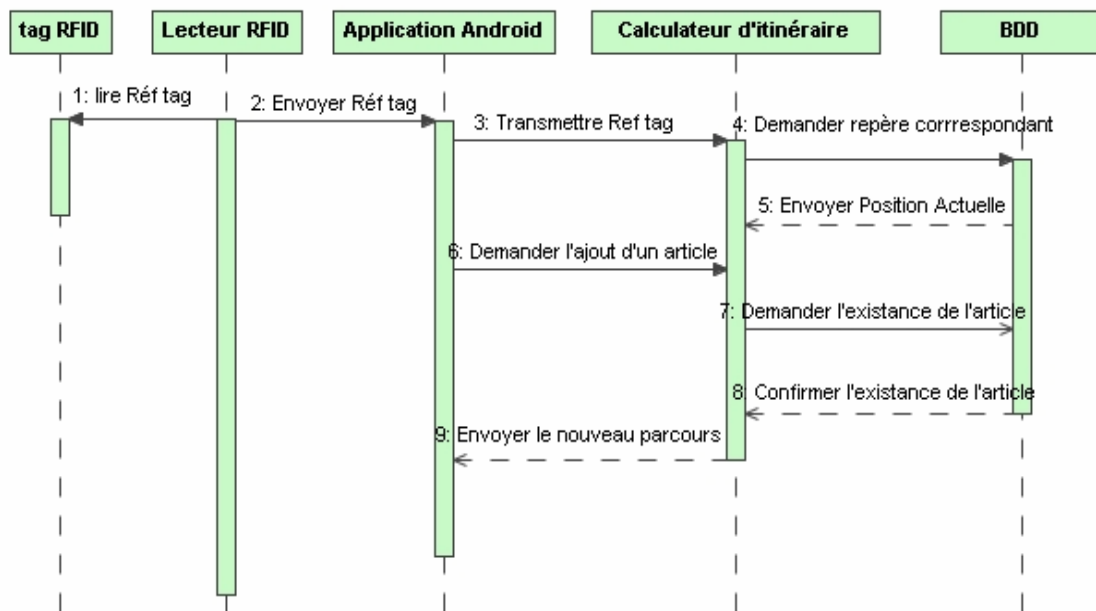


Figure 45 : Modélisation UML de la dynamique applicative du système en cas de modification de la liste de courses

A chaque fois que le client passe par un article de la liste, le système lui demande de le récupérer et le supprime de la liste. Si le client édite sa liste de courses, un nouveau chemin sera recalculé. Ce dernier contient tous les articles à chercher, et ne contient pas les articles déjà récupérés. La figure 46 illustre le cas de modification d'une liste de courses : Quand un article est ajouté à la liste, on cherche la meilleure position à l'insérer (cas (2)). Lorsqu'un article est supprimé, on l'élimine de la solution tabou et on relie son prédécesseur à son successeur (cas (3)).

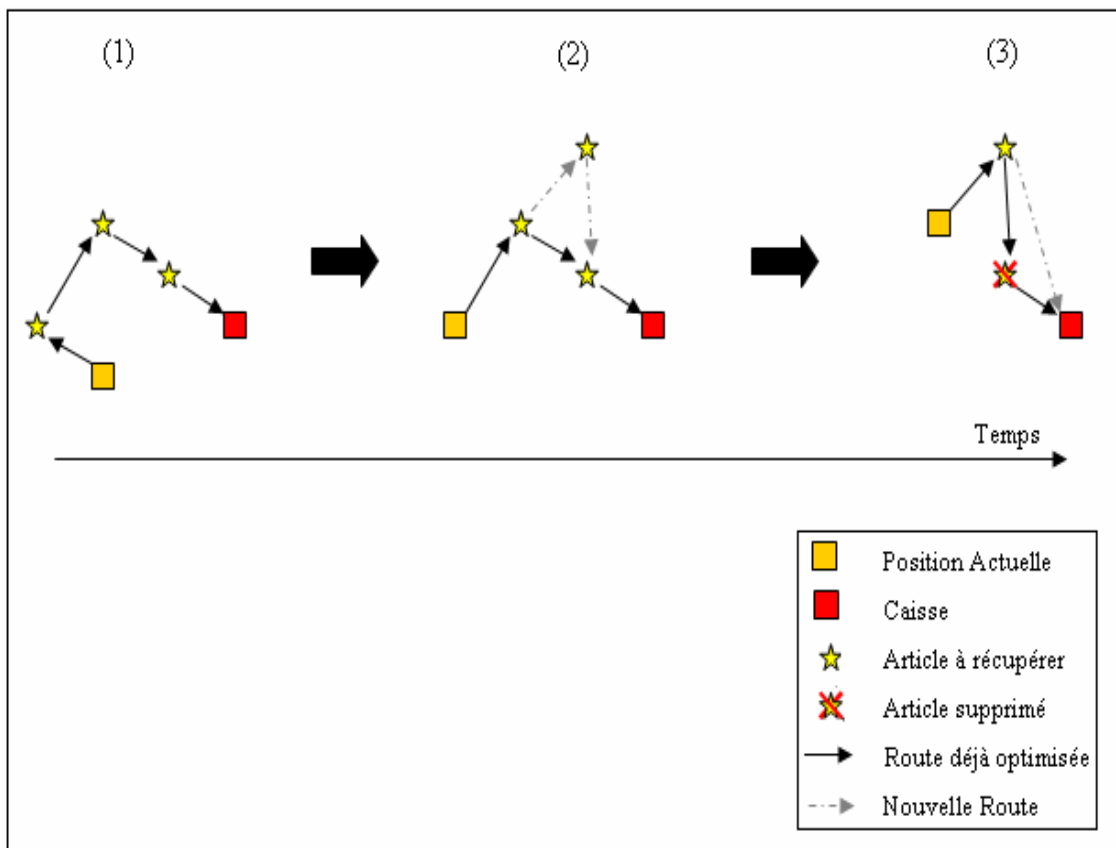


Figure 46 : Résolution du problème en cas de modification de la liste de courses

Par ailleurs, la figure 47 décrit l'algorithme d'insertion d'un article ajouté à la liste de courses après optimisation avec la méthode de recherche tabou adaptée.

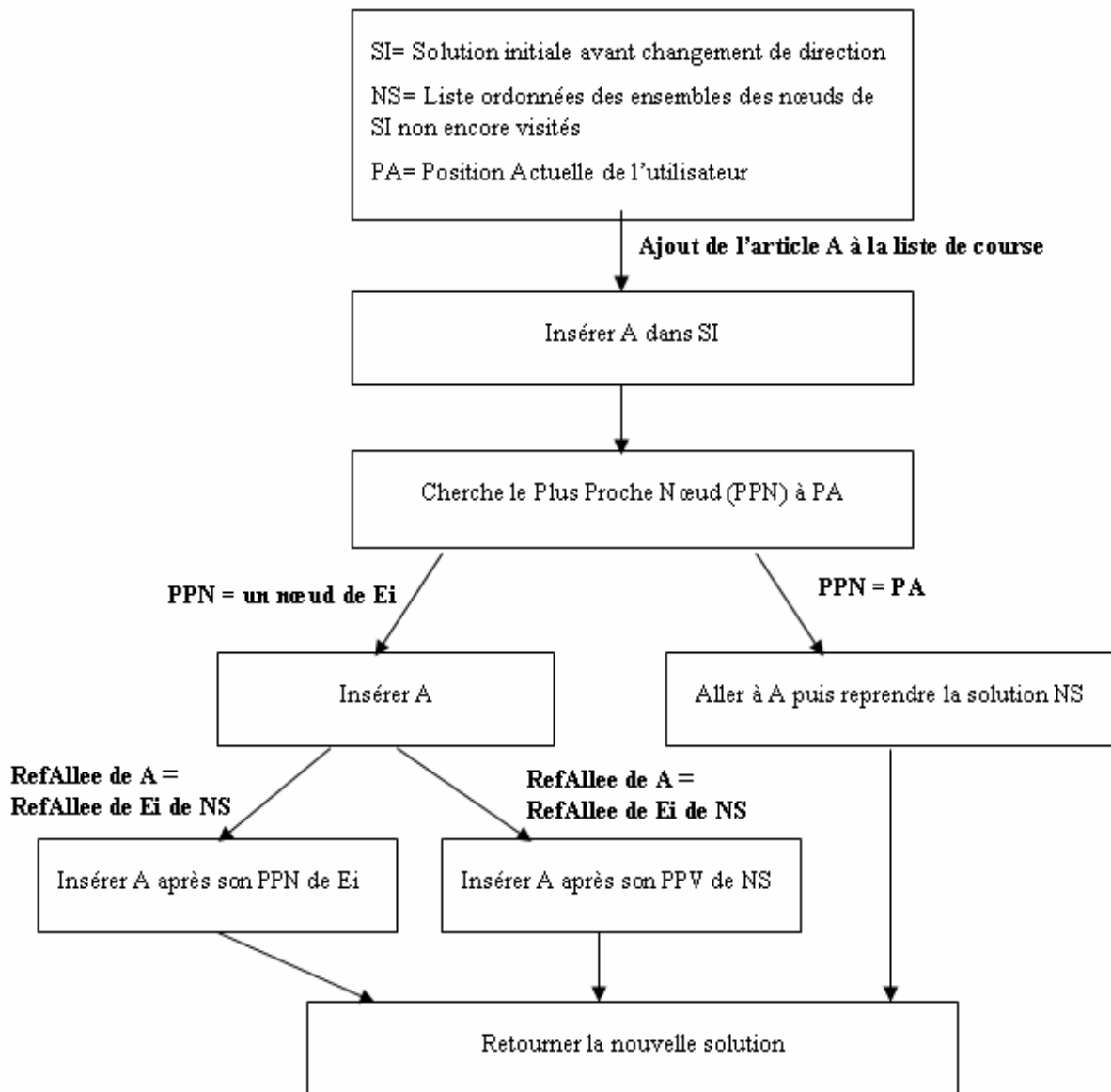


Figure 47 : Algorithme d'insertion d'un article ajouté à la solution de RTA

4.9.2 Cas de changement de direction

La figure 48 illustre une modélisation UML de la dynamique applicative du système du système i-GUIDE en cas de changement de direction.

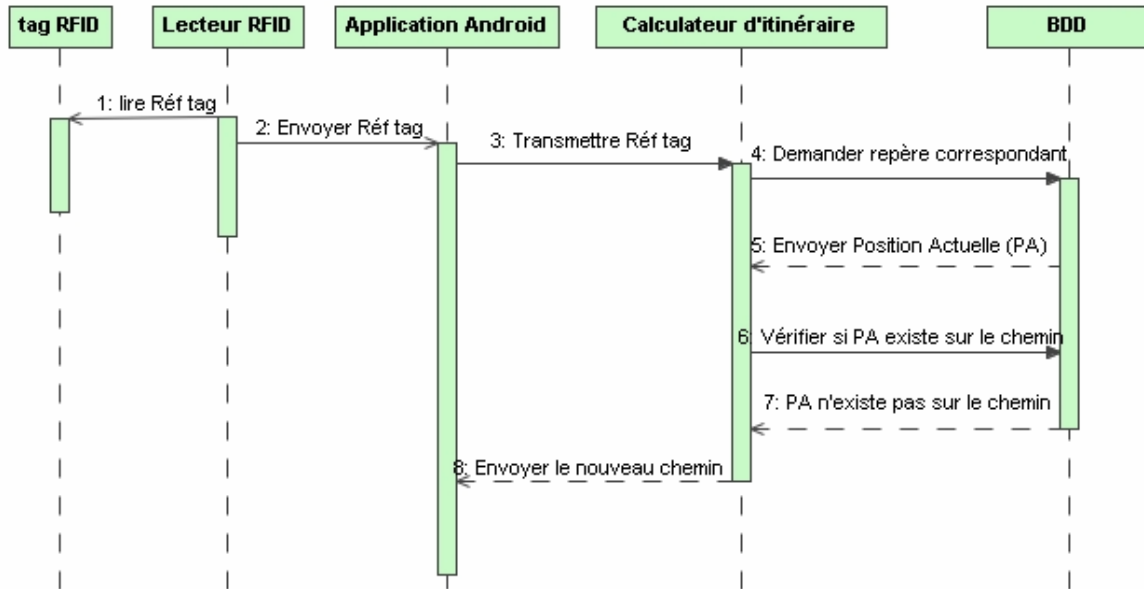


Figure 48 : Modélisation UML de la dynamique applicative du système en cas de changement de direction

Dans la base de données, lors du calcul des distances, nous avons déterminé les repères RC qui définissent le chemin par lequel passe l'utilisateur pour aller d'un point à un autre dans le magasin.

Nous rappelons que notre système de guidage permet de détecter la position actuelle de son utilisateur grâce aux étiquettes RFID. Le maillage choisi dépend essentiellement de la précision recherchée et du budget de l'entreprise. Dans notre cas, nous avons choisi le critère de précision et nous avons proposé une distance qui varie entre 2 et 3 mètres selon la longueur de la gondole et les dimensions des produits.

Si le client ne suit pas le chemin indiqué et change de direction, un nouveau chemin lui sera proposé en fonction de sa position actuelle et ceci comme suit :

1. Dans un magasin, pour passer d'un article à un autre, il faut suivre un chemin. La détermination de ce chemin revient à préciser les repères RC par lesquels passe le client pour aller d'un article à un autre.
2. Le système détecte un changement de direction en cas où la position actuelle de l'utilisateur ne correspond pas à l'un de ces repères ou les repères voisins.
3. Nous avons fixé une zone de tolérance d'un rayon de 3 mètres car l'utilisateur peut aller prendre un article qui existe sur la gondole devant sans changer de direction. Ainsi la distance entre un repère RC et son repère voisin ne doit pas dépasser 3 mètres.

4. En cas de dépassement des 3 mètres, le système propose à l'utilisateur d'ajouter ses nouveaux articles pour recalculer le nouvel itinéraire à parcourir. En cas de non réponse, il considère sa position actuelle comme nouvel article et recalcule dynamiquement le nouveau chemin.

Ainsi, le cas de changement de direction ressemble au cas d'ajout d'un nouvel article mais sans l'intervention de l'utilisateur. C'est le système qui entre sa position actuelle comme « nouveau article ajouté » ensuite lui demande de retourner au chemin déjà calculé. S'il prend un autre chemin, l'algorithme cherche le plus proche article à sa position actuelle et recalcule le chemin en utilisant l'heuristique du plus proche voisin ou une heuristique basée sur la recherche arborescente des échanges appliquées aux ensembles de nœuds qui étaient définis précédemment lors de l'exécution de l'algorithme de recherche tabou. Ces algorithmes n'ont pas été testés. Ils seront traités et analysés dans des travaux futurs.

Dans le cas où le système ne détecte pas un tag RFID, il considère le dernier tag jusqu'à détecter le suivant. Si ce dernier ne fait pas partie du chemin déjà défini, on considère que l'utilisateur a changé de direction.

4.10. Conclusion

Dans ce chapitre, nous avons proposé des approches d'optimisation de parcours dans un hypermarché pour deux problèmes. Pour le premier problème, nous avons choisi le critère de la plus courte distance pour la détermination du chemin et pour le deuxième nous avons ajouté une contrainte de temps pour les articles en promotion.

Avant de chercher le chemin le plus court à parcourir pour trouver les articles existants dans la liste de courses, nous avons proposé une méthode pour la détermination des distances entre les articles de l'hypermarché pris deux à deux. Etant donné que notre problème ressemble à celui du PVC, nous avons testé l'algorithme de Branch and Bound et l'algorithme de Christofides auquel nous avons ajouté des modifications au niveau du nœud final ainsi que l'algorithme du PPV pour déterminer le chemin le plus court à parcourir. Ces deux dernières méthodes ont donné des résultats satisfaisants mais pas toujours optimaux avec un inconvénient de revenir à un rayon déjà visité. Pour cela, nous avons proposé d'ajouter, à l'algorithme au PPV, une règle de passage par tous les articles d'un même rayon avant de passer aux autres pour améliorer les résultats. De plus, nous avons proposé de laisser les produits fragiles ainsi que les produits surgelés (cas des magasins agroalimentaires) à la

fin du parcours pour ne pas les abîmer. Cependant, la méthode du PPV adaptée donne des solutions acceptables mais qui peuvent être améliorés. Pour cela, nous avons proposé d'améliorer la solution en appliquant la méthode de recherche tabou. Nous avons utilisé 2 mouvements d'échange pour la construction de l'ensemble du voisinage et nous avons défini des ensembles pour garantir l'application de la règle de passage par tous les articles d'un même rayon. Cette méthode a été testée pour deux types de problèmes P1 et P2 classés selon le critère de l'appartenance aux zones du magasin et a donné de bons résultats avec un temps raisonnable. Nous avons considéré un magasin Leroy Merlin qui nous a fourni des données réelles pour valider nos applications et tests.

Nous avons, ensuite, ajouté au problème une contrainte de temps pour des articles en promotion qui durent pendant un certain temps et dont la quantité est limitée. Par conséquent, le problème d'optimisation contient des articles sans contraintes de temps et d'autres avec fenêtres de temps. Pour résoudre ce problème, nous avons proposé d'insérer les articles en promotion dans la solution PPVA qui contient seulement les articles de la liste de courses sans contrainte de temps. Ensuite, nous avons amélioré cette solution avec la méthode de recherche tabou. Cette dernière a été testée et a donné des résultats raisonnables pour des articles allant jusqu'à 100.

Par ailleurs, étant donné que la liste de courses peut être modifiée, nous avons proposé une méthode d'insertion pour recalculer le nouveau chemin en cas d'ajout ou de suppression des articles de la liste de courses. De plus, nous avons défini un champ de tolérance, en le dépassant, on considère que le client a changé de direction. Dans ce cas, le système lui propose de retourner au chemin déjà défini ou lui propose un nouveau s'il s'est éloigné.

CONCLUSIONS ET PERSPECTIVES

Les travaux de recherche de ce mémoire ont porté sur l'étude de faisabilité technique et logicielle du système i-GUIDE, système interactif de guidage des personnes dans les hypermarchés.

Dans le premier chapitre, une revue sur les projets de guidage réalisés ou en cours de réalisation a montré l'inexistence d'un système dynamique qui répond à ce besoin d'où l'intérêt de notre étude. Ensuite, nous avons détaillé l'analyse fonctionnelle du besoin du système. Dans notre étude, nous traitons des personnes avec handicap visuel et d'autres à mobilité réduite. Ces personnes n'ont pas les mêmes besoins ni les mêmes contraintes. Par conséquent, nous avons commencé par l'étude des besoins de chaque cas. Ensuite nous avons recommandé un système global qui répond à leurs attentes et aux personnes sans handicap.

Dans le deuxième chapitre, nous avons effectué une étude de l'existant, en prenant le cas des magasins Leroy Merlin, pour une meilleure intégration de notre système dans un hypermarché. Pour cela, nous avons modélisé la chaîne logistique d'achat et de vente des produits dans un magasin pour la détermination des acteurs qui interagissent avec notre système. Cette modélisation a été faite en utilisant le diagramme BPMN et nous a permis de dégager des informations desquelles nous avons tenu compte telles que l'absence d'un adressage par produit dans la base de données, l'application du commerce de changement pour attirer les clients, la difficulté de faire une mise à jour de l'emplacement des produits surtout pour un grand nombre de références des produits... d'où la nécessité de concevoir une base de données facile à mettre à jour. Ensuite, nous avons opté pour l'approche UML pour cerner les principales fonctionnalités de notre système ainsi que les objets nécessaires pour son bon fonctionnement et étudier les interactions entre notre système temps réel et son environnement extérieur.

Dans le troisième chapitre, nous avons proposé un prototype basé sur la technologie RFID couplé au réseau WiFi. Ce prototype est composé d'un serveur contenant la base de

données et le calculateur d'itinéraire, des étiquettes RFID et un terminal client contenant le lecteur RFID. Nous avons choisi la technologie RFID pour localiser l'utilisateur dans l'hypermarché et détecter sa position quand il change de direction. Nous utilisons le Wifi installé dans les magasins Leroy Merlin pour assurer la communication entre le serveur et le terminal client. Ensuite, nous avons proposé une application sous Android où le calculateur et la base de données sont hébergés dans un serveur. Ceci permettra de soulager l'application mobile et rendre son utilisation pratique. Ainsi, le client aura la possibilité de la télécharger sur son smartphone et l'utiliser à l'entrée du magasin.

Dans le quatrième chapitre, nous avons proposé des approches d'optimisation de parcours dans un hypermarché pour deux problèmes. Pour le premier problème, nous avons choisi le critère de la plus courte distance pour la détermination du chemin et pour le deuxième nous avons ajouté une contrainte de temps pour les articles en promotion. Avant de chercher le chemin le plus court à parcourir pour trouver les articles existants dans la liste de courses, nous avons proposé une méthode pour la détermination des distances entre les articles de l'hypermarché deux à deux. Cette méthode se base sur une base de données facile à mettre à jour suite à toute modification de l'organisation de l'hypermarché. Pour faciliter le calcul de ces distances, nous avons divisé le magasin en zones selon la disposition des gondoles et regroupé les articles dans des repères sur les gondoles. Après une modélisation de problème, nous avons testé l'algorithme de Branch and Bound et l'algorithme de Christofides ainsi que l'algorithme du PPV pour déterminer le chemin le plus court à parcourir. Ces deux dernières méthodes ont donné des résultats satisfaisants mais pas toujours optimaux avec un inconvénient de revenir à un rayon déjà visité. Pour cela, nous avons proposé d'ajouter, à l'algorithme au PPV, une règle de passage par tous les articles d'un même rayon avant de passer aux autres pour améliorer les résultats. De plus, nous avons proposé de laisser les produits fragiles ainsi que les produits surgelés (cas des magasins agroalimentaires) à la fin du parcours pour ne pas les abîmer. Cependant, la méthode du PPV adaptée se base sur une méthode qui commence généralement par faire de très bons choix en sélectionnant des arêtes de poids faible, mais, vers la fin, les sommets qui restent peuvent engendrer des distances importantes. Pour cela, nous avons amélioré la solution en appliquant la méthode de recherche tabou. Cette méthode a été testée pour deux types de problèmes P1 et P2 classés selon le critère de l'appartenance aux zones du magasin et a donné de bons résultats avec un temps raisonnable.

Nous avons, ensuite, ajouté au problème une contrainte de temps pour des articles de la liste de courses dans le cas d'une « Vente flash » ou « Promotions pour ouverture exceptionnelle du magasin » qui durent pendant un certain temps et dont la quantité est limitée. Pour résoudre ce problème, nous avons proposé une approche d'optimisation basée sur une méthode d'insertion et une méthode de recherche tabou. Cette dernière a été testée et a donné des résultats raisonnables pour des articles allant jusqu'à 100. Par ailleurs, nous avons proposé une méthode de ré-optimisation du chemin en cas de modification de la liste des courses ou en cas de changement de direction.

Plusieurs perspectives de recherche sont envisagées à court et moyen terme :

- Développer l'application sous Android proposée dans le chapitre 3 et créer un site web dynamique permettant à l'utilisateur de préparer sa liste de course et de calculer le chemin à parcourir.
- Intégrer une carte SD de lecture de puces RFID (ou similaire) avec une portée suffisante permettant de faire des tests dans des conditions réelles en magasin. L'élaboration d'un prototype spécifique dédié aux personnes handicapées sera également envisagé (gros boutons, ergonomie adaptée, infographie attrayante...)
- Déterminer une méthode générique de calcul des distances entre deux articles appartenant à deux zones différentes en se basant sur des méthodes de recherche du plus court chemin dans le cas où il existe plusieurs possibilités de chemins à parcourir.
- Approfondir l'étude d'optimisation avec fenêtres de temps en utilisant des historiques enregistrés par le biais du prototype i-GUIDE permettant de générer de façon plus fiable les paramètres de temps en se basant sur la capitalisation des connaissances et des enseignements de notre système intelligent
- Considérer plusieurs clients s'entraînant afin de minimiser la durée de parcours pour récupérer des articles existants dans la même liste de courses (u-shopping collaboratif). Ce problème peut s'avérer similaire au problème VRP avec plusieurs véhicules.
- Prendre en compte les contraintes de capacité des chariots ou des véhicules utilisés pour le transport de la marchandise tout en étudiant les stratégies de remplissage liées à la nature des produits. Cela revient à résoudre un problème de bin packing tout en optimisant le chemin à parcourir dans le magasin.

- Envisager proposer une étude de marché dans le cadre d'une dimension plutôt commerciale par le biais d'une étude de marché sur l'impact de ces technologies sur la stratégie commerciale et la politique de vente/achat de l'enseigne.

BIBLIOGRAPHIE

- [ABCC07] D. Applegate, RE. Bixby, V. Chvatal et W. Cook, *The Traveling Salesman Problem: a Computational Study*, Princeton University Press, Princeton, 2007.
- [BT85] E. Balas et P. Toth, *Branch and Bound Methods for the Traveling Salesman Problem*", in G. Lawler, J.K. Lenstra, A. H. G. Rinnooy Kan, D. Shmoys, eds., *The Traveling Salesman Problem*, J. Wiley, Chichester, 1985.
- [Bell86] R.E. Bellman, *The Bellman continuum*, Editions Robert S. Roth, 1986.
- [BG05] O. Bräysy, M. Gendreau, *Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms*, *Transportation Science*, vol39 (1), pp 104-118, 2005.
- [Bia00] L. Bianchi, *Notes on Dynamic Vehicle Routing - The State of the Art*, Technical Report, IDSIA-05-01, 20 December 2000.
- [Boo91] G. Booch, *Object Oriented Design with Applications*, ed. Benjamin-Cummings, 1991.
- [Bou98] M.E. Bouzgarrou, *Parallélisation de la méthode du Branch and Cut pour résoudre le problème du voyageur de commerce*, thèse en Informatique à l'Institut National Polytechnique de Grenoble, soutenue le 14 décembre 1998.
- [CDDSS01] J.-F Cordeau, G. Desaulniers, J. Desrosiers, M. Solomon, and F. Soumis. "The VRP with Time Windows". Technical Report, Département de Mathématiques et de Génie Industriel, Ecole polytechnique de Montréal, Canada, June 2001.

- [CDM92] A. Colomi, M. Dorigo, and V. Maniezzo, *Distributed optimization by ant colonies*, Proceeding of the first European Conference on Artificial Life (ECAL 91), pp. 134-142, 1992.
- [CMSAB09] U. B. Ceipidor, C. M. Medaglia, A. Serbanati, G. Azzalin, M. Barboni, F. Rizzo, M. Sironi, *SeSaMoNet: an RFID-based economically viable navigation system for the visually impaired*, International Journal of RF Technologies: Research and Applications, 1754-5749, Vol 1, Issue 3, pp. 214– 224, 2009.
- [CMRS06] U. B. Ceipidor, C. M. Medaglia, F. Rizzo, A. Serbanati, *Radio Virgilio/sesamonet: an RFID-based Navigation system for visually impaired*, Mobile guide 06, Turin, Italie, 18 Octobre 2006.
- [Che05] D. Chen, *Modélisation d'entreprise pour l'intégration et l'interopérabilité des systèmes industriels*, Mémoire d'Habilitation à Diriger des Recherches, Université Bordeaux 1, soutenue le 4 décembre 2005.
- [Chr76] N. Christofides, *Worst-case analysis of a new heuristic for the travelling salesman problem*, Report 388, Graduate School of Industrial Administration, CMU, 1976.
- [DBDG10] E. Delage, N. Bostel, P. Dejax, M. Gendreau, *La ré-optimisation des tournées de techniciens dans un environnement dynamique avec des temps de service stochastiques*, Journée du GT Transport et Logistique et du Projet France-Québec Tournées à Hautes Technologies, 14 Octobre 2010, Angers, France.
- [HKBK08] I. Harbaoui Dridi, R. Kammarti, P. Borne, M. Ksouri, *Un Algorithme génétique pour le problème de ramassage et de livraison avec fenêtres de temps à plusieurs véhicules*, CIFA, 11-14 Septembre 2008, Bucarest, Roumanie.

- [DP03] B. Dobing et J. Parsons, *The role of use cases in the UML: a review and research agenda*, Advanced topics in database research, Idea Group Publishing, Hershey, vol 1, 2003.
- [Dou98] B. Douglass, *Real-time UML: Developing efficient objects for embedded system*, Addison Wesley Longman, 1998.
- [Dur94] N. Durand, J.M. Alliot et J. Noailles, *Algorithmes génétiques : un croisement pour les problèmes partiellement séparables*, Journées Evolution Artificielle Francophones, JEAFF, Toulouse, 1994.
- [DVZC92] G. Doumeingts, B. Vallespir, M. Zanettin, D. Chen, *GRAI Integrated Methodology : A Methodology for Designing CIM Systems*, version 1.0, GRAI/LAP, University Bordeaux I, 1992.
- [DV04] G. Doumeingts, B. Vallespir, *Modélisation d'entreprise : vers le système d'information*, Actes de l'école de modélisation d'entreprise, Ecole des Mines d'Alès, 2004.
- [PKPLC10] E. Piraino, T. King, J. Pascual, G. Lefranc, J. Ceroni. Performance Study of RFID Feedback in MRP Systems, ACCA Congress 2010, Santiago, Chili.
- [Els10] A. El-Sherbeny, *Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods*, Journal of King Saud University, vol 22, pp123–131, 2010
- [Fra07] J. François, *Planification des chaînes logistiques : modélisation du système décisionnel et performance*, Thèse de doctorat de l'université de Bordeaux I , décembre 2007.
- [Gan09] D. Gance, *Retour sur l'expérimentation du système de guidage dynamique BlueEyes dans les espaces souterrains du métro et du RER*, Séminaire Seniors & Mobilité 2009, Paris France, 1er Décembre 2009.
- [Gar09] F. Garin, *ANDROID : Développer des applications mobiles pour les Google Phones*, DUNID, 2009.

- [GCLLB99] R. Gelin, F. Coulon-Lauture, B. Lesigne, J.M. LeBlanc, M. Busnel, *AFMASTER : An industrial rehabilitation workstation*, International Conference on Rehabilitation Robotics, ICORR'99, 1-2 Juillet 1999, Stanford, CA.
- [GCR10] D. Guignard, J. Chable, E. Robles, *Programmation android, De la conception à la publication*, Eyrolles, 2010.
- [GGPS98] M. Gendreau, F. Guertin, J.-Y. Potvin et R. Séguin, *Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries*, Technical Report, CRT-98-10, Centre de Recherche sur les Transports, Université de Montréal, 1998.
- [GGPT99] M. Gendreau, F. Guertin, J.-Y. Potvin et E. Taillard, *Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching*, *Transportation Science* 33 (4), pp. 381–390, 1999.
- [GJ79] M. R. Garey et D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, 1979.
- [GHL94] M. Gendreau, A. Hertz, G. Laporte, *A tabu search heuristic for the vehicle routing problem*, *Manag. Sci.*, vol. 40, pp. 1276- 1290, 1994.
- [Glo89] F. Glover, *Tabu search - part I*, *ORSA Journal on Computing*, 1(3), pp. 190–206, 1989.
- [Glo90] F. Glover, *Tabu search - part II*, *ORSA Journal on Computing*, 2(1), pp. 4–32, 1990.
- [GROCDML03] L.M. Gambardella, A.E. Rizzoli, F. Oliverio, N. Casagrande, A.V. Donati, R. Montemanni and E. Lucibello, *Ant Colony Optimization for vehicle routing in advanced logistics systems*, IDSIA, Galleria 2, 6928 Manno, Switzerland and AntOptima, via Fusion 4, 6900 Lugano, Switzerland, 2003.

- [HKYDG08] I. Hadj Khalifa, A. El Kamel, P. Yim, A. Demeunynck et J-P. Grassien, *Modélisation du processus de brouettage des conteneurs*, Conférence Internationale Francophone d'Automatique, CIFA2008, 3-5 Septembre, Bucarest, Roumanie, 2008.
- [HEYDG09] I. Hadj Khalifa, A. El Kamel, P. Yim et A. Demeunynck, J-P. Grassien, *Modélisation du processus de brouettage des conteneurs*, article sélectionné à la conférence CIFA2008 et publié dans la revue e-STA, CIFA 2008 N°3, pp 20-25, 2009.
- [HEY11] I. Hadj Khalifa, A. El Kamel et P. Yim, *Transportation process of containers BPMN-modeling and transformation into ACTIF model*, accepté et en cours de publication dans ROMJIST 2011.
- [HEB10a] I. Hadj Khalifa, A. El Kamel et B. Barfety, *Supply chain design for the conception of a real time intelligent navigation system inside hypermarkets*, International Conference on Information Systems, Logistics and Supply Chain, ILS 2010, 13-16 Avril, Casablanca, Maroc.
- [HEB10b] I. Hadj Khalifa, A. El Kamel et B. Barfety, *Optimisation de parcours en temps réel dans un hypermarché*, ROADEF 2010, 24-26 Février, Toulouse, France.
- [HEB10c] I. Hadj Khalifa, A. El Kamel et B. Barfety, *Real time indoor intelligent navigation system inside hypermarkets*, Large Scale Systems: Theory and Applications, LSS2010, 11-14 Juillet 2010, Lille, France.
- [HKB11a] I. Hadj Khalifa, A. El Kamel et B. Barfety, *Approches de modélisation UML et optimisation de l'itinéraire dans un hypermarché*, soumis à Journal Européen des Systèmes Automatisés(JESA), 2011.
- [HKB11b] I. Hadj Khalifa, A. El Kamel, *Itinerary optimisation approach with time windows inside hypermarkets*, soumis à IEEE International Conference on Systems, Man, and Cybernetics (IEEE-SMC2011), 9-12 Octobre 2011, Anchorage, Alaska.

- [HKB11c] I. Hadj Khalifa, A. El Kamel, *Android-based autonomous system with path optimization*, soumis à International Conference on Intelligent Systems Design and Applications, ISDA2011, 22-24 Novembre 2011, Córdoba, Espagne.
- [HK69] Held M. et Karp R.M., *The Traveling Salesman Problem and Minimal Spanning Trees*, Operations Research 18, pp.1138-1969.
- [Hol75] J.H. Holland, *Adaptation in natural and artificial systems*, Michigan Press Univ., Ann Arbor, MI, 1975.
- [Hou05] H. Housroum, *Une approche génétique pour la résolution du problème VRPTW dynamique*, Thèse en génie informatique et automatique à l'université d'Artois, Soutenue le 3 Mai 2005.
- [JCJO92] I. Jacobson, M. Christerson, P. Jonson, G. Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, ed. Addison-Wesley, 1992.
- [JM02] D.S. Johnson et L.A. McGeoch, *Experimental Analysis of Heuristics for the STSP, The Traveling Salesman Problem and its Variations*, Gutin and Punnen (eds), Kluwer Academic Publishers, 2002, pp. 369-443.
- [KPS98] P. Kilby, P. Prosser, and P. Shaw, *Dynamic VRPs: A Study of Scenarios*, CSIRO, Canberra ACT 2601, Australia, September 1998.
- [Kul07] V. Kulyukin, *Robot-Assisted Shopping for the Blind: Haptic and Locomotor Spaces in Supermarket*, Proceedings of the AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics Stanford University, pp. 36-38. Palo Alto, California, March 26-28, 2007.
- [LPS03] P. Lacomme, C. Prins and M. Sevaux, *Algorithmes de graphes*, Eyrolles 2e édition 2003, ISBN 2-212-11385-4.

- [Lap92] G. Laporte. *The Travelling Salesman Problem: An overview of exact and approximate algorithms*, European Journal of Operational Research, vol. 59, pp. 231–247, 1992.
- [LAR01] A. Larsen, *The Dynamic Vehicle Routing Problem*, Thèse de doctorat, IMM, Technical University of Denmark, 2001.
- [Lat06] T. Lattanzio, *Caractérisation des entreprises organisées en gestion par affaire*, thèse en Génie Industriel à l’Ecole Nationale Supérieure d’Arts et Métiers, soutenue le 26 octobre 2006.
- [LK95] P. Loucopoulos et V. Karakostas, *System requirements engineering*, McGraw-Hill International series in Software Engineering Version 1.0., 1995
- [MC03] M. MABED, J.P. CAMPAGNE, *Approche Distribuée pour la Résolution de Problèmes Combinatoires*, Conférence Francophone de MOdélisation et SIMulation, MOSIM’03, 23-25 Avril 2003, Toulouse, France.
- [Mec95] R. Mechti, *Tournée de clients à la demande : problèmes et Méthodes*, Technical report, Laboratoire PRiSM, Université de Versaille-St. Quentin, France, 1995.
- [Nie10] F. Niedercorn, *Faber Novel apporte aux grands comptes des services de mobilité*, les Echos, publié le 28 Juin 2010.
- [OMG05] OMG, *Unified Modeling Language: Superstructure*, version 2.0, Aout 2005, disponible sur le site officiel d’UML.
- [OMG09] OMG, *Business Process Modelling Notation Specification*, Version Mai 2009.
- [OTeN07] L’Observatoire des Territoires Numériques (OTeN), *Handi Cité*, Version décembre 2007, disponible sur le lien http://www.oten.fr/IMG/article_PDF/article_3821.pdf, dernière consultation le 01/04/2011.

- [PH04] H. Pérez-Ponce and P.R. Hernandez-Rodriguez, *Navigation system for visual impaired persons based on satellital location*, Proceedings of the 26th Annual International Conference of the IEEE EMBS San Francisco, CA, USA, Septembre 1-5, 2004.
- [Pie90] H. Pierreval, *Les méthodes d'analyse et de conception des systèmes de production*, Edition Hermès, 1990.
- [Pou07] K. Poupée, *Live japon : le corps humain, un réseau sans fil ?*, disponible sur <http://www.clubic.com/actualite-82592-live-japon-corps-humain-reseau-fil.html>, dernière date de consultation le 01/04/2011.
- [Ral02] T.K. Ralphps, *Parallel branch and cut for capacitated vehicle routing*, Parallel Computing archive, Volume 29, Issue 5, 2 December 2002.
- [RBPEL91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, *Object-Oriented Modeling and Design*, ed. Prentice-Hall, 1991
- [Rob93] M. Roboam, *La Méthode GRAI : Principes, Outils, Démarche et Pratique*, Editions Teknéa, 1993.
- [Ron05] Roque M, 2005. *Contribution à la définition d'un langage générique de modélisation d'entreprise*. Thèse de doctorat de l'Université Bordeaux 1, mention productique, 15 novembre 2005.
- [RJB05] J. Rumbaugh, I. Jacobson et G. Booch, *UML 2.0 Guide de référence*, Second edition. Edition Campus-Press, 2005.
- [RPAVTDD06] A. Ramudhin, M. Paquet, A. Artiba, D. Varvaro, V. Thomson, P. Dupre, M. Dumouchel, *A generic framework to support the selection of an RFID based control system with application to the MRO activities of an aircraft engine manufacturer*, Production Planning and Control, ed. Taylor and Francis, 2006.

- [RRLF84] S. Roy, J.-M. Rousseau, G. Lapalme and J. A. Ferland, *Routing and scheduling of transportation services for disabled: summary report*, Technical report, Centre de recherche sur les transports, Université de Montréal, June 1984.
- [Sau11] G. Sauvanet, *Recherche de chemins multiobjectifs pour la conception et la réalisation d'une centrale de mobilité destinée aux cyclistes*, thèse en Informatique à l'Université François Rabelais de Tours, soutenue le 5 avril 2011.
- [Stp04] A. Stephen White, *Introduction to BPMN*, IBM Corporation, 2004
- [Tai93] E. D. Taillard, *Parallel iterative search methods for vehicle routing problems*, Networks, vol. 23, pp. 661-673, 1993
- [TBGGP97] E. D. Taillard, P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin. "A tabu search heuristic for the vehicle routing problem with soft time windows". *Transportation Science*, pp. 170-186, 1997.
- [Tai98] E. D. Taillard, M. Gambardella, M., Gendreau, J.-Y. Potvin, *Programmation à mémoire adaptative*, Technical report IDSIA-79-97, IDSIA, Lugano, Calculateurs Parallèles, Réseaux et Systèmes Répartis 10, pp. 117-140, 1998.
- [Tas06] R. Tassinari, *Pratique de l'analyse fonctionnelle*, Dunod, 4e édition, 2006.
- [TF09] S. Y. Turki, S. Faïz, *Apport des règles d'association spatiales pour l'alimentation automatique des bases de données géographiques*, *Revue Internationale de Géomatique*, Vol 19, pp. 27-44, 2009.
- [Whi09] A. S. White, *Introduction to BPMN*, IBM Corporation, 2009.
- [Wid01] M. Widmer, *Les Métaheuristiques : Des outils performants pour les problèmes industriels*, 3ème Conférence Francophone de MODélisation et SIMulation MOSIM'01, 25-27 avril 2001, Troyes.

- [XK96] J. Xu, J. P. Kelly, *A network flow-based tabu search heuristic for the vehicle routing problem*, Transportation Science 30, 379-393, 1996.

WEBOGRAPHIE

- [AAK04] M. Abboud, L.M. Abou Jaoude et Z. Kerbage, *Real Time GPS Navigation System*, 2004, disponible sur <http://webfea-lb.fea.aub.edu.lb/proceedings/2004/SRC-ECE-27.pdf>, dernière date de consultation le 04/04/2011.
- [Alt11] P. Alto, *Google's Android becomes the world's leading smart phone platform*, Canalys research release 2011/013, publié le 31 janvier 2011, disponible sur <http://www.canalys.com/pr/2011/r2011013.pdf>, dernière date de consultation le 20/04/2011.
- [BPMN06] BPMN, *Business Process Modelling Notation Specification*, Février 2006, disponible sur <http://www.bpmn.org>, site officiel de BPMN, dernière date de consultation le 03/04/2011.
- [CaB10] V. Capron et R. Bachelet, *Guide de l'Analyse Fonctionnelle*, 2010, disponible sur http://rb.ec-lille.fr/l/Analyse_fonctionnelle/RemiBacheletGuideDelAF.PDF, dernière date de consultation le 02/04/2011.
- [Lou08] R. Loukil, *Les réseaux corporels sans fils à l'étude en France*, publié sur le site <http://www.industrie.com> le 16 octobre 2008, dernière date de consultation le 02/04/2011.
- [Mab09] S. Mabile, *La ratp expérimente le projet blueeyes, un système innovant de guidage audio et visuel par téléphone mobile*, disponible sur <http://www.blueeyes.fr/>, 2009, dernière date de consultation le 02/04/2011