



HAL
open science

Intelligence Ambiante Pro-Active : de la Spécification à l'Implémentation

Patrick Reignier

► **To cite this version:**

Patrick Reignier. Intelligence Ambiante Pro-Active : de la Spécification à l'Implémentation. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 2010. tel-00529415

HAL Id: tel-00529415

<https://theses.hal.science/tel-00529415>

Submitted on 25 Oct 2010

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

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

Université Joseph Fourier
- Grenoble -

Rapport scientifique présenté
pour l'obtention d'une
Habilitation à Diriger des Recherches

Spécialité : Informatique

Patrick Reignier

Sujet du mémoire

Intelligence Ambiante Pro-Active
de la Spécification à l'Implémentation

Soutenue le 16 Septembre 2010

devant le jury

Président : Marie-Christine Rousset (Professeur UJF)
Rapporteurs : Bernt Schiele (Professeur Technische Universität Darmstadt)
Jean Vanderdonckt (Professeur Université catholique de Louvain)
Olivier Boissier (Professeur ENSM Saint-Etienne)
Examineurs : Olivier Sigaud (Professeur UPMC)
James L. Crowley (Professeur Grenoble INP)

Université Joseph Fourier
Laboratoire LIG

Inria Rhône Alpes
655 Avenue de l'Europe
Montbonnot
38 334 Saint Ismier Cedex France
Téléphone : 04 76 61 55 11

Table des matières

1	Contexte général : l'informatique ambiante	1
1.1	1991 : l'informatique ambiante	1
1.2	1995 : l'informatique calme	3
1.3	1998 : l'intelligence ambiante	3
1.4	Premier bilan	7
1.5	Notre approche	9
2	Modèle de contexte	13
2.1	Quelques définitions	13
2.2	Notre modèle formel de contexte	14
2.3	Quelle approche pour la spécification ?	16
2.4	Représentationnel vs Interactionniste	20
2.5	De la spécification à l'implémentation	21
I	Point de vue développeur	23
3	Interprétation de la perception	25
3.1	Rôles et relations	25
3.2	Apprentissage supervisé des rôles	26
3.2.1	Apprentissage supervisé de classifieurs	26
3.2.2	Classifieur Bayésien	26
3.2.3	Support Vector Machine	30
3.2.4	Approche hybride pour la prise en charge de nouvelles données	32
4	Reconnaissance de scénarios	35
4.1	Transformation de modèles	35

4.2	Réseaux de Petri	36
5	Méta modèle : RdP synchronisé	41
5.1	Calcul de la synchronisation	41
5.2	Application : acquisition automatique de séminaire	42
5.2.1	Evaluation expérimentale	46
5.3	Bilan	48
6	Méta-modèle : RdP flou synchronisé	49
6.1	Exemples d'approches	49
6.2	Relations floues	50
6.3	Réseau de Petri flou synchronisé	51
6.3.1	Situation et transition floue	51
6.3.2	Evaluation d'une situation	52
6.4	Evaluation d'un contexte	52
6.4.1	Mesure floue	54
6.4.2	λ -mesure floue	55
6.4.3	Intégrale de Choquet	55
6.4.4	Exemple d'évaluation d'un scénario	56
6.5	Expérimentation	56
7	Apprentissage des situations	61
7.1	Principe	61
7.2	Segmentation automatique	62
7.2.1	Divergence de Jeffrey	62
7.2.2	Détection multifenêtres des pics	64
7.2.3	Choix des segments	65
7.2.4	Evaluation	65
7.3	Représentation des situations	69
7.3.1	Principe	69
7.3.2	Exemple : la reconnaissance de séquences de vidéo surveillance	70
7.4	Bilan	71
8	Modèle attentionnel	73
8.1	L'attention implicite	73
8.2	Les systèmes attentionnels	75
8.2.1	Détection de l'état attentionnel	75
8.2.2	Stratégies de présentation	76
8.3	Attention : cause ou conséquence ?	78
8.4	Modèle attentionnel proposé	79
8.4.1	Métaphore gravitationnelle	79
8.4.2	Distribution des ressources attentionnelles	81
8.4.3	Stratégies d'interaction	84

8.5	Validation expérimentale	85
II	Point de vue utilisateur	89
9	Apprentissage supervisé	91
9.1	Pourquoi l'apprentissage automatique?	91
9.2	Cadre général	92
9.3	Retour de l'utilisateur	92
9.4	Approche générale proposée	93
9.5	Evaluation	95
9.5.1	Scénario initial	95
9.5.2	Retour de l'utilisateur	96
9.5.3	Représentation d'une situation	96
9.5.4	Classes d'algorithmes d'apprentissage	97
9.5.5	Résultat	100
9.6	Bilan	100
10	Apprentissage renforcé	103
10.1	Apprentissage renforcé	103
10.1.1	Principe	103
10.1.2	Quelques exemples	105
10.1.3	Environnement partiellement observable	106
10.2	Apprentissage renforcé indirect	107
10.2.1	Principe	107
10.3	Notre approche	108
10.3.1	Choix du formalisme	108
10.3.2	Vue d'ensemble	109
10.3.3	Etat	111
10.3.4	Collecte du renforcement	113
10.3.5	Modèle de l'environnement	114
10.3.6	Apprentissage du modèle de l'environnement	116
10.3.7	Apprentissage non interactif (planification)	118
10.3.8	Enquête grand public	118
10.3.9	Validation Expérimentale	120
III	And now something totally different Jones et al. (n.d.)	127
11	Direction de recherche et perspectives	129
11.1	Bilan	129
11.1.1	Scénarios, pro-activité, engagement utilisateurs	130
11.1.2	Ethique, respect de la vie privée	131

11.2	Quelle direction ?	134
11.2.1	Principes	134
11.2.2	Confidentialité de mes données	136
11.2.3	Schéma général	138
12	Références bibliographiques	141
IV	Annexes	153
A	ARéVi	155
A.1	Avant Propos	155
A.2	Introduction	155
A.3	Les besoins d'un système de réalité virtuelle	156
A.3.1	Modularité	156
A.3.2	Immersion par le langage	157
A.3.3	Généricité	157
A.4	ARéVi 4.0	157
A.4.1	Présentation générale	158
A.4.2	Architecture logicielle	158
A.4.3	Le langage oRis	159
A.5	Exemples	162
A.5.1	Les trains d'engrenages	163
A.5.2	Les réseaux de Petri	163
A.5.3	Couplage engrenages et réseau de Petri	164
A.5.4	Autres exemples d'applications	165
A.6	Conclusion et perspectives	167
B	Curriculum Vitae des activités de recherche et d'enseignement	169
B.1	Formation	169
B.2	Activité professionnelle	169
B.3	Activité de recherche	170
B.3.1	Evolutions de ma thématique de recherche	170
B.3.2	Thématique actuelle	170
B.3.3	Participation à des projets de recherche	171
B.3.4	Relations industrielles	174
B.3.5	Encadrement	174
B.3.6	Dissémination	176
B.3.7	Review, comités de programmes, comités d'organisation	179
B.3.8	Autre	180
B.4	Activité d'enseignement	180
B.4.1	Matières enseignées	180
B.4.2	Responsabilités d'enseignement	180

B.4.3	Responsabilité de filière	182
B.4.4	Conférences pédagogiques	182
B.5	Autres responsabilités collectives	183
B.5.1	Vie de l'UFR	183
B.5.2	Vie du laboratoire	183
C	Liste des publications	185

Table des figures

1.1	Smart Classroom de l'Université de Tsinghua (extrait de (Shi et al., 2003))	6
1.2	Classification de systèmes d'Intelligence Ambiante. Le disque correspond à la zone où l'on cherche à se positionner (repris de (Cheverst et al., 2005)).	11
2.1	La situation 3 va apparaître lors de l'échange entre la situation 1 et la situation 2	17
2.2	Exemple d'images provenant du système de montage	18
2.3	Modèle de contexte du système d'acquisition de séminaire	19
2.4	Architecture générale de nos applications contextuelles	22
3.1	Ellipse de détection des cibles en mouvement	28
3.2	Rôles détectés	28
3.3	Classification par un hyperplan (et sa marge) de deux classes	31
3.4	Classifieur Hybride	33
5.1	Modèle de contexte du système d'acquisition	45
5.2	Réseau de Petri du système d'acquisition de séminaire	46
5.3	Exemple de règles Jess produite à partir du réseau de Petri synchronisé	47
6.1	Exemple de relation floue	51
6.2	Exemple de transition floue entre situations	53
6.3	Exemple d'évaluation d'un scénario	57
6.4	Degré de reconnaissance de "Leaving a Bag" sur l'ensemble des vidéos	58
6.5	Degré de reconnaissance de "Browsing" sur l'ensemble des vidéos	59
7.1	Découpage en situations afin de maximiser la différence de distribution de deux situations adjacentes	62
7.2	Calcul à chaque instant t de la divergence entre les deux intervalles autour de t (pour trois tailles différentes d'intervalles).	63
7.3	Distance normalisée entre deux pics.	64

7.4	Choix du meilleur sous-ensemble de pics afin d'obtenir la séparation la plus "franche" des situations	65
7.5	Exemple de configurations de groupes d'interaction	67
7.6	Situations réelles vs situations détectées	68
7.7	Situations réelles et détectées pour un séminaire	69
7.8	Exemples d'images extraites des vidéos Caviar	70
8.1	(1) A entend la voiture, (2) B entend la même voiture, (3) intelligibilité mutuelle entre A et B	75
8.2	Définition de la force gravitationnelle	80
8.3	Effet de la gravité sur la trajectoire d'une masse	80
8.4	Distribution de la ressource attentionnelle en fonction de la focalisation c	82
8.5	Distribution des ressources attentionnelles que P1 affecte à P2, P3, O1 et O2	83
8.6	Effet Steinzor lors d'une réunion en cercle	87
9.1	Exemple de redécoupage d'une situation en 3 sous-situations	94
9.2	Algorithme général pour le redécoupage de situations	95
9.3	Scénario Initial	96
9.4	Exemple d'apprentissage par <i>find-S</i>	97
9.5	Exemple d'apprentissage par Candidate Elimination	98
9.6	Exemple d'apprentissage par <i>ID3</i>	99
10.1	Méthode d'apprentissage renforcé indirect : DYNA. L'apprentissage s'effectue tour à tour avec le monde réel et le modèle du monde.	108
10.2	Modèle de l'environnement	115
10.3	Influence du choix de l'état initial avec 100 itérations par épisode et des événements tirés au hasard dans la base de données	121
10.4	Influence du nombre d'itérations par épisode avec état initial et événements tirés au hasard dans la base de données.	122
10.5	Notes des Q-tables produites par chaque épisode d'apprentissage par renforcement exécuté en parallèle d'interactions avec l'assistant.	123
10.6	Notes des Q-tables produites par chaque épisode d'apprentissage par renforcement. La baisse correspondant à un changement d'avis de l'utilisateur. La remontée correspond à l'adaptation du système.	124
A.1	Principe d'ARéVi 4.0	158
A.2	Extrait du diagramme OMT du noyau d'ARéVi	159
A.3	Diagramme OMT des classe prédéfinies oRis pour ARéVi	162
A.4	Train d'engrenages dynamique.	164
A.5	Représentation graphique d'un réseau de Petri	165
A.6	Association d'un comportement à une place du réseau de Petri	166

Liste des tableaux

3.1	Nombre d'images et de rôles par séquence	29
3.2	Distribution à l'intérieur d'une séquence	29
3.3	Matrice de confusion de l'apprentissage Bayésien	29
3.4	Matrice de confusion pour le SVM	31
3.5	Taux de reconnaissance des trois classifieurs	34
4.1	Transformation Allen → Petri : meet, before, overlaps et starts	37
4.2	Transformation Allen → Petri : equals, during et finishes	38
5.1	Événements associés aux opérateurs meet, before, overlaps et starts	43
5.2	Événements associés aux opérateurs : equals, during et finishes	44
5.3	Matrice de Confusion pour la détection de situations (Caméraman)	47
6.1	Exemple de mesure floue pour un contexte formé de 3 situations.	56
6.2	Liste des rôles détectés dans les vidéos Caviar	58
7.1	Calcul de la qualité des segments	66
7.2	Matrice de confusion et performance pour l'apprentissage de situations à partir des données Caviar	72
8.1	Exemple de matrice attentionnelle	85
8.2	Correspondance entre ressources attentionnelles et niveau de traitement cognitif	86
9.1	Matrice de confusion pour <i>Find-S</i>	100
9.2	Matrice de confusion pour <i>Candidate Elimination</i>	100
9.3	Matrice de confusion pour <i>ID3</i>	100
10.1	Prédicats de description de l'environnement.	112

10.2 Répartition des sujets de l'enquête selon l'âge et le sexe. 119

Contexte général : l'informatique ambiante

La rédaction d'un rapport d'habilitation à diriger les recherches est une opportunité de faire le point et prendre du recul sur les activités de recherche et les différentes pistes explorées au cours des années passées. Mon parcours scientifique a débuté par une thèse en robotique mobile dans le cadre de la navigation réactive. J'ai rejoint en 1995 l'Ecole Nationale d'Ingénieurs de Brest en tant que Maître de Conférences où je me suis ensuite consacré aux environnements de développement multi-agents pour la réalité virtuelle. En 2000, j'ai rejoint l'Université Joseph Fourier où j'ai débuté mes travaux sur l'informatique ambiante au sein de l'équipe PRIMA. Je ne suis pas focalisé sur un problème précis du domaine mais j'ai plutôt considéré une approche transversale "d'intégration". La question en filigrane a été de savoir comment construire un système ambiant en intégrant les connaissances proposées par des disciplines connexes telles que par exemple l'apprentissage automatique.

D'un point de vue développement, un système d'intelligence ambiante est un logiciel complexe, dynamique, devant fonctionner en milieu hétérogène. Je me suis également intéressé aux aspects architectures logicielles, intergiciels et déploiement. Ces aspects de mon travail n'ont donné lieu qu'à deux publications et ne seront pas évoqués dans ce manuscrit.

1.1 1991 : l'informatique ambiante

Le concept d'informatique ubiquitaire ou informatique ambiante[ⓐ] a été développé par Mark Weiser ([Weiser, 1991](#)) au cours des années 80 à Xerox Parc. Il revient sur l'évolution de l'informatique depuis les années 60, caractérisée par trois ères :

ⓐ. Ubiquitous Computing ou ubicomp

1. Les mini-ordinateurs : une unité centrale partagée par un ensemble d'utilisateurs.
2. Les ordinateurs personnels : une unité centrale par utilisateur.
3. La mobilité : plusieurs unités centrales par utilisateur, ces unités centrales pouvant le suivre dans ses déplacements.

La miniaturisation des unités centrales, la réduction de la consommation d'énergie et la généralisation des réseaux (Wifi, 3G, CPL etc) conduit à une omniprésence de dispositifs informatiques nous accompagnant dans notre vie courante : smart phones, console de jeux portables, PDA, ordinateurs portables ... La vision de Mark Weiser est que la multiplication des systèmes va changer radicalement notre façon de les utiliser. Les applications ne seront plus associées à une machine physique et surtout un écran associé. Elles vont pouvoir migrer et nous suivre au gré de nos déplacements. Les interactions vont être plus naturelle et l'ordinateur va "se fondre" dans notre environnement et disparaître. Il doit agir de manière analogue à des lunettes : elles jouent leur rôle sans encombrer notre esprit :

« The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. »

Cette idée est reprise par Norman dans la préface de son livre "*The Invisible Computer*" (Norman, 1999). L'auteur pense que l'ordinateur de son époque est trop intrusif, trop difficile à utiliser, et inadapté à l'être humain. Il le compare même aux premiers phonographes. Edison l'a inventé. Il a une meilleure technologie que ses concurrents. Mais le modèle qu'il produit est "centré technologie" et ne répond pas aux besoins des usagers. Il finit par faire faillite alors que la technologie sous-jacente constitue une des grandes inventions de son siècle.

L'informatique ubiquitaire peut être vue comme l'opposé de la réalité virtuelle. La réalité virtuelle met une personne à l'intérieur d'un monde créé par l'ordinateur. L'informatique ubiquitaire a pour but de permettre à l'ordinateur de "vivre" dans le monde des hommes et de s'y intégrer au point de disparaître.

Une des premières réalisations associée à l'informatique ubiquitaire sont les trois dispositifs *tabs*, *pads* et *boards* développés à Xerox PARC entre 1988 et 1994. Les *tabs* (Adams et al., 1993) sont des PDAs pouvant communiquer par infrarouge avec un réseau d'émetteurs-récepteurs afin d'accéder à des applications s'exécutant sur des stations de travail traditionnelles. Les *pads* peuvent se comparer à des feuilles de papier. Ils sont à disposition partout et peuvent être utilisés par n'importe qui comme des "ordinateurs brouillons". Ils ne sont donc pas associés à une personne et ne la suivent pas comme le ferait un ordinateur portable. Les *boards* (Elrod et al., 1992) sont des tableaux blancs interactifs utilisables dans le cadre de réunions, de présentations ou de collaborations à distance.

On peut également citer les travaux du consortium *Things That Think*^③ du laboratoire Media Lab du MIT. Ce consortium cherche à inventer les objets du futur en embarquant en particulier l'informatique dans les objets de tous les jours. Le projet Sixth Sense^④ par exemple, de l'équipe *Fluid Interfaces Group*, augmente les objets quotidiens en projetant dessus des informations supplémentaires. Cela peut servir par exemple à guider ses achats dans un supermarché.

1.2 1995 : l'informatique calme

Dans la continuité de sa vision initiale, Weiser introduit la notion *d'informatique calme*. L'informatique calme est une approche qui engage à la fois le centre et la périphérie de notre attention, notre attention pouvant se déplacer de l'un à l'autre. Lorsque nous conduisons une voiture, notre centre d'attention est porté sur la route mais pas sur le bruit du moteur qui est en périphérie. Si ce bruit devient inhabituel, il passera instantanément de la périphérie au centre de notre attention. De la même façon, la technologie calme va se déplacer facilement de la périphérie au centre. Placer des choses en périphérie nous permet d'être à l'écoute de plus de choses que ce que nous pourrions supporter si tout était au centre.

Un exemple d'informatique calme nous est fourni par Natalie Jeremijenko. Elle a créé et installé à Xerox le "*Live wire*". Il s'agit d'un fil relié à un moteur pas à pas dont les soubresauts sont directement reliés au trafic réseau. Cela permet donc de visualiser en périphérie et sans effort particulier une information autrement inconnue : l'intensité du trafic réseau.

1.3 1998 : l'intelligence ambiante

La multiplication des dispositifs informatiques s'accompagne également par une multiplication des capteurs interconnectés. Ces capteurs peuvent mesurer des grandeurs physiques (température, humidité, présence, lumière etc.). Cela peut également être des sondes logiciels comme le prochain rendez-vous dans son agenda, l'arrivée d'un nouvel email etc.

L'intelligence ambiante (ou AmI^⑤) est la rencontre de l'informatique ubiquitaire et de l'intelligence artificielle. L'objectif est d'exploiter les capacités de perception offertes

③. <http://ttt.media.mit.edu>

④. <http://www.pranavmistry.com/projects/sixthsense/>

⑤. Ambient Intelligence

par tous ces capteurs afin d'analyser l'environnement, les utilisateurs et leurs activités et de permettre au système de réagir en fonction du contexte. La vision de l'AmI est donc toujours celle de l'ubicom (des ordinateurs qui imprègnent l'environnement tout en étant transparents à l'utilisateur), mais en y ajoutant la notion d'intelligence, c'est-à-dire la faculté d'analyse du contexte et l'adaptation dynamique aux situations. Nous définirons plus précisément la notion de contexte au cours du chapitre 2.

Ce terme a été employé la première fois par Eli Zelkha et Brian Epstein lors d'une présentation en 1998 auprès de Philips. Philips s'intéresse à ces approches en menant une réflexion en interne sur l'évolution de l'électronique grand public (un monde d'appareils fragmentés) vers un univers à l'horizon 2020 de dispositifs conviviaux et ubiquitaires. En 1999, Philips rejoint le consortium *Oxygen*, un consortium international de partenaires industriels dans le contexte du projet Oxygen du MIT[Ⓔ], dont le but est de développer les technologies pour l'ordinateur de demain. En 2000 naît un projet de laboratoire destiné à étudier la faisabilité et l'utilisabilité des approches d'intelligence ambiante dans un cadre domestique. Le *HomeLab*[Ⓒ] est inauguré le 24 Avril 2002.

En parallèle des recherches menées chez Philips, une série de réunions de l'ISTAG (*Information Societies Technology Advisory Group*) ont eu lieu afin d'élaborer la notion d'intelligence ambiante et de donner une vision guidant la direction globale du programme européen de l'IST (*Information Societies Technology*) dédié aux technologies de l'information. Le résultat de ces réunions est un document (Ducatel et al., 2001) publié en 2001 développant quatre scénarios illustrant l'intelligence ambiante pour 2010. Leurs objectifs étaient d'une part d'alimenter sur un long terme la recherche et d'autre part, d'évaluer la position scientifique de l'Europe à propos de ce domaine. L'idée défendue par l'ISTAG est que l'intelligence ambiante devra être attentive aux caractéristiques spécifiques de chacun, s'adapter aux besoins des utilisateurs, être capable de répondre intelligemment à des indications parlées ou gestuelles, et même d'engager un dialogue. Elle devra être non intrusive et le plus souvent invisible. Enfin, elle ne devra pas impliquer une longue période d'apprentissage pour l'utilisateur et devra pouvoir être utilisée par les gens ordinaires.

Citons quelques réalisations d'environnements intelligents, mise en place autour des années 2000. Les domaines d'application sont respectivement le bureau, l'éducation et l'environnement domestique.

La première réalisation rétrospectivement considérée comme de l'intelligence ambiante, est le système Active Badge de Want et al. en 1992 (Want et al., 1992). L'objectif est de rediriger les appels téléphoniques (dans un hôpital par exemple) en fonction de la position des personnes concernées. La localisation est réalisée à partir d'un émetteur infra rouge (l'active badge) attaché à la personne et de récepteurs situés dans les différentes pièces. Ce système de localisation a été amélioré par Ward et al (Ward et al., 1997) afin

Ⓔ. <http://www.oxygen.lcs.mit.edu/>

Ⓒ. <http://www.research.philips.com/technologies/projects/homelab/index.html>

d'envisager des services plus élaborés comme la redirection d'affichage sur l'écran en face de soi, la construction d'un grand écran par la mise côte à côte de petits écrans etc.

L'Intelligent Room du MIT en 1998 (Coen, 1998). Il s'agit d'une salle de conférence ordinaire augmentée de 12 caméras, de deux vidéo-projecteurs, de divers dispositifs d'affichage, de microphones sans fils et de haut-parleurs. L'objectif de cette salle est d'expérimenter différentes formes d'interactions multimodales. La reconnaissance d'activité est basée sur la vision, la reconnaissance de gestes et la reconnaissance de parole. Le choix de la grammaire active utilisée pour la reconnaissance de parole est par exemple basé sur l'activité courante détectée.

Dans le domaine de l'éducation, le projet *eClass* (Abowd et al., 1996) (connu initialement sous le nom Classroom 2000) est une salle de classe augmentée mise en place à l'université Georgia Tech. Son rôle est de capturer les interactions présentes lors d'un cours. La salle possède un système d'enregistrement audio et vidéo, un tableau blanc électronique ainsi que des systèmes de prises de notes individuels. L'objectif de cet environnement est de capturer les échanges d'information qui ont lieu lors d'un cours et de permettre d'y accéder ultérieurement facilement à l'aide d'une interface Web. On peut également citer le projet *Smart Classroom* (Xie et al., 2001; Shi et al., 2003) de l'Université de Tsinghua. Ce projet propose également d'augmenter une salle de classe traditionnelle (vidéo-projecteurs, tableau tactile, microphones et caméras) afin de permettre une intégration naturelle du télé-enseignement au sein d'une classe traditionnelle (voir figure 1.1).

Mozer a conçu un des premiers "environnement domestique intelligent" à l'Université du Colorado : The Adaptive House (Mozer, 1998). L'Adaptive House est une maison équipée de 75 capteurs permettant de fournir des informations sur la température, la lumière ambiante, le niveau sonore, ainsi que les ouvertures de portes et de fenêtres. Elle est également équipée d'effecteurs tels que le contrôle des éclairages, du chauffage central, du chauffage de l'eau, de la ventilation. L'objectif du système est de gérer au mieux l'énergie tout en procurant un environnement confortable. Il se base pour cela sur un modèle de comportement des usagers lui permettant de prédire par exemple l'occupation des différentes pièces. Ce projet est à rapprocher également de MavHome (Cook et al., 2003) de l'université du Texas à Arlington. En plus de la gestion de l'énergie, ce projet s'intéresse au contrôle de systèmes multimédia pour proposer des services aux usagers (diffusion d'informations par exemple). Ces services sont en particulier capables de suivre les personnes dans leur déplacement à l'intérieur de la maison. Le projet EasyLiving de Microsoft (Brumitt et al., 2000) cherche à proposer des services contextuels aux usagers basés en particulier sur leur position. Le système par exemple choisira de projeter la session d'un utilisateur sur un écran qui lui est proche.

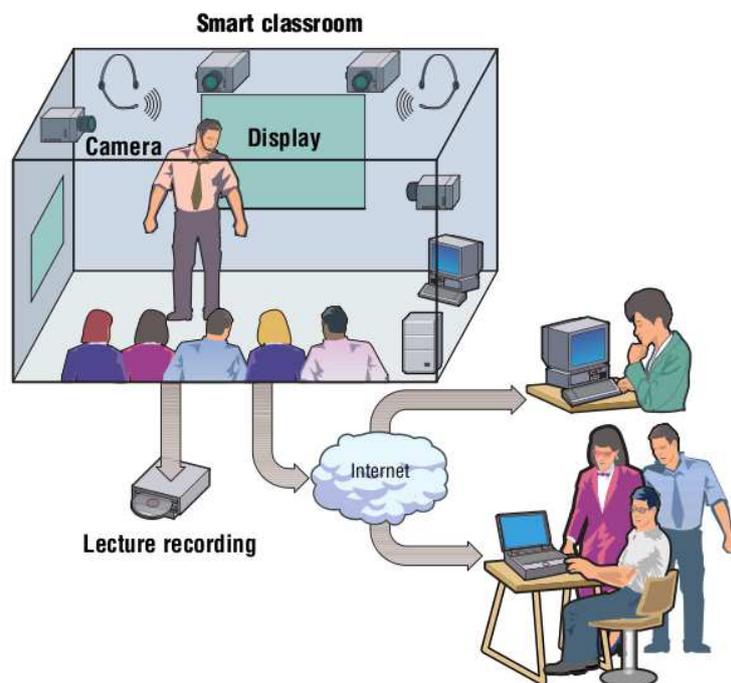


FIGURE 1.1: Smart Classroom de l'Université de Tsinghua (extrait de (Shi et al., 2003))

Ces premiers systèmes ambiants sont essentiellement proactifs. Cela signifie qu'ils sont à l'origine de certaines actions exécutées dans l'environnement. Salovaara et Oulasvirta (Salovaara & Oulasvirta, 2004) soulignent que :

« ... *the concept proactive refers to two critical features of a system : 1) that the system is working on behalf of (or pro) the user, and 2) is taking initiative autonomously, without user's explicit command.* »

Schmidt (Schmidt, 2005) propose d'étendre cette définition à la notion d'*interaction homme machine implicite* (iHCI). iHCI désigne l'interaction d'un usager avec l'environnement afin d'accomplir une tâche. Le système recueille des entrées implicites et propose des sorties implicites. Les entrées implicites sont des actions ou des comportements de l'utilisateur destinées à accomplir sa tâche mais qui ne sont pas considérées comme des interactions avec le système. C'est le système qui les détecte et les interprète comme ses entrées. De la même manière, une sortie implicite n'est pas la réaction à une entrée explicite et s'intègre de manière naturelle à l'environnement et à la tâche en cours.

1.4 Premier bilan

Les années 2000 ont vu l'explosion du nombre de terminaux dans notre vie personnelle et professionnelle. Ce sont par exemple des "postes fixes" comme les traditionnels PC présents dans le cadre professionnel mais également familial (salon, chambre des enfants etc), les consoles de jeu vidéo reliées en réseau, les box des fournisseurs internet etc. Ce sont également des terminaux en mobilité : smart phones, PDAs, lecteurs multimédia, consoles de jeu portables Wifi. La vision de Weiser sur l'informatique ubiquitaire s'est donc en partie réalisée (multiplication des processeurs nous entourant). Le volet "informatique calme" n'est par contre pas encore présent. Nous reviendrons sur ce point lors du chapitre 11.

En revanche, l'intelligence ambiante proactive n'a pas suivi un développement aussi important. Les applications ont du mal à sortir des laboratoires pour se retrouver dans notre vie courante. Quelles en sont les causes ? Quelles sont les difficultés ?

Lueg (Lueg, 2002) pose tout d'abord le problème du choix de la représentation du contexte et de sa mise à jour. Il note en particulier que l'informatique ambiante se heurte au même problème que l'intelligence artificielle "traditionnelle" : le *frame problem*. Quel aspect de l'environnement doit être inclus dans le modèle choisi (les aspects du monde absents du modèle n'ont aucune existence dans le "monde" du modèle) et surtout, comment garder un modèle du monde à jour en fonction des modifications de l'environnement fournies par les capteurs. Ce dernier point pose le problème de l'interprétation des données perçues. L'être humain définit son contexte courant à partir d'une vaste quantité d'informations, ses informations étant interprétées à la lumière de ses propres expériences

passées (Erickson, 2002). Erickson prend comme exemple un théâtre. Nous savons sans difficulté quand est ce que l'on peut parler avec son voisin, quand il faut faire silence, quand il faut applaudir. Ces notions à priori simples sont néanmoins difficiles à modéliser et à intégrer dans un programme (problème en Intelligence Artificielle de l'intégration de la notion de sens commun). Les systèmes sensibles au contexte n'ont généralement accès qu'à un sous-ensemble restreint de données (ce qu'ils sont en mesure de percevoir). Un téléphone capable de couper automatiquement la sonnerie lorsque son propriétaire est au théâtre percevrait au mieux qu'il est immobile dans un endroit sombre avec un niveau sonore ambiant important, ce qui est assez éloigné de l'interprétation humaine de "je suis au théâtre" :

« to map a sparse array of sensor inputs to an actionable interpretation is very different from human awareness »

Lueg a enregistré des séries de réunions afin d'étudier leur déroulement (Lueg, 2002). Il a en particulier constaté que ces réunions incorporaient systématiquement plusieurs phases (comptes rendus, discussions, conception collaborative) mais que les changements entre ces phases ne pouvaient être perçus par les capteurs traditionnellement utilisés et ne peuvent donc être exploités par un agent intelligent. Un agent peut facilement distinguer des réunions informelles, de type pause café, de réunions plus formelles (en se basant par exemple sur un agenda) mais peut très difficilement analyser les différentes phases d'une réunion en cours.

Lueg souligne également que les objets ne sont pas forcément utilisés comme leurs concepteurs l'ont imaginé. Une voiture de sport est par exemple un véhicule destiné à se déplacer (vite). C'est également dans certain cas une marque de statut social. Un agent intelligent ne sera sensible qu'au rôle initialement prévu pour cet objet : le véhicule. Il peut donc vexer le propriétaire en ne le considérant que comme tel. Le MediaCup de Schmidt (Schmidt, 2005) est une tasse de café augmentée d'un dispositif transmettant la position de la tasse au système. Les utilisateurs finissent par ne plus l'utiliser comme une tasse mais comme un badge.

D'une manière plus générale, les décisions d'actions prises par les être humains (et que l'on souhaite donc voir prises par un environnement intelligent) dépendent d'un certain nombre de facteurs tels que l'intérêt courant des personnes, leurs objectifs ((Greenberg, 2001) par exemple), les relations sociales, les phobies, l'humeur etc. Ces paramètres sont actuellement inaccessibles à l'ordinateur et le resteront vraisemblablement toujours. Pour Bellotti et Edwards (Bellotti & Edwards, 2001), ce manque d'informations perçues ne peuvent que conduire l'ordinateur à percevoir les décisions prises par les humains que comme une forme d'improvisation.

Un autre frein à la mise en place d'une application de type intelligence ambiante est la complexité du développement. Ce développement ne peut pas être pris en charge uniquement par le développeur ou l'utilisateur final mais il repose sur ces deux acteurs. En effet, comme le souligne Patty Maes (Maes, 1994), les deux problèmes à résoudre pour

créer un assistant informatique proactif sont celui de la compétence (comment l'assistant acquiert-il la connaissance nécessaire pour pouvoir décider quel service rendre à quel moment) et celui de la confiance (comment faire en sorte que l'utilisateur se sente à l'aise en déléguant des tâches à l'assistant : ce point est développé à la section suivante). Les deux solutions les plus immédiates sont de faire spécifier ces services par le développeur ou bien par l'utilisateur. Pattie Maes montre qu'aucune de ces deux solutions ne satisfait totalement les deux critères de la compétence et de la confiance. En effet, si le développeur prédéfinit tous les services, le problème de confiance n'est pas bien résolu car l'utilisateur aura du mal à maîtriser un système trop complexe et sophistiqué. Si c'est l'utilisateur qui spécifie seul le fonctionnement de l'assistant, c'est le problème de la compétence qui est mal résolu car il sera difficile pour l'utilisateur de fournir des spécifications englobant tous les aspects d'un système complexe.

Nous pensons donc que les deux acteurs doivent coopérer (ce n'est pas la conclusion de Pattie Maes qui préconise l'apprentissage : ce point sera abordé lors de la seconde partie du manuscrit). Le développeur a en charge la mise en place des "briques" de base sur lesquels vont se construire l'application. Ces briques de base sont multi-domaines et couvrent un large spectre de l'informatique comme par exemple :

- la gestion de réseaux de capteurs (zigbee ...)
- le traitement du signal : acoustique, vision par ordinateur
- les interfaces homme-machine
- les intergiciels
- la modélisation de connaissances
- le déploiement
- etc.

Une fois ces briques en place, le développeur va éventuellement spécifier des situations pertinentes et les réactions associées. La notion de situation pertinente est une notion personnelle. De la même manière que dans le cadre d'une application "traditionnelle", le développeur fournit des choix par défaut (taille des fenêtres, couleur, police etc) que l'utilisateur peut ensuite modifier à sa convenance, le développeur doit fournir un ensemble de situations clés à priori pertinent pour l'ensemble des utilisateurs, chaque utilisateur devant pouvoir adapter ces situations à ses propres besoins.

La complexité de cette tâche est grande, aussi bien du point de vue du développeur que de l'utilisateur final. Il est important de proposer une approche adaptée à ces deux catégories d'acteurs.

1.5 Notre approche

Face aux constatations citées lors du paragraphe précédent, j'ai décidé en 2001 de m'intéresser au problème du développement d'applications sensibles au contexte. Il s'agit

de fournir des solutions aux deux acteurs que sont le développeur et l'utilisateur final afin de faciliter leur tâche. Comme l'a souligné Erickson (Erickson, 2002) :

« *we are not designing context-aware systems because we believe this technology is a good thing in and of itself. Rather, the purpose of context-awareness is to allow our systems to take action autonomously.* »

Nous nous plaçons donc dans le cadre d'applications proactives, d'assistants virtuels. Le système n'a pas pour objectif de percevoir le contexte uniquement dans le but de le communiquer à l'utilisateur : il doit être en mesure d'agir par rapport à la situation courante afin de proposer des services appropriés.

Le paragraphe précédent a rapporté les difficultés importantes dûes en particulier aux limites de la perception et aux difficultés de la modélisation. Des recommandations ont été proposées afin de pallier ces problèmes. Il est donc très important que nous intégrions ces recommandations à nos approches.

Comme nous l'avons précédemment indiqué, Lueg montre qu'un agent intelligent peut distinguer une pause café d'une réunion formelle mais ne peut pas facilement déceler les différentes phases de cette réunion formelle. En général, il ne faut pas chercher à mettre en place des services s'appuyant sur une description trop fine des situations mais se limiter à une description de type "gros grain". Il ne faut pas chercher à modéliser "l'ensemble du monde", la totalité des situations possibles mais se restreindre à un sous-ensemble réaliste pour l'application concernée.

La recommandation suivante concerne la confiance que l'utilisateur peut avoir dans un système ambiant proactif. Le problème de la confiance dans le cadre plus large des systèmes automatisés (contrôle de procédé de fabrication, ...) a été étudié en particulier par Muir et Moray (Muir & Moray, 1996). Ils ont montré que la confiance d'un usager dans les capacités d'un système automatique est basée sur la perception que cet usager a de la compétence du système. Cette confiance dans le système peut être dégradée par tout signe d'incompétence, même non important par rapport à la bonne marche du système. L'utilisateur ne doit pas en particulier sous-estimer ou surestimer les capacités du système. L'utilisation du système automatisé est optimale si la confiance de l'utilisateur dans le système est compatible avec les compétences de celui-ci⁽⁷⁾. La relation de confiance homme-machine est similaire à la relation de confiance homme-homme.

Pour établir cette relation de confiance, il est recommandé que le système fournisse un retour à l'utilisateur à propos de son fonctionnement interne. Bellotti et Edwards (Bellotti & Edwards, 2001) ont défini le terme *d'intelligibilité* : le système doit être en mesure de fournir aux usagers ce qu'il sait, comment il le sait et ce qu'il en fait. Cheverst (Cheverst et al., 2005) propose le terme de *compréhensibilité* pour définir que l'utilisateur doit pouvoir regarder à travers le système (boîte de verre) pour percevoir son fonctionnement

⁽⁷⁾. Calibration of Trust

interne. Les utilisateurs craignent en effet que le système fasse *quelque chose dans leur dos* (Abowd & Mynatt, 2000). La compréhensibilité est associée à la *scrutabilité* qui définit la capacité que l'utilisateur a d'interroger ce modèle. Kay (Kay et al., 2003) souligne que cette notion de scrutabilité est contraire à la vision d'ordinateur invisible de Weiser mais reste une nécessité. Parallèlement à la notion de scrutabilité, Cheverst s'intéresse au degré de contrôle sur les actions de l'utilisateur et du système respectivement. Ces deux notions, placées sur deux axes, lui permettent de classifier les approches existantes (voir figure 1.2).

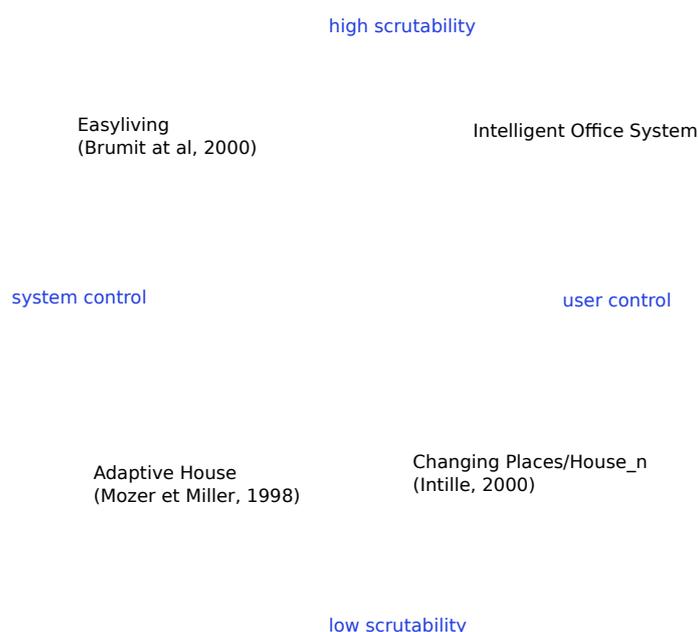


FIGURE 1.2: Classification de systèmes d'Intelligence Ambiante. Le disque correspond à la zone où l'on cherche à se positionner (repris de (Cheverst et al., 2005)).

En résumé, mes recherches concernent les approches permettant de faciliter le développement d'applications proactives sensibles au contexte. Cette aide concerne à la fois le développeur et l'utilisateur final. Afin de permettre l'établissement d'une relation de confiance entre le système et les utilisateurs (et donc permettre l'acceptation du système), celui-ci devra être compréhensible et scrutable : il doit être capable à tout moment d'expliquer ses choix.

Modèle de contexte

Nous avons introduit la notion d'intelligence ambiante au cours du chapitre précédent. Une des composantes de l'intelligence ambiante est une interprétation par le système de sa perception de l'environnement. Elle s'appuie sur la notion de contexte. L'objectif de ce chapitre est de présenter le modèle que nous avons proposé en 2002.

2.1 Quelques définitions

Le mot contexte provient du latin *contextere* qui signifie : "tisser ensemble". Le dictionnaire Petit Larousse (édition 98) le définit comme :

« *Ensemble des circonstances, situation globale où se situe un événement : replacer un fait dans son contexte historique* »

Le terme "sensible au contexte" a été introduit la première fois dans le cadre de l'informatique mobile par Schilit et Theimer ([Schilit & Theimer, 1994](#)). Le contexte est alors défini comme :

« *location information [that] enables software to adapt according to its location of use, the collection of nearby people and objects, as well as the changes to those objects over time* »

Dans le cadre des stick-e documents ("post-it" dont l'affichage du contenu est déclenché par le contexte), Brown ([Brown, 1996](#)) définit le contexte en se basant essentiellement sur la localisation, mais également sur la présence d'objets, des valeurs seuils (température) et l'état interne de l'ordinateur. Le système GUIDE ([Cheverst et al., 2005](#)) ([Davies et al., 1999](#)) permet d'assister les visiteurs de Lancaster grâce à des informations sensibles au contexte. Le modèle de contexte proposé repose sur les deux notions de *localisation* et de *point de navigation* (afin de représenter un parcours).

Ces premières approches abordent essentiellement le contexte sous l'angle de la localisation. De plus, les modèles proposés sont adaptés aux applications envisagées et ne sont pas facilement transposables à d'autres types d'applications. Comme l'avait déjà pressenti Schilit et al en 1994 (Schilit et al., 1994), le contexte ne se limite pas à la localisation

« Context encompasses more than just the user's location, because other things of interest are also mobile and changing. Context includes lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social situation; e.g., whether you are with your manager or with a co-worker. »

Schmidt et al (Schmidt et al., 1998) proposent de classifier les éléments de contexte selon deux catégories :

1. les facteurs humains : les utilisateurs, l'environnement social, les tâches, ...
2. l'environnement physique : la luminosité, le niveau sonore, l'infrastructure, la localisation, ...

Dans le cadre de dispositifs mobiles, Rodden et al. (Rodden et al., 1998) considèrent également différentes catégories : infrastructure, applicatif, système, localisation et paramètres physiques.

Dey, dans son article de 2001 (Dey, 2001), considère que toutes ces définitions ne sont pas *opérationnelles*. Elles ne permettent pas de sélectionner les aspects importants des situations à considérer. Dey propose de se centrer sur les interactions :

« Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves »

Nous souhaitons mettre en place un cadre plus formel permettant au développeur et à l'utilisateur de pouvoir *spécifier* les différentes situations importantes pour l'application. Ce modèle, proposé en 2002 (Crowley et al., 2002) est présenté au paragraphe suivant.

2.2 Notre modèle formel de contexte

L'objectif de ce modèle de contexte est de proposer un cadre pour la *spécification* des contextes clés lors du développement d'une application. Il ne s'agit pas d'une boîte à outil utilisable à l'exécution (comme le Context Toolkit par exemple (Salber et al., 1999)) mais d'un cadre de conception. Les acteurs concernés sont d'une part le développeur et d'autre part l'utilisateur final. Nous nous sommes donc fixé comme contrainte que ce modèle soit générique et suffisamment simple et intuitif pour pouvoir offrir des explications aux usagers.

Le contexte que l'on cherche à représenter est celui de l'environnement composé du système, d'objets et de un ou plusieurs utilisateurs. Ces utilisateurs ont généralement plusieurs objectifs qu'ils poursuivent en parallèle. Ils basculent entre ces buts de manière dynamique et non prévisible. Notons au passage, que dans notre hypothèse, interagir avec le système n'est pas une tâche principale pour l'utilisateur : nous nous plaçons dans le cadre de l'informatique évanescence. A la notion d'objectif est également associée la notion de tâche et d'activité. Le concept associé à la définition de tâche est celui d'état (Nilsson, 1982). Pour satisfaire notre contrainte de simplicité, nous avons choisi une représentation des états sous forme de prédicats.

Si l'on se place du point de vue utilisateur, le monde est alors représenté sous la forme d'un graphe d'états, les liens entre ces états étant les actions provoquant les changements d'état. Il a une perception de l'état courant et sa liste d'objectifs lui fournit la liste des états finaux qu'il souhaite atteindre. Une tâche peut alors se définir comme l'association (état courant, état final). En fonction de sa liste de tâches, l'utilisateur va choisir des actions de manière opportuniste. La liste de tâches actives correspond à l'activité de la personne.

Plaçons-nous maintenant du point de vue Système. Celui-ci doit reconstruire les états à partir de sa perception. Il perçoit ses propres actions (système proactif), les changements d'état, mais pas nécessairement les actions des utilisateurs ayant provoqué ces transformations. Le monde, du point de vue du système, est un graphe dont les liens ne sont plus les actions mais des contraintes temporelles entre états.

Notre modèle de contexte se place du point de vue système. Il est composé d'entités, de rôles, de relations, de situations (les états) et enfin de contextes (les graphes d'états). Nous allons maintenant préciser tous ces termes.

Les entités

Une entité est un ensemble de propriétés. Ces propriétés peuvent être symboliques ou numériques. Elles correspondent à ce que le système est en mesure de percevoir à partir de ses capteurs.

Les rôles

Une situation donnée ne s'intéresse en général qu'à un sous-ensemble particulier d'entités parmi toutes celles présentes. Le rôle est un *filtre* destiné à *sélectionner* des entités. Il s'exprime sous la forme d'un prédicat unaire dont la variable est une entité.

Par exemple :

Pointeur(Objet)
Conférencier(Personne)

Remarque : nous sommes ici au niveau spécification. Il n'y a donc pas d'implémentation à fournir à ce niveau pour les prédicats des rôles.

Les relations

Les relations sont des prédicats n-aires. Elles permettent d'ajouter des contraintes inter-entités. Par exemple, si l'on souhaite décrire que deux entités sont proches l'une de l'autre : *Proche*(E_1, E_2)

Les situations

Une situation est une conjonction de rôles et de relations. Les rôles permettent de préselectionner des entités pertinentes pour la situation. Les relations ajoutent des contraintes supplémentaires à ces entités sélectionnées. Par exemple, on souhaite décrire la situation où le conférencier est proche de la zone de projection (afin de démarrer le powerpoint par exemple). Cette situation peut être décrite de la manière suivante :

$Conférencier(C) \wedge Ecran(E) \wedge Proche(C, E)$

Les actions exécutées par le système sont attachées aux situations.

Le contexte (ou scénario)

Le contexte est un graphe de situations. Les arcs entre les situations représentent des contraintes temporelles qualitatives de type opérateurs de Allen (Allen, 1983; 1984). Ces arcs peuvent connecter des situations ou des regroupements de situations (voir figure 2.1)

2.3 Quelle approche pour la spécification ?

Etant donné une application à réaliser, quelle démarche employer pour spécifier le modèle de contexte correspondant, comment doit-on choisir les situations clés ? Il est

FIGURE 2.1: La situation 3 va apparaître lors de l'échange entre la situation 1 et la situation 2

impossible de décrire l'ensemble des situations possibles que l'application va rencontrer. Comme nous l'avons indiqué lors du chapitre précédent, il ne faut pas vouloir non plus décrire des situations trop fines que l'on ne sera pas capable de percevoir. Il faut au contraire rester le plus général possible.

Nous cherchons à mettre en place des applications proactives. La spécification doit donc être guidée par les actions. Il faut dans un premier temps lister l'ensemble des actions que peut prendre le système. Pour chacune d'entre elles, il faut rechercher la situation qui va amener à son déclenchement. La description de la situation permettra ensuite de spécifier la perception qu'il faudra mettre en oeuvre.

Illustrons cette approche sur un exemple : le système d'acquisition automatique de séminaires. Le système d'enregistrement automatique de séminaires a été développé dans le cadre du projet Européen FAME (FAME, 2001). L'objectif est de réaliser automatiquement un enregistrement d'un séminaire afin d'une part de le diffuser en direct sur internet pour les personnes ne pouvant être présentes, et d'autre part de l'archiver en vue d'une indexation ultérieure. La salle de séminaire est équipée de plusieurs microphones (microphone Lapel sur le locuteur et microphones d'ambiance) ainsi que de plusieurs caméras. La sortie du PC du locuteur est également enregistrée grâce à un boîtier d'acquisition vidéo intercalé entre l'ordinateur et le vidéo-projecteur. Le système d'enregistrement est sensible au contexte : il détermine à chaque instant la situation courante afin de choisir la caméra et le microphone adapté. La figure 2.2 illustre quelques images tournées lors d'un séminaire InTech à l'INRIA. La liste des actions disponibles est :



FIGURE 2.2: Exemple d'images provenant du système de montage

- Démarrer l'enregistrement
- Filmer le locuteur
- Filmer l'assistance
- Filmer les transparents
- Faire un plan large de la salle
- Interrompre l'enregistrement

A partir de ces actions, on détermine les situations correspondantes. Quand doit-on filmer les transparents? Réponse : quand un nouveau transparent est projeté. Appliqué à l'ensemble des actions disponibles, ceci nous donne la liste des situations suivantes :

- Quelqu'un appuie sur le bouton *start* de l'interface.
- Le locuteur parle.
- Une personne de l'assistance pose une question.
- Projection d'un nouveau transparent.
- Quelqu'un entre dans la salle.
- Quelqu'un appuie sur le bouton *stop* de l'interface.

Le modèle de contexte correspondant est présenté figure 2.3

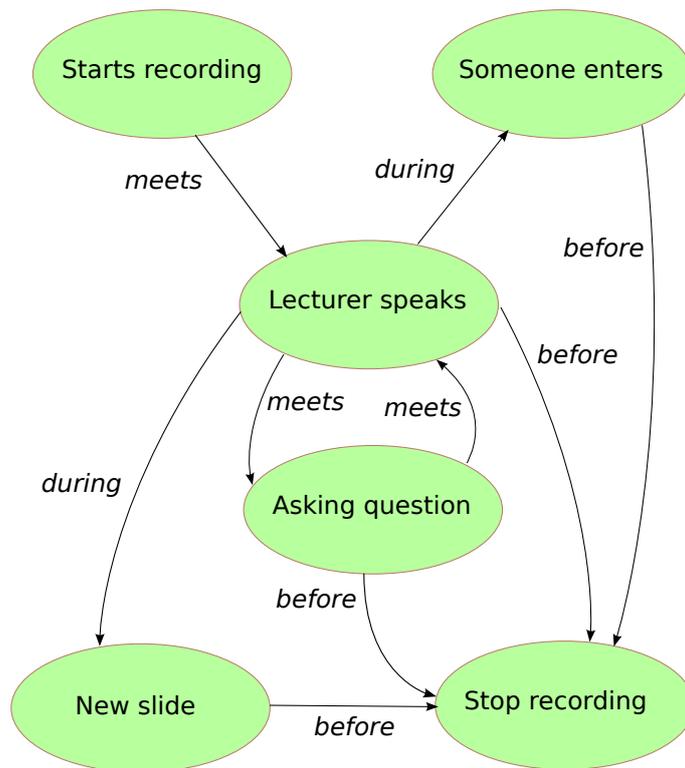


FIGURE 2.3: Modèle de contexte du système d'acquisition de séminaire

2.4 Représentationnel vs Interactionniste

Greenberg ([Greenberg, 2001](#)) dans son article "Context as a Dynamic Construct" remet en cause la vision de Dey selon laquelle le contexte est une construction statique pouvant être décrite en amont par le développeur. Cette vision est partielle et omet la partie dynamique du contexte qui n'apparaît qu'en situation :

« *Although some contextual situations are fairly stable, discernable, and predictable, there are many others that are not. The result is that similar looking contextual situations may actually differ dramatically* »

Il rapproche la définition du contexte de théories antérieures comme l'action située[Ⓐ] ou la théorie de l'activité[Ⓑ].

L'action située proposée par Suchman en 1987 est une critique à une approche de type "plan" de la cognition qui suppose que l'activité est le résultat uniquement de buts et de plans décidés en amont. L'action située envisage les actions d'un individu comme le résultat de l'interaction à chaque instant de cet individu avec son environnement physique et social. L'existence d'un plan n'est pas rejetée : le plan est une ressource comme une autre qui guide l'activité.

La théorie de l'activité reprise[Ⓒ] par Nardi ([Nardi, 1995](#)), ([Nardi, 1996](#)) stipule que l'activité définit le contexte. L'activité est constituée d'un sujet (la personne ou le groupe réalisant cette activité), d'un objet (dans le sens d'objectif : l'intention qui motive cette activité) et des opérations (les actions résultantes de cette activité). L'activité en tant que contexte est donc constituée de ressources externes (l'environnement) mais également internes (les intentions). La présence des buts (objectifs) en amont est ce qui différencie la théorie de l'activité de l'action située.

Cette vision dynamique du contexte est reprise par Paul Dourish en 2004 dans son article de référence "What we talk about when we talk about context" ([Dourish, 2004](#)). Dourish propose de classer les approches existantes en deux catégories : les représentationnelles et les interactionnistes. Le modèle représentationnel est le point de vue classique du programmeur. Il s'agit de déterminer comment un système informatique peut encoder, modéliser, représenter le contexte. De ce point de vue, le contexte est une information sur des aspects de l'environnement dans lequel se déroule une activité. L'autre point de vue est le côté interactionniste. Le contexte est une propriété relationnelle entre des objets et des activités. Le contexte n'est pas une propriété fixe de l'environnement mais émerge de l'activité. Le point de vue interactionniste tente de répondre à la question : comment et pourquoi, au cours de leurs interactions, les personnes atteignent et maintiennent-elles

Ⓐ. Situated Action

Ⓑ. Activity Theory

Ⓒ. Théorie développée en 1920 en Union Soviétique

une compréhension commune du contexte de leurs actions? Ce point de vue est orienté utilisateur.

Paul Dourish oppose ces deux points de vues, argumentant que la vision représentationnelle est une mécompréhension de la nature du contexte. Nous pensons au contraire que ces deux visions ne doivent pas être opposées mais sont complémentaires. Tout d'abord, comme le souligne Nardi (Nardi, 1995) à propos de l'action située (vision interactionniste) :

« *the inquiry [of analysis of situated action] is meant to take place at a very fine-grained level of minutely observed activities, inextricably embedded in a particular situation* »

Ce niveau *fin* de description n'est pas forcément ce que l'on cherche systématiquement à atteindre (cf rester à une description "gros grain" des situations). Mais surtout, comme nous l'avons souligné lors du chapitre précédent, la mise en place d'une application sensible au contexte repose sur deux acteurs : le développeur et l'utilisateur final. Le développeur fournit une vision représentationnelle en construisant un premier modèle de contexte. Ce premier modèle est certainement trop générique mais sert de *connaissance initiale* au système. Ce modèle est ensuite complété, modifié par l'utilisateur final au cours de ses interactions avec l'application. C'est le point de vue interactionniste qui vient en complément du point de vue représentationnel.

Le modèle de contexte que nous avons proposé en 2002 ne se limite pas à une vision de l'état uniquement sous forme de valeurs d'attributs d'entités. Il repose en particulier sur la notion de *relation*. Il est suffisamment général pour pouvoir être utilisé selon un point de vue représentationnel *et* interactionniste.

2.5 De la spécification à l'implémentation

La spécification du contexte est une étape amont qui intervient lors de la conception de l'application. De même qu'un modèle UML permet de spécifier une application sous forme d'objets qu'il faut ensuite implémenter, l'étape suivante est de transformer la spécification de contexte en une application contenant les modules présentés figure 2.4 Comme nous l'avons signalé auparavant, les deux acteurs impliqués dans ce développement sont d'une part le développeur et d'autre part l'utilisateur final. Nous allons au cours des prochains chapitres présenter notre contribution afin de faciliter la tâche de ces deux catégories d'acteurs.

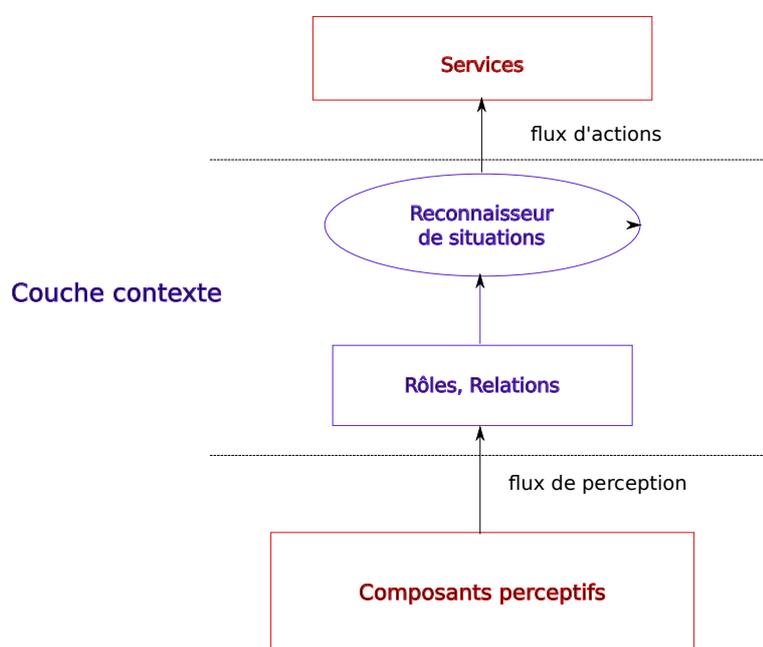


FIGURE 2.4: Architecture générale de nos applications contextuelles

Première partie

Point de vue développeur

Interprétation de la perception

3.1 Rôles et relations

Un rôle a pour objectif de sélectionner parmi l'ensemble des entités perçues, celles susceptibles d'être importantes pour la situation que l'on cherche à reconnaître. Il s'agit donc d'un problème de classification : on dispose d'un nombre fixé de catégories. A chaque cycle de perception, les entités se retrouvent éventuellement affectées en fonction de leurs propriétés à l'une de ces catégories[ⓐ]. Chaque catégorie possède une étiquette fournissant une interprétation de ce qui a été perçu : c'est le rôle. Ces étiquettes ainsi que le nombre de catégories ont été fournis par un utilisateur expert.

L'ensemble des rôles (des catégories) ayant été spécifié, il s'agit donc pour le développeur de mettre en place les algorithmes permettant d'affecter les entités perçues aux différentes catégories présentes. Ces algorithmes peuvent être programmés explicitement, en codant directement la sémantique affectée à chacun des rôles. On peut également utiliser des algorithmes génériques de classification que l'on adapte aux données par une phase d'apprentissage (section 3.2). Cette adaptation est réalisée à partir d'un ensemble de données de référence : des séquences de perception, chaque donnée de perception étant associée par un expert à un des rôles possibles.

Remarque : nous sommes dans un cadre d'apprentissage supervisé. Il existe une seconde catégorie d'adaptation automatique de classifieurs aux données : l'apprentissage non supervisé. En apprentissage non supervisé, le nombre de catégories n'est pas forcément fixé au préalable. Mais surtout, il n'y a pas d'étiquette associée à chacune de ces catégories. L'apprentissage non supervisé travaille directement sur les données brutes de

[ⓐ]. Elles peuvent n'être affectées à aucune catégorie si elles ne sont d'aucun intérêt pour l'application que l'on cherche à mettre en place

perception et construit automatiquement ses propres catégories. La mise en oeuvre est donc beaucoup plus simple (pas de phase fastidieuse d'étiquetage des données). En revanche, il n'est pas du tout certain qu'un utilisateur expert soit en mesure après coup de décrire à quoi correspondent chacune des catégories apprises automatiquement. Nous sommes alors face à des boîtes noires. Nous ne sommes plus dans le cadre de transparence et d'explication à l'utilisateur final que nous nous sommes initialement fixés. Nous avons donc décidé de nous limiter aux approches de type apprentissage supervisé.

3.2 Apprentissage supervisé des rôles

Par opposition à la programmation explicite, l'apprentissage automatique paramètre un algorithme générique de classification à partir d'un échantillon représentatif de données annotées. Nous allons présenter dans un premier temps les principes généraux.

3.2.1 Apprentissage supervisé de classifieurs

Nous nous sommes intéressés dans le cadre de la thèse d'Oliver Brdiczka à une approche générative (classifieurs bayésien) et discriminative (Support Vector Machine), ainsi qu'une approche hybride regroupant les deux précédentes. Nous avons également comparé sur un problème particulier une approche neuronale à une approche de type Support Vector Machine.

3.2.2 Classifieur Bayésien

3.2.2.1 Présentation

Le classifieur Bayésien est une approche générative. Il fournit à partir d'une observation des propriétés T d'une entité, le rôle r correspondant ayant la plus forte probabilité (équation 3.1) :

$$r = \operatorname{argmax}_r P(r|T) \quad (3.1)$$

Cette approche a par exemple été utilisée par Ribeiro (Ribeiro & Victor, 2005) ainsi que par Muehlenbrock (Muehlenbrock et al., 2004).

La phase d'apprentissage a pour objectif de construire à partir des exemples annotés la probabilité d'appartenance $P(r|T)$. Par application de la règle de Bayes, (équation 3.2),



thèse
d'Oliver
Brdiczka

$P(r|T)$ peut s'exprimer en fonction de $P(T|r)$.

$$P(r|T) = \frac{P(T|r)P(r)}{P(T)} \quad (3.2)$$

On suppose que la probabilité à priori $P(r)$ est constante sur l'ensemble des exemples. De même, on ne cherche pas à calculer la valeur exacte de la probabilité mais le rôle maximisant cette probabilité. Le terme $P(T)$ est une constante pour l'opération argmax_r et peut donc également être supprimé. On obtient donc (équation 3.3)

$$r = \operatorname{argmax}_r(P(r|T)) = \operatorname{argmax}_r P(T|r) \quad (3.3)$$

Pour chaque rôle r , $P(T|r)$ est modélisé sous forme d'une distribution de type mixture de Gaussiennes. Chacune de ces distributions est construite à partir des exemples annotés grâce à l'algorithme EM (Bilmes, 1998). Le nombre initial de Gaussiennes est initialement fixé à une valeur importante (128). Le nombre de Gaussiennes est un des paramètres à fixer. De manière à automatiser le processus, nous avons choisi de le fixer à une valeur arbitrairement élevée (128) et d'éliminer au fur et à mesure les Gaussiennes n'ayant qu'une faible participation à la distribution finale.

3.2.2.2 Evaluation

L'environnement d'expérimentation est une salle du laboratoire équipée d'un ensemble de caméras et de microphones. Elle est divisée en deux zones : une zone "salon" comprenant un canapé, des fauteuils bas et une table basse, et une zone "travail" comprenant une table de réunion et un vidéo projecteur. Nous avons uniquement utilisé pour cette expérience une caméra grand angle permettant de voir la partie salon dans sa totalité. Un tracker 2D (Caporossi et al. (Caporossi et al., 2004)) permet de suivre les différentes cibles. A chaque image, le tracker fournit pour chaque cible la position (x, y) de l'ellipse englobante, ainsi que ses dimensions (h, w) et son orientation θ (voir figure 3.1).

Les rôles que l'on cherche à détecter sont : marcher, debout, assis, interagir avec la table et allongé (voir figure 3.2). 8 séquences ont été réalisées afin de servir de base d'exemples. Chaque séquence contient plusieurs rôles. La table 3.1 résume pour chaque séquence le nombre d'images et le nombre de rôles. Ces séquences ont été étiquetées à la main, image par image. La table 3.2 résume la distribution des rôles à l'intérieur de chaque séquence.

L'apprentissage du classifieur Bayésien est réalisé en utilisant un 8 fold cross-validation (7 séquences d'apprentissage et une séquence de validation). La matrice de confusion est donnée table 3.3. Cette matrice de confusion met en évidence un taux de reconnaissance

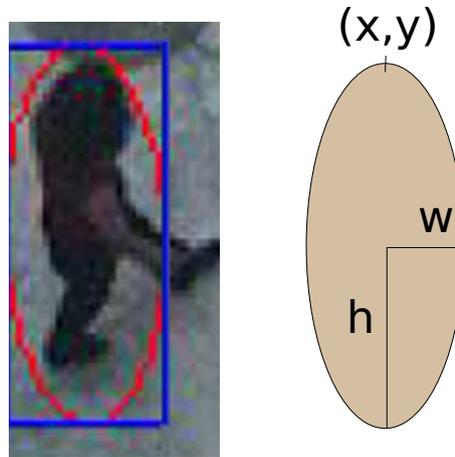


FIGURE 3.1: Ellipse de détection des cibles en mouvement



FIGURE 3.2: Rôles détectés

Séquence	Nombre d'images	Nombre de rôles
1	1352	4
2	6186	5
3	4446	5
4	4684	5
5	4027	5
6	4477	5
7	3067	5
8	3147	5
Total	30885	

TABLE 3.1: Nombre d'images et de rôles par séquence

Séquence	Marcher (%)	Debout (%)	Assis (%)	Interagir (%)	Allongé (%)
1	79	3	11	0	8
2	14	6	48	24	8
3	14	9	40	24	12
4	14	8	50	18	10
5	17	9	46	18	11
6	19	12	42	16	10
7	13	12	50	15	11
8	15	9	48	19	9
Total	18	9	45	19	10

TABLE 3.2: Distribution à l'intérieur d'une séquence

	Marcher	Debout	Assis	Interagir	Allongé
Marcher	0.9335	0.0499	0.0154	0.0008	0.0004
Debout	0.6759	0.2738	0.0393	0.0106	0.0004
Assis	0.0882	0.0169	0.7757	0.1046	0.0145
Interagir	0.014	0.0248	0.154	0.8073	0
Allongé	0.0142	0.0051	0.0202	0.0053	0.9553

TABLE 3.3: Matrice de confusion de l'apprentissage Bayésien

très satisfaisant pour des activités telles que *Marcher* et *Allonger*, un taux inférieur pour *Assis* et *Interagir* et une reconnaissance très faible pour *Debout*. Ces résultats s'expliquent en particulier par la difficulté de distinguer des activités telles que *Debout* et *Marcher* ou *Assis* et *Interagir* avec la table en ne se basant que sur les données provenant d'une seule image.

3.2.3 Support Vector Machine



thèse
d'Oliver
Brdiczka

3.2.3.1 Présentation

Par opposition aux approches génératives, nous nous sommes également intéressé aux approches discriminatives et en particulier à l'une des plus performantes : les Support Vector Machine[ⓑ] (voir par exemple Cortes ([Cortes & Vapnik, 1995](#)), Boser ([Boser et al., 1992](#)) ou Hsu ([Hsu et al., n.d.](#)) pour une mise en oeuvre pratique).

SVM est un algorithme de classification à deux classes. Etant donné un ensemble de données d'apprentissage $\{(x_1, y_1), \dots, (x_n, y_n)\}$ où $x_i \in \mathbb{R}^n$ et $y_i \in \{-1, 1\}$ (2 classes), SVM recherche l'hyperplan permettant de réaliser la plus grande séparation entre les deux classes (voir figure 3.3). Cette séparation n'est pas réalisée dans l'espace initial des x_i mais dans un espace de dimension supérieur $\phi(x_i)$. L'hyperplan est construit à partir de l'équation 3.4.

$$\min_{\omega, b, \xi} \frac{1}{2} \omega^T \omega + C \sum_{i=1}^n \xi_i \quad \text{sachant que } y_i(\omega^T \phi(x_i) + b) \geq 1 - \xi_i \text{ et } \xi_i \geq 0 \quad (3.4)$$

$K(x_i, x_j) = \phi^T(x_i)\phi(x_j)$ est appelée la fonction noyau. La détermination de nouveaux noyaux est un sujet de recherche. Notre objectif n'est pas de proposer de nouveaux noyaux mais plutôt de valider si les noyaux "classiques" existants peuvent convenir pour nos applications. Il existe 4 noyaux classiques :

- Linéaires : $K(x_i, x_j) = x_i^T x_j$
- Polynomiaux : $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- Radial Basis Function (RBF) : $K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|)}, \gamma > 0$
- Sigmoides : $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

où γ , r et d sont les paramètres du noyau.

Remarque : SVM réalise une classification entre deux classes. Cette limitation peut être contournée dans le cadre de N classes en construisant $\frac{N(N-1)}{2}$ SVMs.

ⓑ. SVM

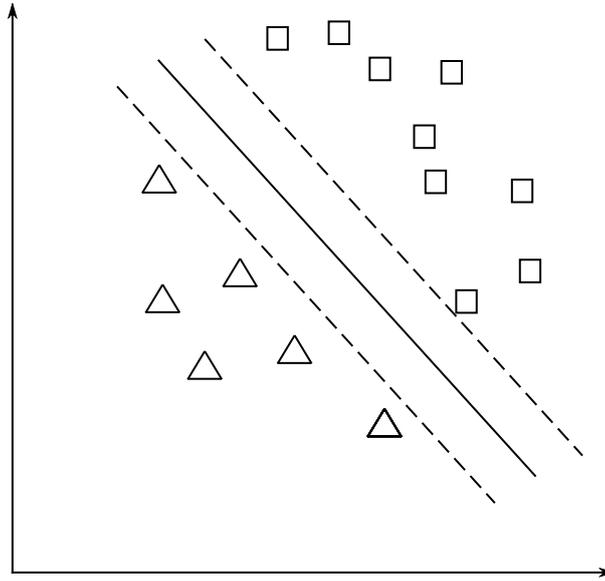


FIGURE 3.3: Classification par un hyperplan (et sa marge) de deux classes

3.2.3.2 Evaluation

L'évaluation a été réalisée dans le même cadre que le classifieur Bayésien :

- Utilisation des propriétés (X, Y, w, h, θ) des entités fournies par le tracker.
- Données d'apprentissage fournies par l'annotation image par image des 8 films
- 8 cross-fold validation.

Le noyau retenu est un noyau de type RBF. Les paramètres sont $C = 11.0$ et $\gamma = 11.0$. Ces valeurs ont été choisies après plusieurs essais. La matrice de confusion est fournie table 3.4.

	Marcher	Debout	Assis	Interagir	Allongé
Marcher	0.8198	0.0824	0.0732	0.0206	0.004
Debout	0.6035	0.2884	0.0793	0.0288	0
Assis	0.0293	0.0005	0.8471	0.0593	0.0638
Interagir	0.0014	0.0014	0.1661	0.8275	0.0036
Allongé	0	0	0.1422	0	0.8578

TABLE 3.4: Matrice de confusion pour le SVM

Les résultats sont plus homogènes que ceux obtenus par le classifieur bayésien. On retrouve la difficulté de reconnaissance de l'activité *Debout* mais SVM obtient de bien meilleurs résultats pour la reconnaissance de *Assis* et *Interagir*.

SVM est une approche discriminative. Son but est d'optimiser la classification des exemples fournis. Elle ne cherche pas à reconstruire la structure des données mais à trouver les frontières entre les différentes classes. Ce principe le rend peu apte à rejeter des données non préalablement rencontrées provenant par exemple de nouveaux rôles. La prise en compte de cette limitation est abordée au cours de la section suivante.

3.2.4 Approche hybride pour la prise en charge de nouvelles données



thèse
d'Oliver
Brdiczka

Les deux approches précédentes ont montré la faisabilité d'apprendre automatiquement la fonction d'interprétation des données perçues en rôles. La nature supervisée de l'apprentissage nécessite l'acquisition de données expérimentales pour un traitement hors ligne. Comme dans toute approche de ce type, il faut se poser la question de la couverture de ces données par rapport aux cas futurs rencontrés lors de l'utilisation en ligne de l'application.

Comme nous l'avons déjà indiqué précédemment, il n'est pas possible de prévoir l'ensemble des situations possibles dans le cadre d'applications sensibles au contexte. En particulier, le comportement des utilisateurs est susceptible d'évoluer au cours du temps, donnant naissance à de nouveaux rôles qui n'ont pas pu être initialement prévus, et donc répertoriés dans les données d'apprentissage.

Nous allons dans un premier temps vérifier comment ces nouvelles données sont prises en charge par SVM et par le classifieur bayésien. Nous verrons ensuite comment ces nouvelles données peuvent être détectées afin de rendre plus robuste ces deux classifieurs (approche hybride).

SVM et Bayésien

SVM, de par son principe, va systématiquement proposer une classe, quelle que soit la donnée fournie en entrée. En particulier, il n'est pas en mesure d'indiquer si la donnée qu'on lui soumet est cohérente avec les données utilisées lors de l'apprentissage. Lors de l'apparition de nouvelles classes, SVM cherchera à affecter ces données aux classes connues lors de l'apprentissage. La mise en place d'une classe "autre" ou "poubelle" pour résoudre ce problème n'est pas facilement envisageable, n'ayant par définition pas de représentant de ces classes lors de l'apprentissage.

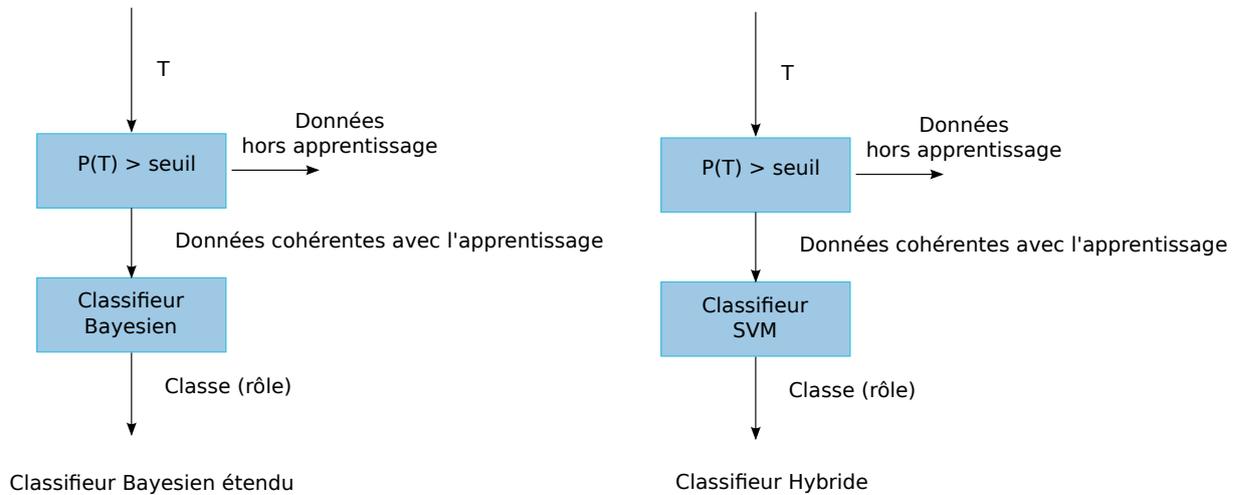


FIGURE 3.4: Classifieur Hybride

Le classifieur bayésien, de par sa nature générative, cherche à construire un modèle des données d'apprentissage. Il fournit la probabilité d'appartenance d'une donnée aux différentes classes. Ses capacités de généralisation sont moindres mais il est en mesure de détecter la présence de nouvelles classes.

Approche hybride

De la même façon que nous avons construit $P(T|r)$ à l'aide d'une mixture multidimensionnelle de Gaussienne estimée par EM (section 3.2.2), nous pouvons également construire selon le même principe $P(T)$. Cette distribution modélise les données d'apprentissage et fournit donc un indicateur de cohérence d'une nouvelle donnée perçue par rapport aux données initiales. Cet indicateur peut donc être utilisé pour détecter les nouvelles classes et servir de filtre aux deux approches de classification précédentes (voir figure 3.4).

3.2.4.1 Evaluation

L'évaluation de l'approche hybride a été réalisée en reprenant les 8 séquences d'apprentissage et en supprimant à tour de rôle une des classes lors de l'apprentissage. Nous avons comparé les résultats obtenus par le classifieur bayésien étendu, le classifieur Hybride et le SVM simple afin de servir de référence pour la comparaison. Les résultats sont résumés table 3.5.

	Classifieur bayésien étendu	Classifieur Hybride	SVM
Moyenne	75.2157	77.8555	71.0088
Ecart Type	5.4840	6.3881	8.3958

TABLE 3.5: Taux de reconnaissance des trois classifieurs

Le filtrage des données avant classification permet au classifieur bayésien étendu ainsi qu'au classifieur hybride d'obtenir de meilleures performances par rapport à un SVM simple. Le classifieur le plus performant des trois est le classifieur hybride. Il permet de tirer parti à la fois de la modélisation des données apporté par le bayésien et du pouvoir discriminatif de SVM.

Reconnaissance de scénarios

Nous avons présenté au cours du chapitre 2 notre modèle de contexte. Ce modèle formel est à la disposition du concepteur de l'application afin de *spécifier* l'ensemble des situations (et leur enchaînement) pertinentes pour l'application.

Il n'existe pas de formalisme générique de reconnaissance robuste de situations à partir des données de perception. Il existe diverses approches qui restent néanmoins très dépendantes des applications. Notre point de vue est qu'au lieu de rechercher un méta-modèle de "reconnaisseur universel", il est préférable de chercher à automatiser la transformation du méta-modèle formel de spécification du contexte vers différents méta-modèles de reconnaissance. Cela permet de ne travailler qu'au niveau spécification et de pouvoir facilement basculer entre les modèles au sein d'une même application.

4.1 Transformation de modèles

L'approche que j'ai mise en place en 2003 s'apparente à de l'Ingénierie Dirigée par les Modèles[Ⓐ]. Dans le cadre de l'informatique ambiante, ce type d'approche a été utilisé plus récemment pour faciliter le développement des applications. Dans le cadre par exemple de l'informatique ambiante domestique, les périphériques utilisés et les normes d'interconnexion évoluent très rapidement (KNX/EIB, X10 ...). Clemente et al. ont proposé un langage spécifique de description[Ⓑ] de l'architecture de l'application et les transformations de modèles associées permettant de générer le code vers une technologie cible (Clemente et al., 2009). Georgalas et al. (Georgalas et al., 2007) génèrent égale-

Ⓐ. IDM

Ⓑ. DSL

ment à partir d'un même modèle le code des interfaces Homme-Machine vers plusieurs technologies cibles : des téléphones portables `j2me` ou des PDA sous `.net`, par exemple.

Les deux exemples précédents adressent le problème de la prise en compte de la multiplicité des matériels et logiciels impliqués dans une application d'intelligence ambiante. Nous nous plaçons plutôt sur un problème interne de choix de formalisme de reconnaissance. Cela se rapproche de ce que Pires et al. ont proposé en 2008 (Pires et al., 2008). Leur approche se base sur la mise en oeuvre d'un langage spécifique de type Event-Control-Action permettant de spécifier le comportement de l'application. Une transformation de modèle permet ensuite de traduire cette spécification en un ensemble de règles destinées à Jess^⑦.

Nous allons présenter dans la suite de ce chapitre une première transformation de notre modèle de contexte vers un réseau de Petri équivalent. Les deux chapitres suivants seront consacrés à la transformation de ce réseau de Petri vers deux méta-modèles pour la reconnaissance de contexte.

4.2 Réseaux de Petri

Le modèle de contexte tel que nous l'avons défini est un graphe de situations. Les arcs du graphe sont étiquetés par les relations temporelles de la logique de Allen. Afin de faciliter les transformations ultérieures, nous allons dans un premier temps homogénéiser les arcs en transformant le modèle en réseau de Petri équivalent.

Le réseau de Petri (ou RdP) est un outil graphique de modélisation de systèmes dynamiques discrets (Petri, 1962). C'est un graphe orienté composé d'arcs et de deux catégories de noeuds : les places et les transitions. Les places représentent les variables d'état du système et contiennent un nombre positif ou nul de marques. Les transitions connectent les places entre elles et représentent l'évolution du système. Une transition est valide si les places en amont possèdent au moins une marque. Lorsqu'une transition est "tirée", une marque est supprimée de chaque place amont et une marque est ajoutée à chaque place en aval. Une définition plus formelle est disponible dans (Murata, 1989)

Cette première transformation du modèle de contexte vers un réseau de Petri est de type $m2m$ ^⑧. Elle est manuelle et n'a pas été automatisée. Chaque opérateur temporel de Allen est remplacé par un motif de réseau de Petri équivalent. Les situations sont représentées par les places. La structure du graphe (via les transitions) modélise les contraintes temporelles d'enchaînement des situations. L'ensemble de ces transformations est résumé

⑦. Jess, the Rule Engine for the Java Platform

⑧. model to model

dans les tables 4.1 et 4.2. Remarque : les situations étiquetées *Unspecified* ne sont associées à aucun rôle ou relation et sont donc activables quoi qu'il arrive dans l'environnement.

La spécification de contexte est maintenant sous la forme d'un réseau de Petri. La prochaine étape va consister à sélectionner un méta-modèle *pour la reconnaissance*. Il s'agit de transformer la spécification en un programme chargé de valider que les événements de perception (et donc l'enchaînement de situations) se déroulent dans un ordre compatible avec la spécification. Nous avons retenu deux méta-modèles : les réseaux de Petri synchronisés et les réseaux de Petri flous. Nous allons les présenter au cours des deux prochains chapitres.

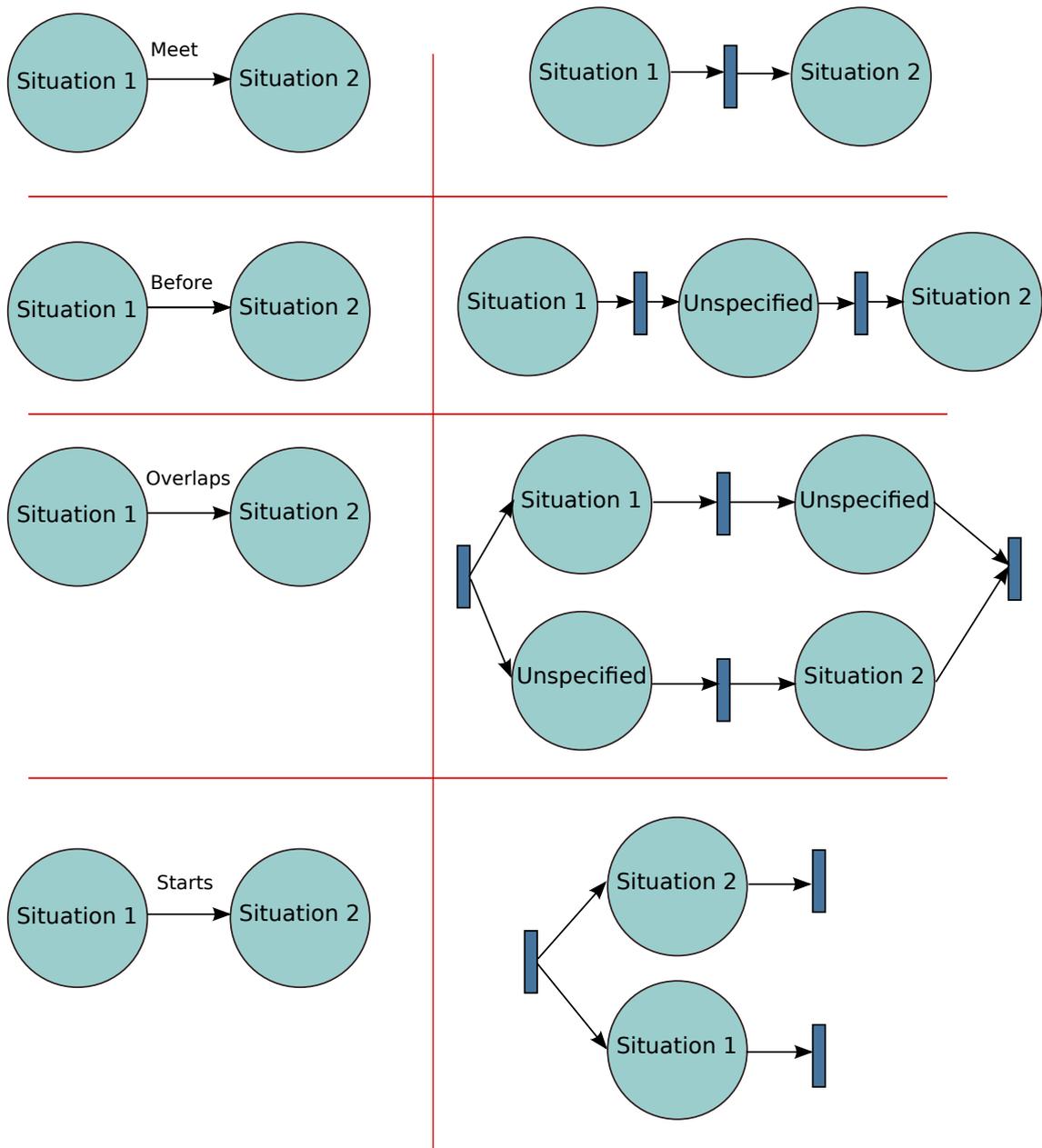
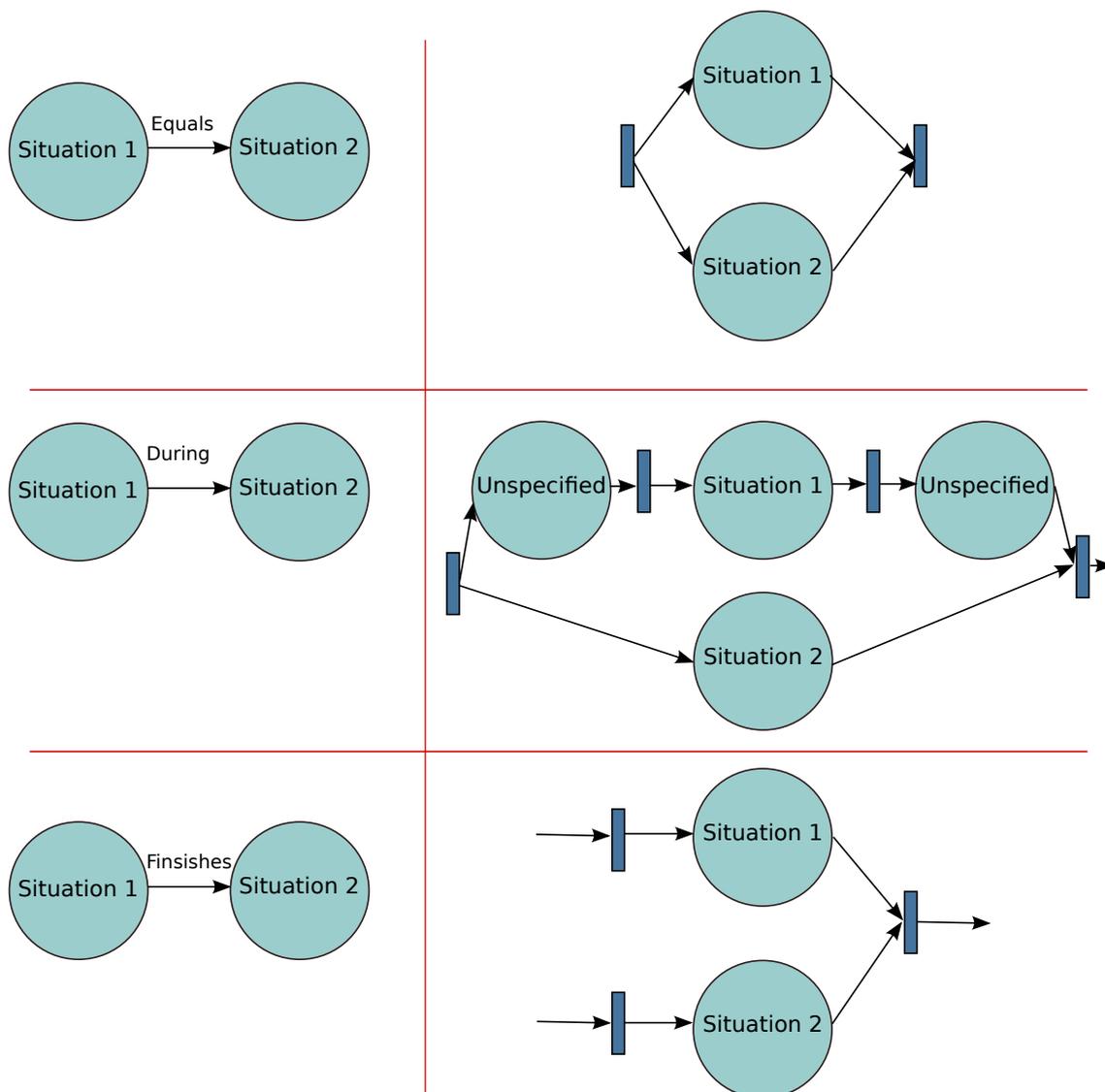


TABLE 4.1: Transformation Allen → Petri : meet, before, overlaps et starts

TABLE 4.2: Transformation Allen \rightarrow Petri : equals, during et finishes

Méta modèle : RdP synchronisé

Le réseau de Petri décrit dans la section précédente est un langage permettant au développeur de décrire formellement les enchaînements de situations pertinents pour l'application qu'il est en train de concevoir. L'étape suivante va consister à transformer ce modèle de description en un modèle de reconnaissance. Rappel :

- *Le modèle de description* fournit par une analyse statique (hors ligne) l'ensemble des séquences possibles de situations (à travers le graphe d'accessibilité)
- *Le modèle de reconnaissance* est un modèle en ligne, relié à la perception. Son rôle est de vérifier que l'interprétation des données de perception sous forme d'entités, rôles et relations conduisent à un enchaînement de situations conforme à ceux spécifiés par le modèle de description.

L'évolution du réseau de Petri dans le cadre du modèle de reconnaissance doit donc être contrôlée par la perception. Nous allons pour cela transformer le réseau de Petri classique issu de la modélisation en un réseau de Petri synchronisé. Un réseau de Petri synchronisé associe un événement à une transition. La transition est franchissable si classiquement l'ensemble des places en amont possèdent au moins une marque et si l'événement associé a été émis.

5.1 Calcul de la synchronisation

Considérons par exemple deux situations $S1$ et $S2$ telles que $S1$ *meets* $S2$. La transition entre les deux situations doit être activée si la perception permet de conclure que $S1$ n'est plus valide **et** que $S2$ est valide :

$$T = \neg valid(S1) \wedge valid(S2)$$

De manière plus générale, l'événement associé à la transition est la conjonction d'un prédicat associé aux places en amont de cette transition et d'un prédicat pour les places en aval :

$$T = PreT \wedge PostT$$

Le prédicat $PreT$ correspond à ce qui est actuellement observé. $PostT$ correspond à ce que le système s'attend à voir étant donné le contexte courant. La liste des prédicats Pre et $Post$ sont fournis pour l'ensemble des opérateurs de Allen dans les tables 5.1 et 5.2.

La dernière étape consiste à transformer le réseau synchronisé en un programme exécutable. Ce programme a pour objectif de réagir aux événements produits par la perception afin de calculer la nouvelle situation et l'action éventuelle associée. Ce paradigme peut naturellement s'implémenter dans le cadre d'un système à base de règles de production (type chaînage avant). Les événements produits par la perception sont injectés sous forme de faits. Le réseau de Petri synchronisé est compilé en règles :

- Un premier groupe de règles pour les règles de transition au sein d'un RdP.
- Un second groupe pour le calcul des événements de synchronisation.

Ces deux groupes de règles calculent la situation courante, également injectée sous forme de faits. Un troisième groupe de règles propose finalement l'action associée à la situation courante.

5.2 Application : acquisition automatique de séminaire

Considérons le système d'acquisition automatique présenté lors de la section 2.3. Pour rappel, la liste des actions disponibles est :

- Démarrer l'enregistrement
- Filmer le locuteur
- Filmer l'assistance
- Filmer les transparents
- Faire un plan large de la salle
- Interrompre l'enregistrement

Les situations correspondantes sont :

- Appuie sur le bouton *start* de l'interface
- Le locuteur parle
- Une personne de l'assistance pose une question
- Projection d'un nouveau transparent
- Quelqu'un entre dans la salle
- Appuie sur le bouton *stop* de l'interface

Les rôles et relations associés sont calculés à partir des événements de perception fournis par un tracker vidéo (entrée d'une nouvelle personne dans la pièce), d'un calcul d'énergie

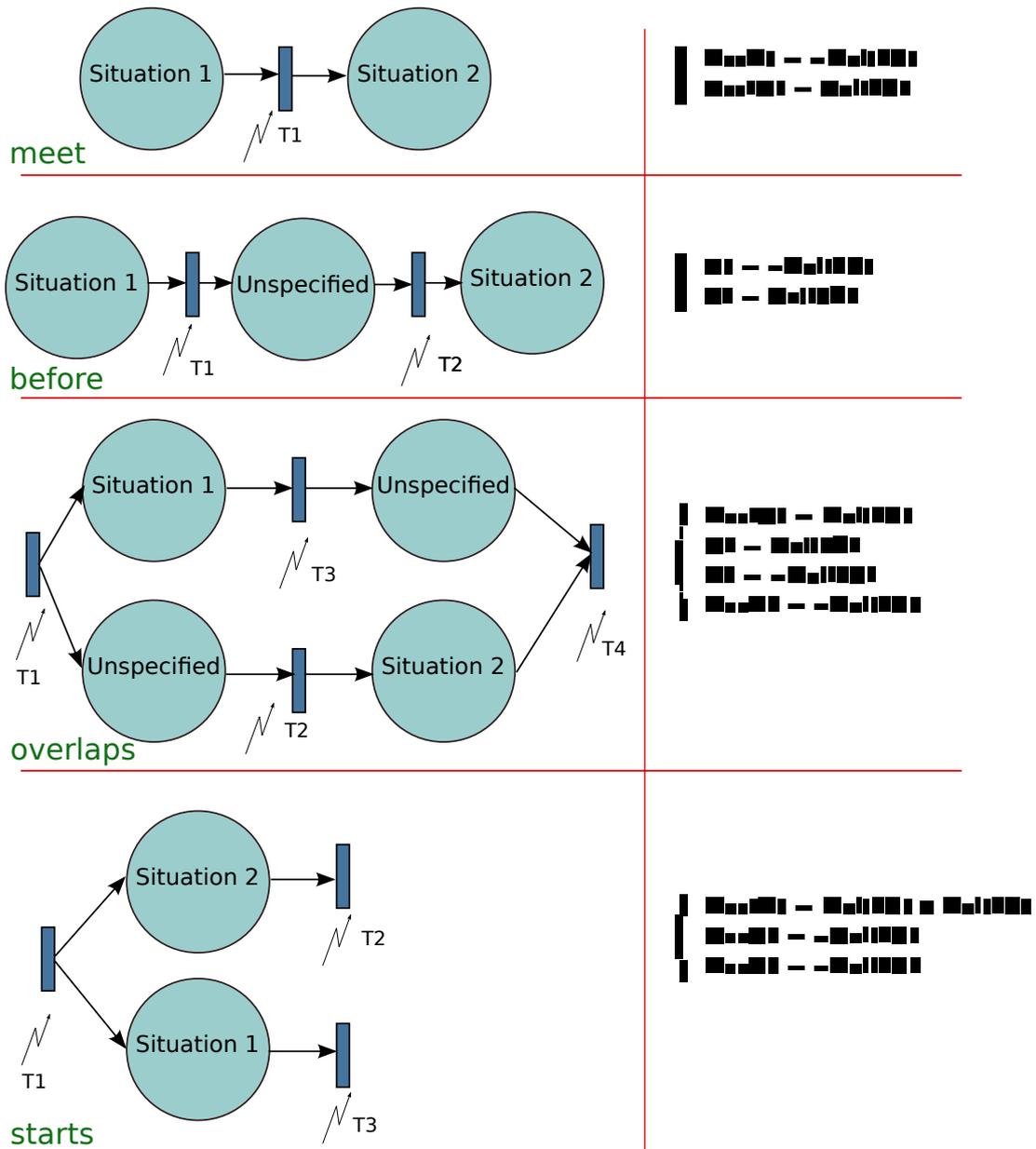


TABLE 5.1: Evénements associés aux opérateurs meet, before, overlaps et starts

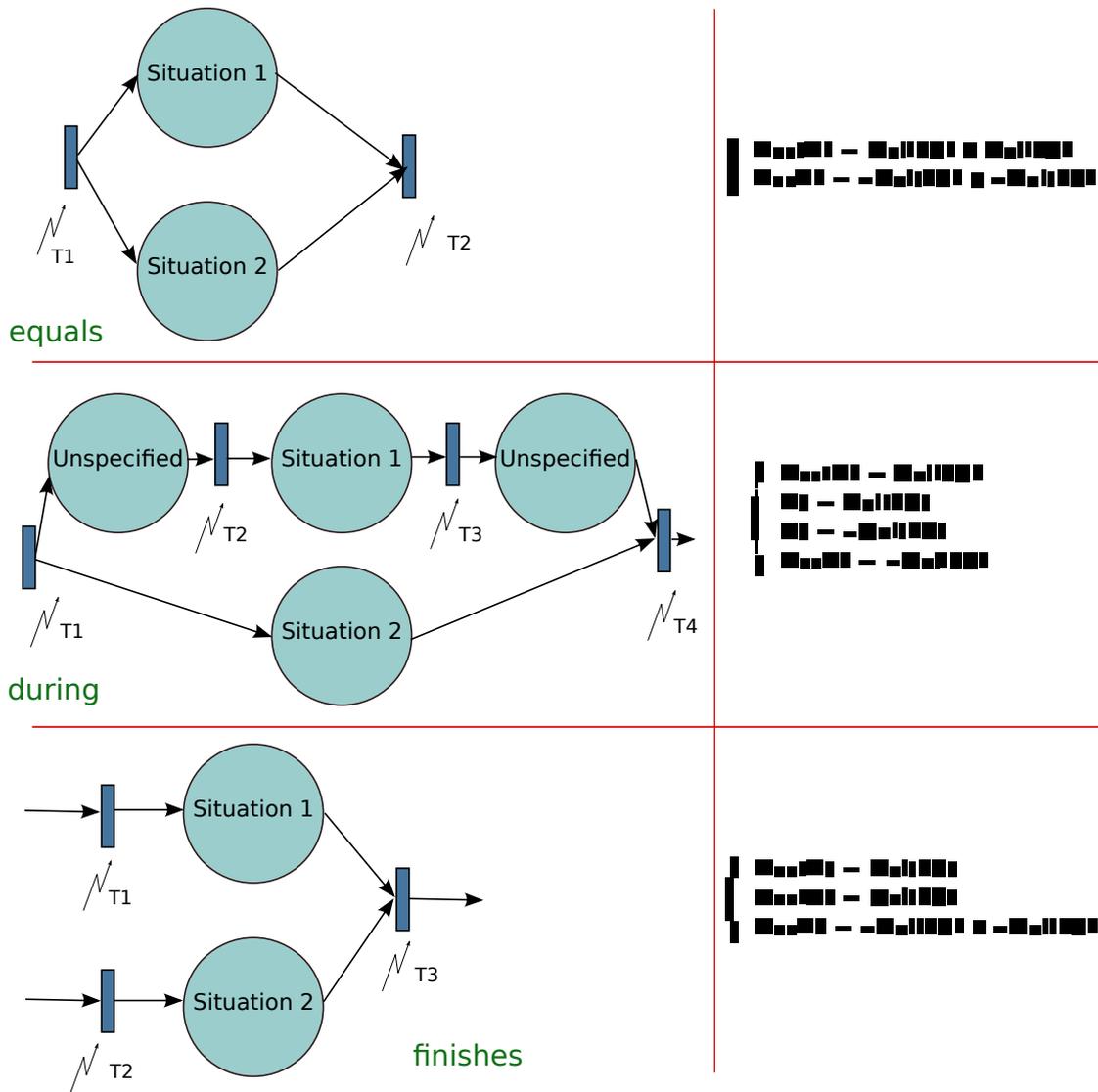


TABLE 5.2: Evénements associés aux opérateurs : equals, during et finishes

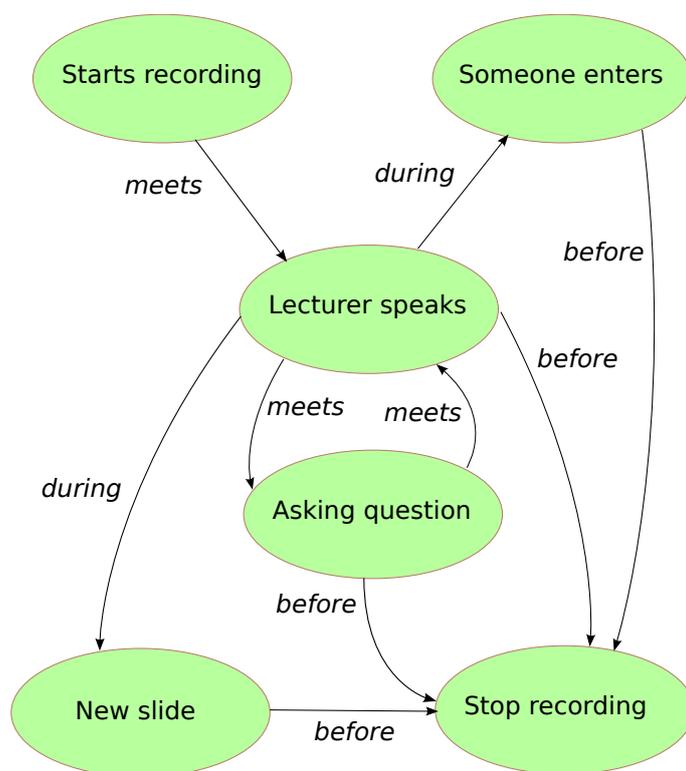


FIGURE 5.1: Modèle de contexte du système d'acquisition

sur une différence d'images (détection d'un nouveau transparent), d'un classifieur de son (détection d'activité vocale pour le locuteur ou pour l'audience) et des événements "classiques" d'interfaces Homme-Machine (début et fin d'enregistrement).

Le modèle de contexte est fourni Figure 5.1. Il est ensuite transformé en réseau de Petri équivalent (Figure 5.2), lui-même transformé en réseau de Petri synchronisé puis compilé sous forme de règles Jess. Un exemple de règle est fourni Figure 5.3.

Ce système a été utilisé pour filmer l'intégralité du workshop *Tecnologies de la llengua : darrers avenços* qui s'est déroulé sur une semaine à Barcelone en Juillet 2004. Il a également permis de diffuser deux séminaires InTech de l'INRIA. Il est en cours d'installation de manière permanente[Ⓐ] dans l'amphithéâtre de la maison Jean Kutzmann. Une version modifiée est également en cours d'installation dans une salle de réunion (avec un modèle de contexte adapté pour filmer une réunion).

Ⓐ. dans le cadre de la plateforme du LIG

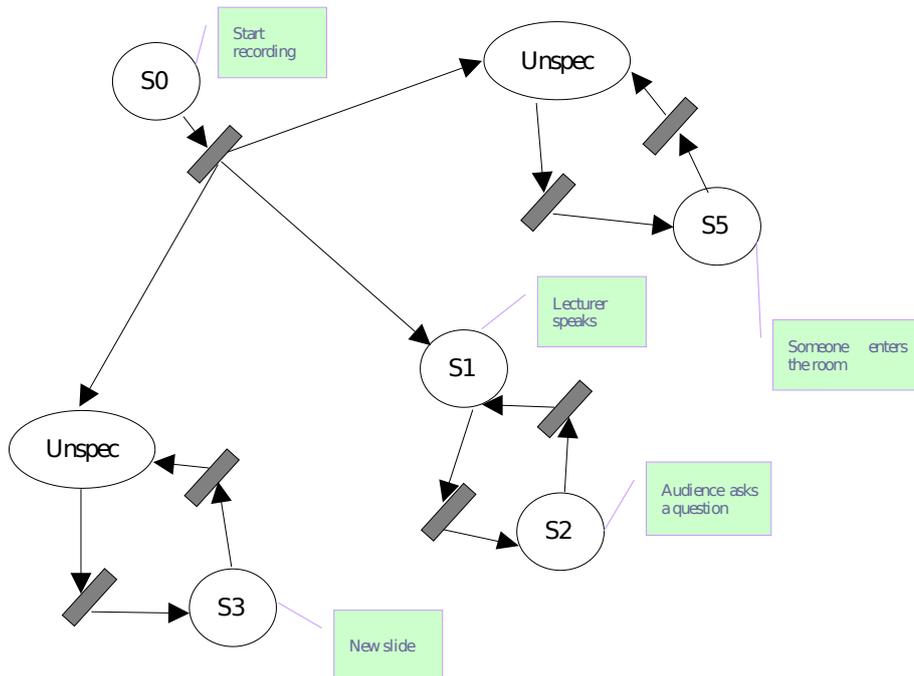


FIGURE 5.2: Réseau de Petri du système d’acquisition de séminaire

5.2.1 Evaluation expérimentale

La table 5.3 représente la matrice de confusion associée à la reconnaissance de situations au cours de deux heures d’enregistrement. Les données de référence ont été produites manuellement en annotant directement les situations. Les trois situations considérées sont :

- *New Slide* : un nouveau transparent est projeté à l’écran.
- *Lecturer Speaks* : le locuteur fait sa présentation ou répond à une question.
- *Audience Speaks* : une personne de l’audience pose une question.

Les lignes de la matrice contiennent le résultat de la détection. Les colonnes contiennent les réponses attendues.

Nous obtenons un taux de reconnaissance de 93.7%. Néanmoins, la détermination du contexte étant déterministe, la matrice de confusion montre essentiellement la robustesse des algorithmes de perceptions utilisés (détection d’activité vocale et détection d’un nouveau transparent).

Nous avons également réalisé une étude informelle de satisfaction des utilisateurs lors de la diffusion sur internet d’un séminaire InTech de l’INRIA. Les spectateurs distants pouvaient accéder à deux flux : un flux provenant d’une caméra fixe et un flux provenant

```

;;
;; Rules for firing transition t2
;;
(defrule t2EventTransition
  ?tr <- (transitionTrigger (name "t2") (lock ?l&:(neq ?l 0)))
  ?pre_S1 <- (situation (name "S1") (entities ?lecturer ?speaker))
  (situationPlace (name "S1") (mark ?m_S1&:(neq ?m_S1 0)))
  (or
    (not (lecturer (isPlayedBy ?lecturer)))
    (not (speaker (isPlayedBy ?speaker)))
    (not (isSameAs (isVerifiedBy ?lecturer ?speaker)))
  )
  (audience (isPlayedBy ?new_audience))
  (speaker (isPlayedBy ?new_speaker))
  (isSameAs (isVerifiedBy ?new_audience ?new_speaker))
  ?post_S2 <- (situation (name "S2"))
  =>
  (modify ?tr (lock (- ?l 1)))
  (assert (event (name "t2")))
  (modify ?pre_S1 (entities))
  (modify ?post_S2 (entities ?new_speaker ?new_audience))
)

```

FIGURE 5.3: Exemple de règles Jess produite à partir du réseau de Petri synchronisé

	Slides	Speaker	Audience
Slides	0.96	0.04	0.0
Speaker	0.0	0.89	0.11
Audience	0.37	0.05	0.58

TABLE 5.3: Matrice de Confusion pour la détection de situations (Caméraman)

de notre système d'acquisition. La totalité des usagers contactés ont préféré le second flux, indiquant qu'il permettait une meilleure compréhension de la présentation.

5.3 Bilan

L'utilisation des réseaux de Petri synchronisés a donné de bons résultats dans le cadre du réalisateur automatique. Cette application s'est montrée en particulier robuste aux bruits de perception alors que le formalisme utilisé (prédicats pour la description des situations, réseau de Petri pour l'enchaînement de ces situations) ne prend pas explicitement en compte la notion de bruit. La "rigidité" du modèle choisi est bien adaptée au cas d'une application où on ne s'intéresse pas à reconnaître un scénario parmi de multiples possibles mais plutôt à suivre l'évolution d'un seul scénario prédéfini. Il est donc bien adapté à des applications s'appuyant sur des scénarios scriptés.

Méta-modèle : RdP flou synchronisé

Nous avons présenté au cours du chapitre précédent un premier méta-modèle de reconnaissance basé sur les réseaux de Petri synchronisés. Ce méta-modèle est adapté au contrôle d'actions dans le cadre d'un scénario unique prédéfini.

Nous allons aborder au cours de ce chapitre un second méta-modèle basé sur les réseaux de Petri flous. L'objectif est de prendre en compte l'imprécision des données. Nous ne sommes plus dans un cadre d'un seul scénario dont on cherche à suivre le déroulement mais plutôt dans le contexte d'identifier un épisode observé parmi une bibliothèque de scénarios possibles. Cette bibliothèque n'est pas exhaustive : l'épisode observé n'a peut-être pas été décrit au préalable. Il faut donc pouvoir quantifier le degré de ressemblance de cet épisode parmi les scénarios connus. Nous avons proposé dans ce but un méta-modèle basé sur les réseaux de Petri flous.

6.1 Exemples d'approches

et al, () ont proposé un réseau Bayésien pour la reconnaissance d'activités humaines en se focalisant sur le changement de la position de la tête chez l'homme comme indice pour l'activité. Hongeng et al. () ont proposé un modèle Bayésien plus complexe basé sur une représentation hiérarchique de scénarios (hiérarchie de classifieurs Bayésiens). Une des difficultés pour l'utilisation de réseaux Bayésien est la nécessité d'injection de connaissances initiales sous forme de multiples probabilités. Un candidat naturel pour la reconnaissance de séquences temporelles bruitées est le HMM[ⓐ]. Yamato () a proposé par exemple cette approche pour la reconnaissance de coups au tennis. Duong () a utilisé le "Switching Hidden Semi-Markov Model" (S-HSMM) pour la reconnaissance d'activités

ⓐ. Hidden Markov Model

journalières. Ces approches cherchent à mettre en évidence des anomalies sur la durée d'un état plus que sur le séquençement de ces états.

Shi et al, () ont proposé le "Propagation Network (P-Nets) pour la représentation et la reconnaissance de séquences d'activités. La reconnaissance d'activité s'appuie sur l'utilisation d'un filtre à particules (DCondensation). Cette approche ne permet pas de reconnaître des séquences d'activités présentant des boucles (par exemple, le client fait plusieurs fois l'aller-retour entre l'accueil et le plan du bâtiment).

6.2 Relations floues



thèse de
Suphot
Chunwiphat

Le modèle de situation, tel que nous l'avons présenté lors du chapitre 2, s'appuie sur la notion de relation. Une relation est un prédicat entre les entités préselectionnées par les rôles. Le premier méta-modèle de reconnaissance (*réseau de Petri synchronisé*, chapitre 5) utilise une implémentation des relations sous la forme de prédicats de la logique du premier ordre. Dans le cadre du méta-modèle *réseau de Petri flou synchronisé*, les relations vont s'appuyer sur la logique floue.

De la même manière que la logique peut se construire à partir de la théorie ensembliste, la logique floue est définie au dessus de la théorie des ensembles flous proposée par Lofti Zadeh dans son article de référence (Zadeh, 1965). Un sous-ensemble flou A d'un ensemble (ou domaine) de discours E [ⓑ] est défini par une fonction d'appartenance μ_A indiquant pour tout élément x de E le degré d'appartenance de cet élément x au sous-ensemble A . Ce degré d'appartenance est compris entre 0 et 1. Soit $P(X)$ le prédicat X est A , A étant un sous-ensemble flou. La valeur de vérité du prédicat P pour la valeur x de la variable X est défini par l'équation 6.1.

$$P(X = x) = \mu_A(x) \tag{6.1}$$

Considérons par exemple une borne d'information dans un hall d'entrée. Les entités manipulées sont les personnes présentes dans le hall. La relation que l'on souhaite utiliser pour décrire une situation est $proche(P, borne)$. Le domaine du discours est \mathfrak{R}^+ , représentant la distance entre une personne et la borne d'information. Ce domaine du discours peut être couvert par deux sous-ensembles flous : *proche* et *loin* (voir figure 6.1). A une distance de $1m$, la valeur de vérité du prédicat est de 1 : il est totalement vrai. A $10m$, la valeur de vérité est 0 : totalement faux. A $3m$, la valeur est 0.8 : plutôt vrai. Remarque : le "découpage" du domaine de discours en sous-ensemble flou est de la responsabilité du développeur. Il s'appuie sur la sémantique que celui-ci donne aux termes utilisés (*proche*, *loin* etc). On constate que ce choix ne nécessite pas une grande précision numérique.

ⓑ. L'ensemble non flou de tous les éléments possibles

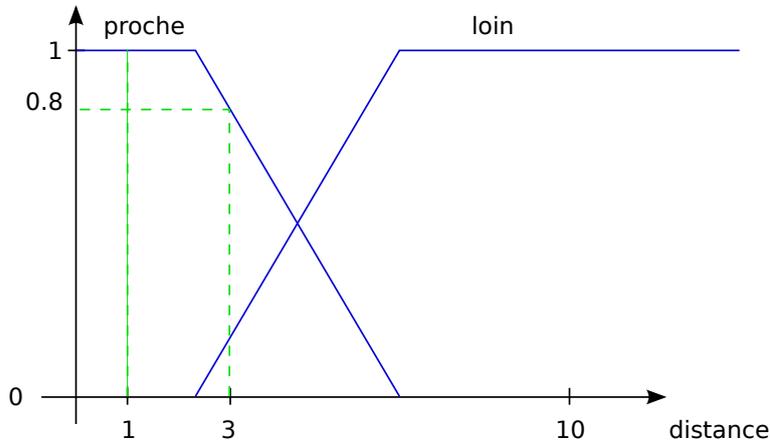


FIGURE 6.1: Exemple de relation floue

6.3 Réseau de Petri flou synchronisé

L'objectif du réseau de Petri flou est de relâcher le caractère instantané du tirage d'une transition. Dans la plupart des cas, le passage d'une situation à la suivante n'est pas immédiat : il existe un laps de temps pendant lequel les deux situations sont simultanément vraies, la précédente se désactivant alors que la suivante se met en place.

6.3.1 Situation et transition floue

Nous avons vu dans le cadre du méta-modèle réseau de Petri synchronisé (chapitre 5) que l'événement associé à une transition est émis lorsque l'ensemble des situations précédant cette transition ne sont plus valides et que les situations suivantes le sont. Par exemple, dans le cadre d'une transition entre $S1$ et $S2$ (équation 6.2) :

$$T = \neg valid(S1) \wedge valid(S2) \quad (6.2)$$

Pour prendre en compte le fait qu'une situation ne passe pas brutalement de l'état valide à non valide (et réciproquement), nous allons redéfinir cette validité de manière floue. Une situation est définie par un ensemble de rôles et de relations. Les rôles sélectionnent les entités, les relations évaluent ces entités sélectionnées. La validité d'une situation est un prédicat formé par la conjonction des relations et des rôles la définissant (équation 6.3).

$$valid(S) = Role_1 \wedge \dots \wedge Role_k \wedge Rel_1 \wedge Rel_2 \dots \wedge Rel_n \quad (6.3)$$

Cette équation s'étend naturellement à la version floue de la validité. Les relations floues sont définies par l'équation 6.1. En logique floue, une conjonction est définie par une

norme triangulaire $\textcircled{\vee}$. Il y a donc une infinité de conjonction. L'opérateur classiquement choisi est le min.

L'équation 6.2 définit une transition instantanée. Les situations étant maintenant floues, les transitions vont avoir une durée non nulle. Dans l'exemple précédent, il s'agit de représenter le passage progressif de la validité de $S1$ vers $S2$. $S1$ étant valide, la transition va s'activer dès que $S2$ commence à être valide. La transition s'achève lorsque $S1$ n'est plus valide.

Exemple : considérons deux situations $S1$ et $S2$. $S1$ est définie par "la personne est loin de la borne" et $S2$ par "la personne est proche de la borne". On suppose que $personne(P) = 1$ (on reconnaît parfaitement une entité de type personne). Le prédicat d'activation de la transition entre $S1$ et $S2$ est donc :

$$\begin{aligned} tirer_T(P) &= personne(P) \wedge loin(P, borne) \wedge pres(P, borne) \\ &= loin(P, borne) \wedge pres(P, borne) \end{aligned} \tag{6.4}$$

Le processus de transition est résumé Figure 6.2.

6.3.2 Evaluation d'une situation

L'équation 6.3 fournit la valeur instantanée de la validité d'une situation floue. La validité de la situation sur la totalité de la durée d'activation de la place associée est la moyenne de ces valeurs instantanées.

6.4 Evaluation d'un contexte

Notre réseau de Petri Flou est un modèle de scénario. Lors de la confrontation de ce modèle à un épisode réel observé, les différentes situations le composant ne seront pas toutes totalement actives. L'activation partielle de situations peut être le résultat d'erreurs de perception. Elle peut également correspondre, et c'est le cas qui nous intéresse le plus, au fait que la situation réelle ne correspond pas tout à fait à la situation *prototypique* qui a été décrite. Chaque activation de situation peut être interprétée comme une preuve supplémentaire que le scénario qui se déroule correspond bien au scénario modélisé. On se retrouve donc dans le cadre d'un problème de fusion de ces preuves afin d'évaluer à quel point un modèle de scénario correspond à la réalité observée. Nous avons réalisé cette fusion de preuves en nous appuyant sur deux outils :

$\textcircled{\vee}$. opérateur binaire respectant des contraintes de commutativité, associativité, monotonie et valeurs aux limites

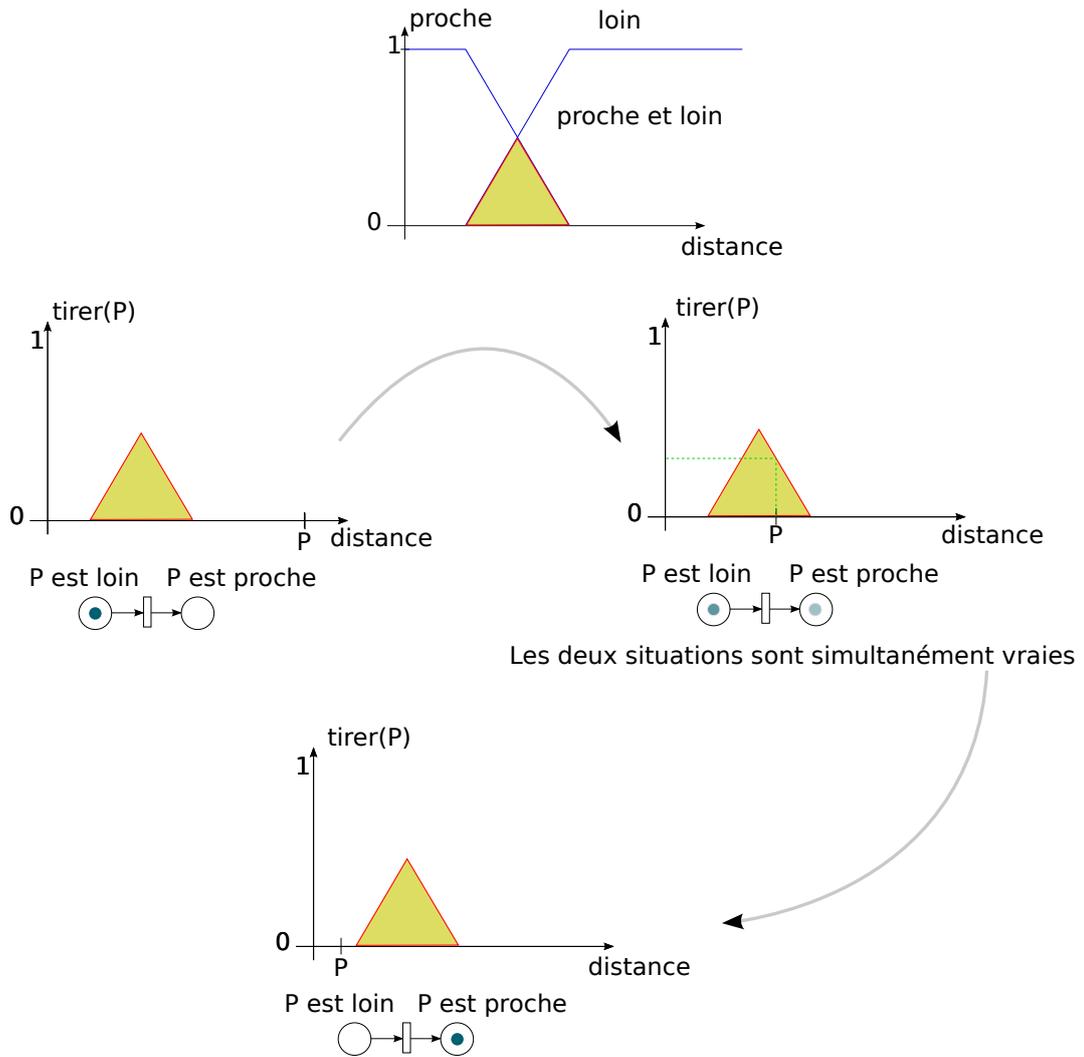


FIGURE 6.2: Exemple de transition floue entre situations

- la mesure floue comme indicateur de l'importance d'un ensemble de situations par rapport au contexte associé.
- les intégrales floues pour la fusion effective des éléments de preuve (degré d'activation des différentes situations), en se basant sur leur importance respective (mesure floue).

6.4.1 Mesure floue

En 1970, Sugeno a introduit la notion de mesure floue (Sugeno, 1977). Soit X un ensemble fini [Ⓞ] et $P(x)$ [ⓔ] l'ensemble de tous ses sous-ensembles. Une mesure floue est une fonction g respectant les conditions suivantes :

$$\text{domaine : } \quad g : P(X) \rightarrow [0, 1] \quad (6.5)$$

$$\text{valeurs aux limites : } \quad g(\emptyset) = 0, g(X) = 1 \quad (6.6)$$

$$\text{Monotonie : } \quad A \subseteq B \Rightarrow g(A) \leq g(B) \quad (6.7)$$

Les éléments $x_i \in X$ sont interprétables comme des éléments d'information tendant à estimer la présence de ce que l'on cherche (cf. critères dans l'aide à la décision multicritères). La mesure floue $g(\{x\})$ fournit l'importance que revêt l'élément d'information x par rapport à ce que l'on cherche. La mesure floue $g(\{x_i, \dots, x_k\})$, indique l'importance de la présence de l'ensemble des éléments d'information x_i, \dots, x_k . On retrouve cette interprétation dans les trois contraintes précédentes. En particulier, l'équation 6.6 indique qu'en l'absence d'information (\emptyset), on n'a aucun résultat. Si, en revanche, toutes les informations sont disponibles (X), alors on a ce que l'on cherche. L'équation 6.7 indique que plus on a d'information, plus on est en présence de ce que l'on cherche. Dans notre cas, l'élément d'information est la situation. Ce que l'on cherche, c'est la présence d'un contexte. La mesure floue indique l'importance de la présence d'un ensemble de situations dans l'identification d'un contexte.

En pratique, la construction d'une mesure floue commence par la détermination par un expert de l'importance de chacune des sources d'information (g^i) :

$$X = \{x_1, \dots, x_n\} : \forall i, g^i = g(\{x_i\}) \quad (6.8)$$

Ces valeurs g^i sont appelées les densités floues associées à X . Ces densités floues étant fournies, il s'agit alors de construire la mesure floue sur la totalité de $P(X)$. Le passage d'un singleton à un sous ensemble quelconque de X peut se faire grâce à l'opérateur d'union.

Ⓞ. Il est également possible de définir une mesure sur un ensemble infini

ⓔ. Power Set

Remarque : les équations 6.5, 6.6 et 6.7 définissant une mesure floue n'imposent aucune contrainte sur $g(A \cup B)$. Sugeno a proposé une famille particulière de mesures floues en ajoutant une contrainte supplémentaire sur $g(A \cup B)$: il s'agit des λ -mesures floues.

6.4.2 λ -mesure floue

L'objectif des λ -mesures floues est de pouvoir construire la mesure g sur la totalité de $P(X)$ à partir des densités floues g^i (importance des sources élémentaires d'information fournies par un expert). Cette construction est possible grâce à la contrainte imposée sur $g(A \cup B)$ (équation 6.9) :

$$\begin{aligned} \lambda > -1 \text{ et } \lambda \neq 0, \forall A, B \in P(X), A \cap B = \emptyset : \\ g_\lambda(A \cup B) = g(A) + g(B) + \lambda g(A)g(B) \end{aligned} \quad (6.9)$$

La valeur de λ n'est pas fixée par l'utilisateur. Elle doit être calculée afin de respecter la valeur aux limites $g(X) = 1$ (voir Tahani et Keller ([Tahani & Keller, 1990](#))).

6.4.3 Intégrale de Choquet

L'importance de la présence d'un ensemble de situations pour la reconnaissance d'un contexte est spécifiée (la λ -mesure floue g). On dispose également à partir des données de perception de l'activation d'une situation : la moyenne des activations floues instantanées (voir section 6.3.2). Appelons $h(s_i)$ cette activation. La prochaine étape consiste à fusionner les activations des situations, graduées par leur degré d'importance, afin d'évaluer l'appartenance à un contexte. Cette fusion est le rôle de l'intégrale floue de Choquet (voir ([Grabish, 2006](#)) par exemple).

L'intégrale floue de Choquet d'une fonction h (activation floue des situations) par rapport à une mesure floue g (importance de ces situations) est donnée par l'équation 6.10

$$\int_X f = \sum_{i=1}^n (h(\sigma(i)) - h(\sigma(i+1)))g(A_i) \quad (6.10)$$

où σ est une permutation sur l'ensemble des situations permettant d'ordonner ces situations par ordre d'activation décroissant $h(\sigma(i)) > h(\sigma(i+1))$ et A_i correspond au sous-ensemble des situation ordonnées à partir du rang i : $A_i = \{\sigma(i), \sigma(i+1), \dots, \sigma(n)\}$

Sous-ensemble X de S	$g(X)$
\emptyset	0
$\{s_1\}$	0.14
$\{s_2\}$	0.45
$\{s_3\}$	0.12
$\{s_1, s_2\}$	0.71
$\{s_1, s_3\}$	0.29
$\{s_2, s_3\}$	0.67
$\{s_1, s_2, s_3\}$	1.0

 TABLE 6.1: Exemple de mesure floue pour un contexte formé de 3 situations.

6.4.4 Exemple d'évaluation d'un scénario

Considérons par exemple un scénario composé de 3 situations : $S = \{s_1, s_2, s_3\}$. Un expert fixe tout d'abord le degré d'importance (densité floue) de la présence de chaque situation pour la reconnaissance du scénario. Ce sont par exemple les valeurs $g^1 = 0.14$, $g^2 = 0.45$ et $g^3 = 0.12$. A partir de l'équation 6.9, cette densité peut être calculée sur l'ensemble des sous-ensembles de S (voir table 6.1). Soit O l'observation d'une séquence de situations et de leur degré d'activation h correspondant. Cette séquence est découpée en sous-séquences correspondant au sous-ensemble de la mesure floue. Les activations individuelles des situations sont dans un premier temps combinées au sein de ces sous-ensembles par l'intégrale de Choquet (equation 6.10). La note obtenue correspond à l'estimation de présence du sous-groupe correspondant de situations. Toutes ces notes intermédiaires sont de nouveau combinées par l'intégrale de Choquet afin d'obtenir la note finale de correspondance entre la perception et le contexte. Ce processus est résumé Figure 6.3.

6.5 Expérimentation

Nous avons évalué la reconnaissance de scénarios par RdP flous synchronisés et intégrales de Choquet à partir d'un sous-ensemble des données produites par le projet Européen Caviar (Fisher & al., n.d.). Il s'agit de 13 vidéos filmées par une caméra grand angle dans le hall d'entrée de l'INRIA Rhône Alpes. Les entités détectées sont des objets (sacs, bureau d'accueil, borne d'information), les personnes et les groupes de personnes. Les rôles associés sont fournis Table 6.2. La détection des rôles est réalisée par un classifieur de type SVM.

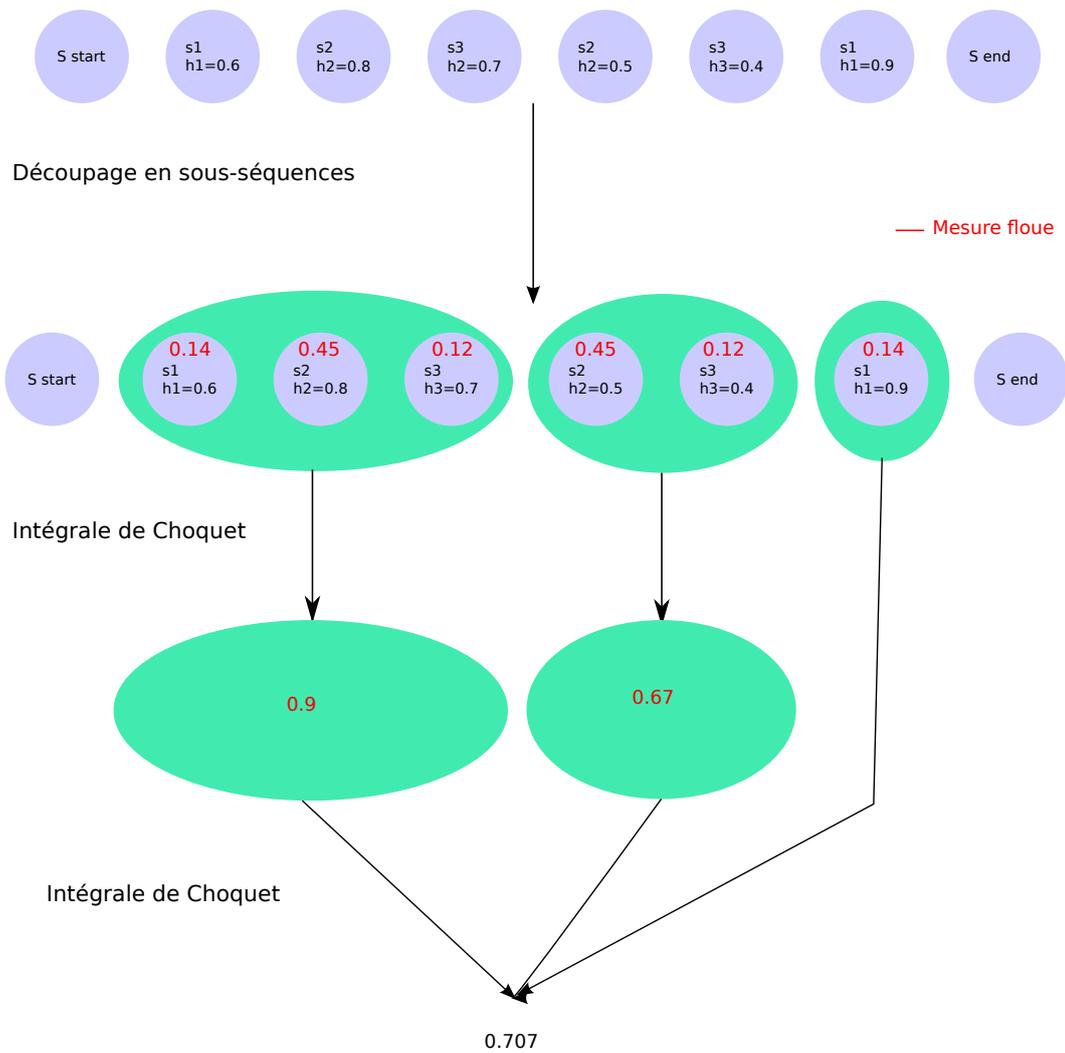


FIGURE 6.3: Exemple d'évaluation d'un scénario

<i>Objet</i>	<i>Personne</i>	<i>Groupe de personnes</i>
Left Object	Walker	Walkers
	Browser	Meeters
	Fighter	Fighters

TABLE 6.2: Liste des rôles détectés dans les vidéos Caviar

Nous avons modélisé 4 scénarios sous forme de RdP flous synchronisés :

1. *Browsing* : une personne marche entre les points d'intérêt de la scène (borne d'information, bureau d'accueil) et marque un temps d'arrêt à proximité de chacun de ces points.
2. *Leaving a bag* : une personne marche, s'arrête quelque part dans le hall et repart en laissant un objet derrière elle.
3. *Meeting* : plusieurs personnes marchent et se rejoignent pour former un groupe (le groupe est arrêté).
4. *Fighting* : plusieurs personnes marchent, forment un groupe. Une bagarre éclate au sein du groupe.

Les quatre réseaux de Petri ont été soumis aux 13 vidéos. Quelques-uns des résultats obtenus sont repris ci-dessous.

La figure 6.4 montre le degré de reconnaissance du modèle *Leaving a bag* lorsqu'on lui soumet l'ensemble des vidéos. Les vidéos 4, 5 et 6 correspondent bien au scénario



FIGURE 6.4: Degré de reconnaissance de "Leaving a Bag" sur l'ensemble des vidéos

modélisé. La vidéo 7 ne correspond que partiellement au scénario : l'utilisateur a repris son sac. La reconnaissance est bien partielle : 0.26.

La figure 6.5 montre le degré de reconnaissance du modèle *Browsing*. Ce comportement est bien reconnu dans les trois premières vidéos. Il est également bien reconnu dans la vidéo étiquetée *Person leaving a bag but then pick it up again*. En effet, dans cette vidéo, la personne s'est arrêtée à côté d'une borne d'information, a posé son sac pour regarder la borne. Elle a ensuite repris son sac et est repartie. Elle a donc exécuté, en plus de *Leaving a bag*, un scénario de type *Browsing*. De même, une partie du scénario de *Browsing* se retrouve dans l'ensemble des autres vidéos (se diriger vers un des points d'intérêt du hall).

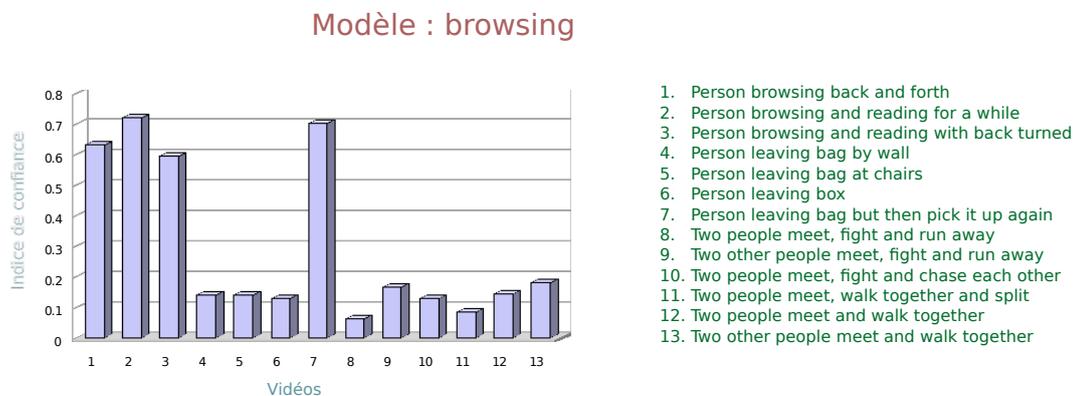


FIGURE 6.5: Degré de reconnaissance de "Browsing" sur l'ensemble des vidéos

Apprentissage des situations

Le modèle de contexte défini au cours du chapitre 2 propose une définition de la situation sous forme prédicative : la conjonction de rôles et de relations. Deux implémentations ont été réalisées : en logique classique (chapitre 5), fournissant une valeur binaire d'appartenance, et en logique floue (chapitre 6), permettant la construction d'une fonction continue d'appartenance. Il est possible d'utiliser d'autres représentations pour les situations. L'objectif de ce chapitre est de proposer, étant donné un choix d'un formalisme de représentation de situations, un cadre permettant d'apprendre automatiquement les situations décrivant un ensemble de scénarios observés.

7.1 Principe



thèse
d'Oliver
Brdiczka

Le point de départ de l'approche est une bibliothèque de séquences de perceptions obtenues à partir de l'observation de scènes réelles. Chacune de ces séquences contient une ou plusieurs situations.

L'apprentissage de la description des situations se déroule en trois étapes :

1. une première phase d'analyse des séquences. L'objectif est de découper automatiquement une séquence temporelle d'observations en sous-séquences adjacentes. Chaque séquence correspond à une situation
2. l'utilisateur expert (le développeur) donne un nom à chacune de ces situations. Plusieurs séquences peuvent avoir le même nom (décrire la même situation)
3. Une représentation de chacune des situations nommées est construite par apprentissage supervisé (segment de données \leftrightarrow nom de situation).

7.2 Segmentation automatique

7.2.1 Divergence de Jeffrey

L'objectif de la segmentation automatique est de proposer un découpage d'une séquence temporelle d'observations en sous-séquences adjacentes. Chaque séquence doit correspondre à une situation. L'hypothèse formulée est qu'une situation est caractérisée par une distribution des observations associées. Cette hypothèse a été expérimentalement validée dans le cadre de situations de type réunions formelles (groupes de travail autour d'une table) ou informelles (apéritif par exemple). Le principe consiste donc à découper l'axe temporel en points de séparation. Ces points délimitent des intervalles tels que la différence entre la distribution des observations de deux intervalles adjacents soit la plus grande possible (voir Figure 7.1). Dans le cadre d'observations discrètes, nous représentons la distribution de ces observations pour un intervalle de temps sous forme d'un histogramme. Cet histogramme est calculé sur un intervalle de temps correspondant à une fenêtre d'observation. La différence entre les histogrammes de deux intervalles adjacents est fournie par la divergence de Jeffrey (Puzicha et al., 1997). Cette divergence est en particulier utilisée en reconnaissance d'image afin de mesurer la similarité entre des textures. Le calcul de cette divergence est fourni par l'équation 7.1.

$$J_{p,q} = \sum_{x \in X} p(x) \times \log\left(\frac{p(x)}{\frac{p(x)+q(x)}{2}}\right) + q(x) \times \log\left(\frac{q(x)}{\frac{p(x)+q(x)}{2}}\right) \quad (7.1)$$

La recherche du découpage temporel est réalisé en faisant glisser deux fenêtres temporelles adjacentes le long de cet axe et en calculant la divergence entre ces deux fenêtres. Le pas de déplacement est fixé par le pas d'échantillonnage des données de perception.

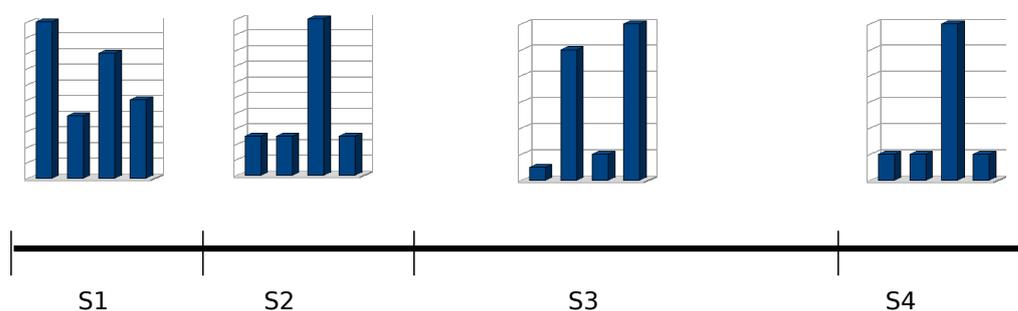


FIGURE 7.1: Découpage en situations afin de maximiser la différence de distribution de deux situations adjacentes

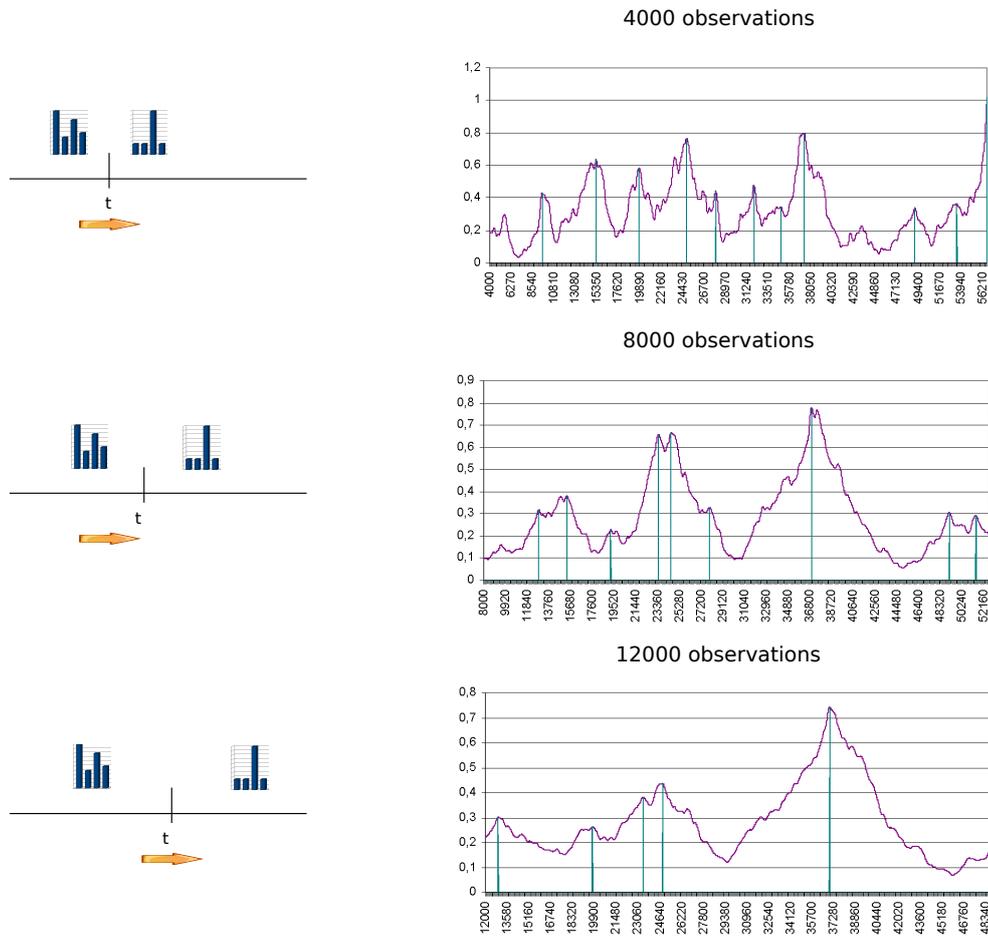


FIGURE 7.2: Calcul à chaque instant t de la divergence entre les deux intervalles autour de t (pour trois tailles différentes d'intervalles).

Les pics obtenus sur la courbe représentent les instants où la divergence entre les deux fenêtres de part et d'autre est maximum et correspondent donc à un changement net de situation. Ne faisant aucune hypothèse sur la durée d'une situation, cette recherche est effectuée pour différentes tailles de fenêtre. Cette approche est illustrée figure 7.2. Remarque : une grande fenêtre temporelle permet de réduire le nombre de pics (de frontières de situations) mais positionne ceux-ci avec moins de précision qu'une fenêtre de plus petite taille. L'étape suivante va consister à fusionner les pics provenant des différentes tailles de fenêtres.

7.2.2 Détection multifenêtres des pics

La détermination des pics est réalisée en fusionnant les informations fournies par les différentes courbes. Cette opération se déroule en deux étapes :

1. recherche des maxima pour chacune des courbes
2. fusion des différents maxima.

La recherche des maximums est réalisée par une approche de type “*Robust Mean Estimation*”. Cette approche a été proposée par Qian et al. (Qian et al., 1998) dans le cadre de la recherche d’un visage dans une image. La recherche robuste d’un maximum est basée sur un calcul successif de moyennes et d’écart types $(\mu(0), \sigma(0)) \dots (\mu(j), \sigma(j))$, chaque couple $(\mu(i), \sigma(i))$ étant calculé sur l’intervalle $[\mu(i-1) - \sigma(i-1), \mu(i-1) + \sigma(i-1)]$. Cette suite converge vers un intervalle $[\mu(k) - \sigma(k), \mu(k) + \sigma(k)]$ correspondant au pic recherché. Tous les points de cet intervalle sont ensuite retirés de la courbe afin de rechercher le pic suivant. Ce processus est répété tant qu’il reste des points au dessus de la valeur moyenne de la courbe initiale.

Cette recherche robuste de pics est répétée pour chacune des courbes (durée de l’intervalle de calcul des histogrammes). Tous les pics obtenus doivent ensuite être fusionnés. Chaque pic est caractérisé par une position et la taille de la fenêtre de calcul des histogrammes correspondants. La distance normalisée entre pic est tout d’abord calculée (voir Figure 7.3). Deux pics dont la distance est inférieure à 1 se recouvrent. Le plus grand des deux est conservé.

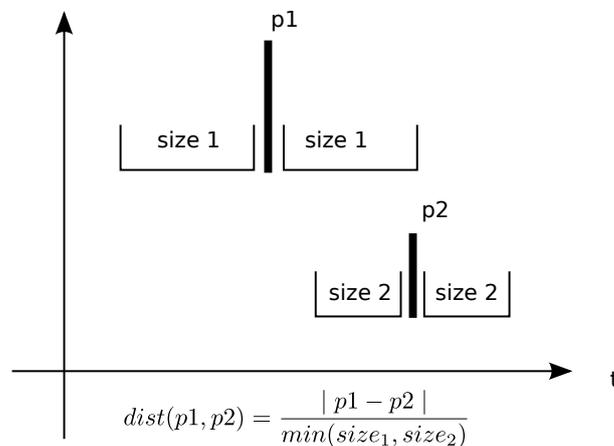


FIGURE 7.3: Distance normalisée entre deux pics.

Une fois cette opération de fusion effectuée, on supprime les pics peu significatifs. Un pic est jugé peu significatif si sa taille est inférieure à la moyenne de la taille des pics

pour la même fenêtre de mesure. Il est également peu significatif s'il n'est pas le résultat de la fusion d'au moins deux pics (présents dans au moins deux tailles de fenêtre).

7.2.3 Choix des segments

Le résultat de l'opération de fusion est la mise en place de n pics ($n+1$ intervalles). On recherche à présent parmi ces n pics, le sous-ensemble permettant le meilleur découpage en situations. Le critère retenu pour ce découpage est la plus forte séparation des situations produites, c'est-à-dire, la plus grande valeur moyenne de la divergence de Jeffrey. La présence de n pics nécessite l'évaluation de n^2 sous-ensembles (voir Figure 7.4).

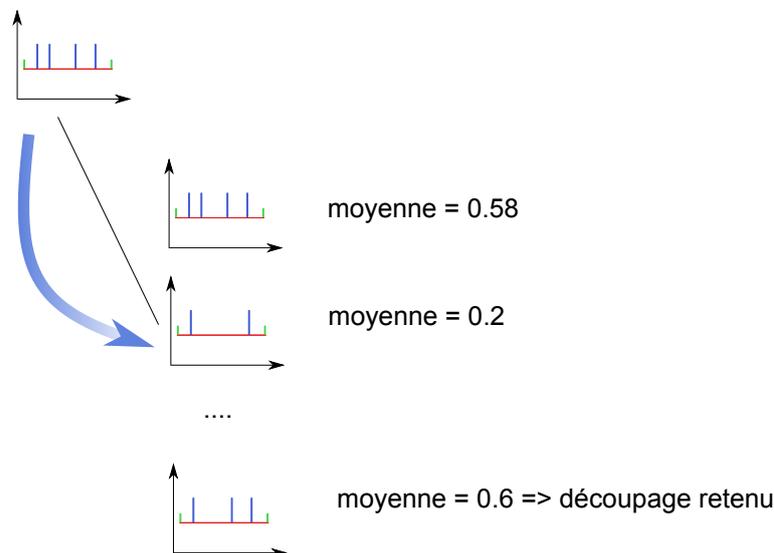


FIGURE 7.4: Choix du meilleur sous-ensemble de pics afin d'obtenir la séparation la plus "franche" des situations

7.2.4 Evaluation

La segmentation automatique en situations a été évaluée à travers plusieurs scénarios. Nous en détaillerons 2 : une discussion au sein d'un groupe, et un séminaire. Pour chacun de ces scénarios, la segmentation automatique est comparée à la segmentation manuelle réalisée par un opérateur humain. La qualité des segments obtenus est également mesurée en utilisant les critères proposés par Zhang et al (Zhang et al., 2004) :

- la pureté d'un segment (asp) : elle mesure à quel point un segment est limité à une seule activité.
- la pureté d'une activité (aap) : elle mesure à quel point une activité est présente dans un seul segment.
- la mesure globale (Q) : c'est une combinaison des deux autres mesures.

Les définitions de asp , aap et Q sont fournies Table 7.1

$$asp = \frac{1}{N} \sum_{i=1}^{N_s} p_{i,\bullet} n_{i,\bullet} \quad (7.2)$$

$$aap = \frac{1}{N} \sum_{j=1}^{N_a} p_{\bullet,j} n_{\bullet,j} \quad (7.3)$$

$$Q = \sqrt{aap \times asp} \quad (7.4)$$

$$p_{i,\bullet} = \sum_{j=1}^{N_a} \frac{n_{ij}^2}{n_{i,\bullet}^2} \quad (7.5)$$

$$p_{\bullet,j} = \sum_{i=1}^{N_s} \frac{n_{ij}^2}{n_{\bullet,j}^2} \quad (7.6)$$

Avec :

- n_{ij} nombre total d'observation pour l'activité j dans le segment i .
- $n_{i,\bullet}$ nombre total d'observation dans le segment i .
- $n_{\bullet,j}$ nombre total d'observation de l'activité j .
- N_a nombre d'activités.
- N_s nombre de segments.
- N nombre total d'observations.

TABLE 7.1: Calcul de la qualité des segments

Discussion au sein d'un groupe

5 enregistrements audio d'un groupe de 4 personnes ont été réalisés à l'aide de microphones Lapel. Ces enregistrements durent entre 9 et 16 minutes. Chaque microphone est relié à un détecteur d'activité vocale (Vaufreydaz et al., 2006), proposant une réponse

binaire (parle ou ne parle pas) toute les 16 ms. Ces 4 observations binaires sont combinées pour proposer un vecteur d'observation de $2^4 = 16$ valeurs.

Les quatre personnes ont formé durant l'enregistrement des groupes d'interaction (voir Figure 7.5). La séquence des groupes (qui parle avec qui) a été fixée par avance. Les participants étaient par contre libres de choisir la durée de chaque configuration. Les moins bon résultats de l'expérience 2 et 3 sont dûs en partie à un problème de qualité d'enregistrement. La figure 7.6 compare la configuration réelle des groupes (situations)

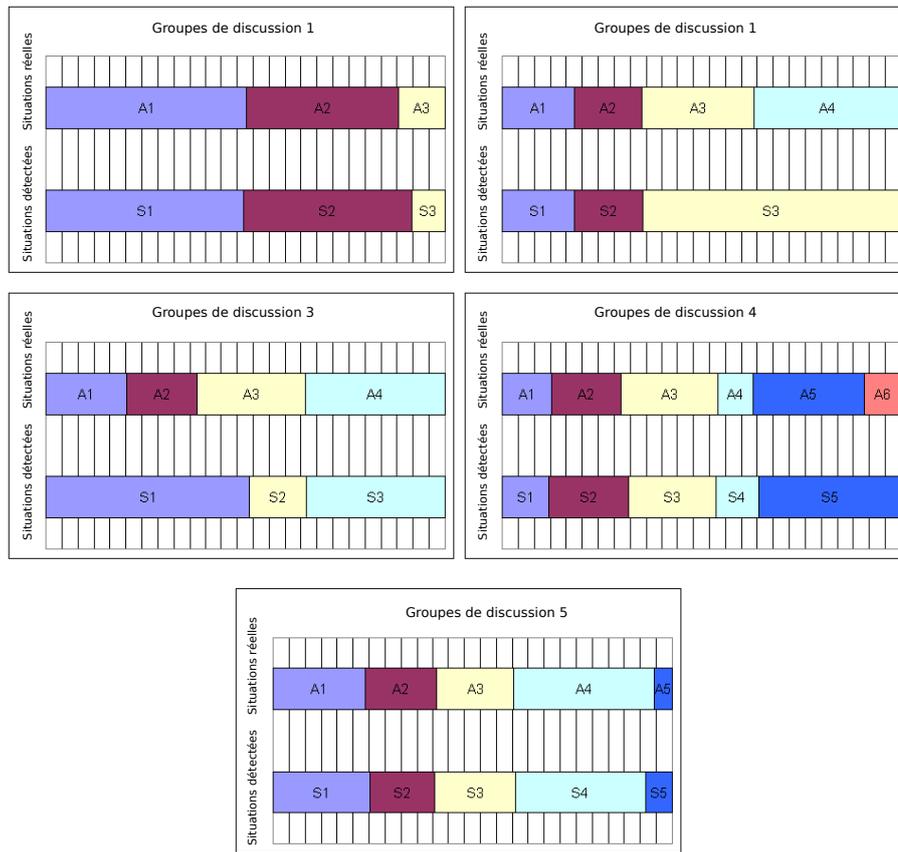


FIGURE 7.5: Exemple de configurations de groupes d'interaction

avec celles détectées et indique les valeurs asp , aap et Q pour chaque expérience.

Séminaire

Un séminaire de 25 minutes impliquant 5 participants a été enregistré. Le son a été capturé par 5 microphones Lapel et traité par un détecteur d'activité vocale (*parle*, *ne parle pas*). Les données perçues sont combinées en un vecteur d'observation de 2^5 valeurs. Le séminaire comprend quatre phases successives : "discussion en petits groupes"



	Durée	asp	aap	Q
Discussion 1	9m 14s	0.94	0.93	0.93
Discussion 2	10m 14s	0.68	0.99	0.82
Discussion 3	16m 11s	0.66	0.86	0.75
Discussion 4	14m 47s	0.78	0.91	0.85
Discussion 5	16m 12s	0.93	0.92	0.92
Moyenne		0.80	0.92	0.85

FIGURE 7.6: Situations réelles vs situations détectées

($D1$), "présentation" (P), "questions" (Q) et de nouveau "discussion en petits groupes" ($D2$). La figure 7.7 montre le déroulement des quatre phases et le résultat de la détection automatique. La valeur Q obtenue est de 0.90.

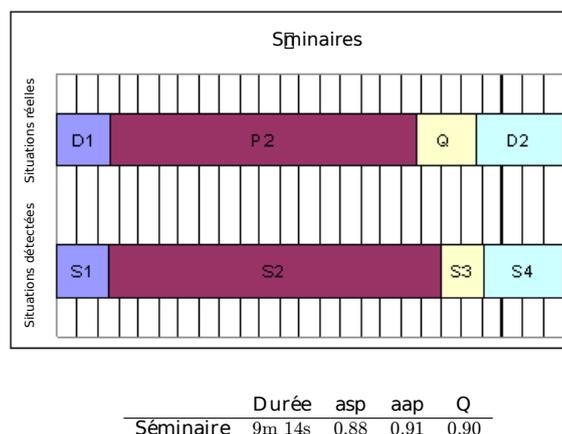


FIGURE 7.7: Situations réelles et détectées pour un séminaire

7.3 Représentation des situations

7.3.1 Principe

L'étape précédente a permis de construire m intervalles représentant n situations différentes. Chaque situation est associée à une étiquette fournie par l'expert humain (le développeur). $m > n$: plusieurs intervalles peuvent être associés à la même situation. L'étape suivante consiste à apprendre automatiquement une représentation pour chacune de ces n étiquettes. Cette représentation permet ensuite de mettre en place "en production", la reconnaissance de situations à partir des données de perception. De manière à être le plus générique possible, on ne privilégie pas un algorithme d'apprentissage particulier mais on s'appuie sur des familles d'algorithmes. Les familles peuvent être par exemple les HMMs, les réseaux de neurones, les arbres de décision . . . Ces familles sont constituées d'instances. Une instance est un choix d'une structure de données et d'un algorithme d'apprentissage. Pour la famille des HMMs, une instance peut être par exemple l'utilisation de l'algorithme EM pour l'apprentissage d'un HMM de 5 états.

L'objectif est de choisir la représentation la plus discriminante pour les situations par rapport aux perceptions associées. La recherche d'une représentation discriminante peut être réalisée en s'inspirant du critère de Fisher (Fisher, 1938) : on cherche à maximiser le rapport entre la distance inter et intra situations.

7.3.2 Exemple : la reconnaissance de séquences de vidéo surveillance

L'objectif est l'apprentissage et la reconnaissance de situations dans des séquences de vidéo surveillance. Les données utilisées proviennent du projet Européen Caviar[®]. Les différentes séquences vidéos fournies illustrent chacune une situation parmi l'ensemble des situations possibles : ("walking", "browsing", "fighting", "waiting", "object left"). A chaque vidéo est associé un fichier XML décrivant pour chaque image la liste des objets et leurs rôles. Ces données nous servent de perception. Quelques exemples d'images sont fournis Figure 7.8

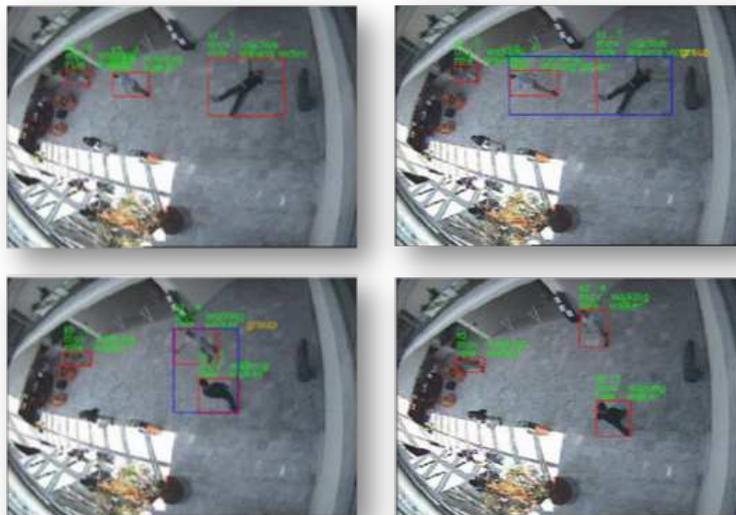


FIGURE 7.8: Exemples d'images extraites des vidéos Caviar

[®] <http://groups.inf.ed.ac.uk/vision/CAVIAR/>

Algorithme 1 : Algorithme de sélection des représentations des situations pour les données Caviar

répéter

Phase d'apprentissage

pour chaque *nom de situation* **faire**

┌ Choisir un nombre d'états pour le HMM

└ Apprendre les paramètres du HMM en utilisant EM

Phase de validation

Calcule du ratio :

– probabilité de la séquence de perception associée à une situation (HMM de cette situation)

– probabilité de la séquence de perception pour les autres situations (les autres HMMs)

jusqu'à *ratio optimal trouvé* ;

On choisit les HMMs comme formalisme de représentation des situations. Une instance de cette famille est donc un réseau HMM avec un nombre fixé d'état. L'apprentissage est réalisé à partir de 114 séquences de perception. Un tiers des séquences sont utilisées pour l'apprentissage, deux tiers pour la validation. La recherche de la représentation de situations la plus discriminante est explicitée par l'algorithme 1. Les résultats sont résumés Table 7.2.

7.4 Bilan

Nous avons mis en oeuvre une approche pour le découpage automatique de séquences d'observations en intervalles de temps correspondant à des situations. Cette première phase est non supervisée. Ces intervalles de temps servent ensuite de base pour l'apprentissage de représentations des situations correspondantes. Cet apprentissage est supervisé (séquence d'observations → étiquette d'une situation). Cette approche est assez lourde de part sa nature hors ligne. Elle n'est donc pas adaptée à un modèle de situation évoluant. Le processus de découpage temporel est non supervisé. Il se base uniquement sur une analyse statistique des données de perception : ce découpage n'est donc pas centré sur les besoins des usagers en termes d'actions. Le découpage en situation risque donc de ne pas être optimal selon ce point de vue. Cet aspect sera pris en compte dans le chapitre 9.

	Walking	Browsing	Fighting	Waiting	Object left
Walking (3 états)	0.8889	0	0.0635	0.0476	0
Browsing (3 états)	0.1944	0.8056	0	0	0
Fighting (7 états)	0	0	0.9167	0.0833	0
Waiting (7 états)	0.0476	0	0.1429	0.8095	0
Object left (5 états)	0	0	0	0	1

Classes	% vrai-positifs	% faux-positifs	Precision	Rappel	F-mesure
Walking (3 états)	0.8889	0.0692	0.9446	0.8889	0.9147
Browsing (3 états)	0.8056	0	1	0.8056	0.8857
Fighting (7 états)	0.9167	0.0695	0.6127	0.9167	0.7313
Waiting (7 états)	0.8095	0.0444	0.8519	0.8095	0.8194
Object left (5 états)	1	0	1	1	1
Total	0.8841	0.0366	0.8818	0.8841	0.8702

TABLE 7.2: Matrice de confusion et performance pour l'apprentissage de situations à partir des données Caviar

Le modèle de contexte tel que nous l'avons défini au cours du chapitre 2 est un modèle générique. Même s'il n'empêche pas une prise en compte de l'attention des utilisateurs, il ne propose également rien pour en faciliter la mise en place. Nous allons motiver au cours de ce chapitre l'utilisation de l'attention dans le cadre des environnements sensibles au contexte et nous allons présenter le modèle attentionnel proposé par Jérôme Maisonnasse au cours de sa thèse.

8.1 L'attention implicite



thèse de
Jérôme
Maisonnasse

Le modèle de contexte que nous avons proposé est un modèle générique. Aucune hypothèse n'est faite sur les types des entités, rôles ou relations manipulés, permettant ainsi la mise en place d'une grande variété d'applications. Le choix néanmoins d'une représentation de type prédicative induit une vision du contexte énumérable, plutôt indépendante de l'activité (même si rien n'interdit de la représenter). Cela a pour conséquence d'orienter les développements vers la création d'une représentation sémantique ou ontologique du contexte comme base de connaissances. Nous sommes dans ce que Dourish appelle une vision "représentationnelle" du contexte (Dourish, 2004).

Cette vision représentationnelle induit une prise en compte implicite de l'attention. L'attention se retrouve sous forme d'hypothèses au sein des relations entre les différentes entités modélisées. Par exemple, la détection d'un nouvel arrivant dans une pièce va déclencher le masquage des informations confidentielles d'un individu (en changeant par exemple le support de présentation de l'information). L'hypothèse implicite ici est que l'individu qui entre, de part sa présence, va porter attention aux informations de la première personne. De même, dans le scénario d'acquisition automatique de séminaire (cf. section 5.2), le changement des prises de vue est défini par rapport à la description

formelle des situations, des rôles et des relations entre les individus. Dans la description du rôle du présentateur et de l'audience, les concepteurs font l'hypothèse que l'audience et le présentateur se portent mutuellement attention. Mais aucun indice explicite ne vérifie la réalité de cette dépendance. Ces deux exemples font implicitement l'hypothèse de l'engagement attentionnel des utilisateurs sans complètement l'expliciter.

Par opposition à la vision "représentationnelle", Dourish propose une vision "interactionniste" du contexte. Le contexte est alors vu comme une propriété émergente de l'interaction (entre les personnes, les objets ...). C'est une propriété relationnelle entre l'environnement et les activités des utilisateurs. L'interaction est au coeur du processus.

La réussite de l'interaction repose sur l'intelligibilité mutuelle, c'est-à-dire la possibilité d'inférer les mêmes représentations au même moment :

« les significations portées par le langage mis en œuvre au cours de cette production de l'intelligibilité mutuelle ne reposent pas seulement sur une interprétation des unités linguistiques utilisées, mais aussi sur une interprétation d'autres éléments de la situation (Salembier & Zouinar, 2004). »

Imaginons un individu A dans une pièce. La fenêtre ouverte donnant sur la rue laisse passer le bruit du moteur d'une voiture. L'individu A de par ses moyens de perception va créer une représentation mentale de ce véhicule. Supposons que cet individu partage la pièce avec un deuxième individu B. De la même manière que l'individu A a eu accès au bruit du moteur, l'individu B va lui aussi pouvoir créer une représentation mentale de la voiture. Parmi les objets perçus par les deux individus, il y a leur présence respective. Alors comme pour la voiture, les deux individus peuvent créer une représentation supplémentaire de leur présence. A partir de ces représentations, sans avoir à décrire leurs situations, ils pourront directement communiquer sur les faits qui leurs sont communs, inférant le fait supplémentaire que l'autre a aussi eu accès aux mêmes informations. Cet espace de compréhension s'appelle l'intelligibilité mutuelle (voir Figure 8.1). Si telle n'était pas le cas, l'un des deux individus serait alors obligé de rapporter verbalement les éléments de contexte manquant à son interlocuteur pour qu'il puisse être compris. Les événements extérieurs (tels que le bruit de la voiture) ne sont pas les seuls éléments susceptibles de créer une intelligibilité mutuelle : il y a également les actes ostensifs tels que pointer du doigt, modifier la direction de son regard, énoncer le nom de la personne avec laquelle on souhaite communiquer etc. L'attention, en tant que processus servant d'interface entre le monde physique et nos représentations mentales, est un élément fondamental de la construction de cette intelligibilité mutuelle.

Il est donc important de redonner sa place à l'attention, de la sortir de *l'implicite* pour la rendre explicite dans nos modèles de contexte.

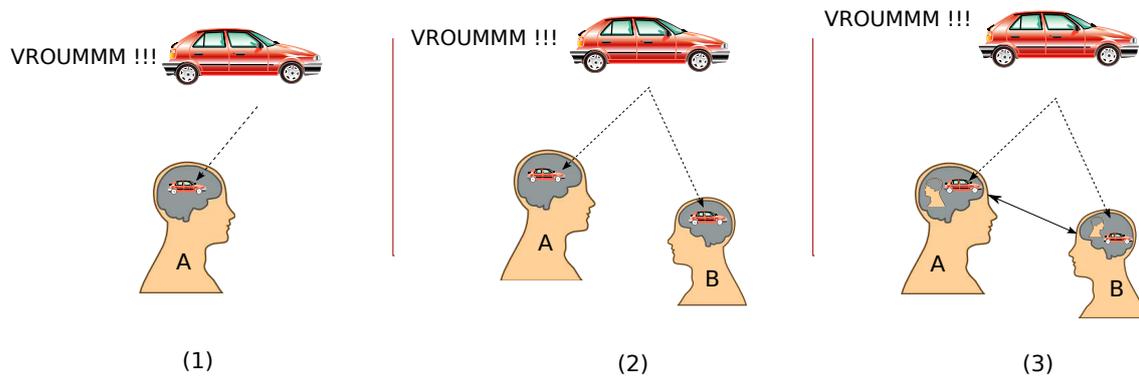


FIGURE 8.1: (1) A entend la voiture, (2) B entend la même voiture, (3) intelligibilité mutuelle entre A et B

8.2 Les systèmes attentionnels

Les systèmes attentionnels présents dans la littérature utilisent une estimation de l'attention des personnes comme donnée d'entrée. Ces différents systèmes proposent par exemple la réalisation d'interfaces attentionnelles pour le contrôle du démarrage ou de l'arrêt d'un service (Shell et al., 2004), la création d'un système de notification d'information (Horvitz et al., 2003), la gestion d'interruption (Mcfarlane, 2002) . . . Ces systèmes, désignés sous le nom "Attention User Interface" (AUI) regroupent plusieurs fonctionnalités qui sont : percevoir l'état attentionnel de l'utilisateur, raisonner à partir de cet état, communiquer, négocier graduellement le passage du focus de la périphérie au centre (Shell et al., 2003). Ils ont besoin entre autres :

- de détecter l'état attentionnel du sujet : vers quel objectif ou tâche courante est dirigée l'attention de l'utilisateur
- de choisir des stratégies de présentation de l'information pour maintenir l'attention ou au contraire, modifier l'état actuel de l'attention de l'utilisateur.

8.2.1 Détection de l'état attentionnel

La détection de l'état attentionnel s'appuie sur le recueil d'indices perceptibles par le système. Ces indices peuvent être de bas niveau (orientation des capteurs sensoriels de l'individu) ou de plus haut niveau (reconnaissance d'activité afin de déduire l'attention associée).

Concernant le bas niveau, la détection de l'état attentionnel se résume très souvent à la détection de l'orientation du regard ou de la tête. Cette mesure est très complexe (difficulté à détecter les yeux, faible résolution des images . . .). Mais la direction du re-

gard est effectivement une indication de l'attention dans de nombreux cas comme par exemple, lors de l'interaction verbale en face à face avec une autre personne. De même, une saccade oculaire provoquée par un événement surprenant (bruit, flash lumineux etc.) s'accompagne nécessairement d'un déplacement de l'attention vers la source de cet événement. Cela ne reste néanmoins pas une mesure totalement fiable. Dans certaines situations publiques où l'espace est réduit (un tramway par exemple), les regards ont tendance à se fuir. Ils risquent sinon d'être perçus de manière négative en empiétant l'espace personnel des autres. La direction du regard dans ce cadre ne reflète pas l'attention de la personne.

La détection de l'attention auditive reste également une tâche délicate. De même que pour le regard, la direction de la tête n'est pas un indice totalement fiable. Un conducteur aura par exemple son regard dirigé vers la route alors que son attention auditive pourra être totalement dirigée vers son passager avec lequel il est en conversation.

8.2.2 Stratégies de présentation

La prolifération des moyens informatiques ne doit pas devenir un handicap mais au contraire fournir plus de moyens et créer plus d'opportunités pour interagir de façon souple et discrète avec les utilisateurs. Chaque objet communiquant disposant de moyens d'interagir avec l'utilisateur constitue une opportunité. La stratégie de présentation va chercher à concilier des contraintes comme l'urgence du service, la confidentialité etc., avec l'état attentionnel estimé de l'utilisateur. Les questions posées pour la stratégie de présentation sont *qui, où, comment et quand* :

Qui ?

La plupart des travaux s'intéressent à des applications mono-utilisateur et ne se posent donc pas cette question. En situation multi-utilisateur, les individus appartiennent à un réseau de relations où tous les individus en interaction se prêtent plus ou moins attention. Si le système interrompt un utilisateur en interaction, tous les individus en interaction avec lui risquent d'être interrompus. Par exemple, lors d'un séminaire, un téléphone portable interrompant une personne de l'assistance ne va perturber que cette personne. Un téléphone mobile interrompant l'orateur va perturber la totalité de l'assistance. C'est donc un point qu'il est important de prendre en compte.

Où ?

Une fois l'attention déterminée, il s'agit de choisir parmi l'ensemble des périphériques disponibles (écrans, projecteurs, PDA etc.) celui présent dans le champ attentionnel de l'utilisateur et susceptible de pouvoir lui délivrer le message. Vertegaal et al. ont développé par exemple des objets dédiés à l'attention. Ils détectent la présence d'un regard sur eux et peuvent être équipés d'une paire d'yeux artificiels simulant ainsi la prise de contact par le regard. Ce sont des "Eyepliances" (voir (Shell et al., 2004; Vertegaal et al., 2005) par exemple).

Comment ?

C'est le rôle des "objets" ambiants ou des systèmes d'affichage périphériques (voir en particulier les définitions proposées par Pousman (Pousman & Stasko, 2006)). Le problème à résoudre est d'apporter de l'information à l'utilisateur en marge de sa tâche principale en veillant à ne pas consommer trop de ressources attentionnelles. La quantité d'attention détournée de la tâche principale doit en particulier être directement reliée à l'urgence de l'information à présenter. Par rapport à cet objectif, un système d'affichage périphérique peut se définir selon trois dimensions (Matthews et al., 2004) :

1. L'abstraction : cela consiste à réduire la fidélité de l'information de façon à la simplifier afin de consommer le moins possible d'attention. De plus, elle permet d'adapter un contenu suivant les capacités de présentation du dispositif. Ce problème est particulièrement étudié dans le cadre de la plasticité des interfaces.
2. Le niveau de notification : le niveau de notification est directement lié à l'importance de l'information. Plus l'information est importante et plus le périphérique va "émettre" avec force afin de saisir l'attention de l'utilisateur.
3. Les transitions : leur but est de faire coïncider l'affichage de l'information avec le degré d'urgence. Les transitions peuvent par exemple être des variations lentes de couleurs pour une information peu urgente ou à l'opposé la production de sons et de flashes lumineux.

A partir de ces trois dimensions, il est possible de définir des objets attentionnels sollicitant l'attention de l'utilisateur à différents niveaux : préattentionnel, inattentionnel, divisée et focalisée

Quand ?

Il s'agit pour le système de déterminer le meilleur moment afin d'entrer en interaction avec l'utilisateur. Une approche consiste par exemple à décrire la tâche globale sous forme d'une arborescence de tâches et sous-tâches (Adamczyk & Bailey, 2004). Le meilleur

moment pour déclencher une interruption est entre deux tâches (une interruption au cours d'une sous-tâche est perçue comme intempestive). Des études ont également été menées afin de déterminer la modalité d'interruption préférée des utilisateurs (Mcfarlane, 2002). Ces approches s'intéressent particulièrement aux propriétés de la tâche et de ces effets sur les processus cognitifs de haut niveau des utilisateurs. Elles sont orientées mono-utilisateur et ne peuvent donc pas appréhender les répercussions que peuvent avoir sur un ensemble de personnes la modification de l'attention de l'une d'entre elles.

En résumé, les travaux sur l'attention peuvent être regroupés en deux catégories : les systèmes raisonnant sur l'état attentionnel de l'individu et ceux proposant des stratégies de présentation. Les approches proposées sont orientées *mono-utilisateur*. Elles ne sont de plus à notre connaissance pas couplées : les modifications de l'environnement provoquées par la présentation de services ne sont pas réinjectées dans l'estimation de l'état attentionnel.

L'objectif de la thèse de Jérôme Maisonnasse est de proposer un modèle attentionnel *global*. Global signifie que ce modèle n'est plus centré sur l'utilisateur mais sur l'environnement et les différents utilisateurs et systèmes interactifs présents. Il permet de plus de prédire les conséquences induites par une modification de l'environnement sur l'attention de l'ensemble des acteurs présents. Ce modèle est présenté section 8.4.

8.3 Attention : cause ou conséquence ?

L'étude de l'attention relève de la psychologie cognitive, domaine que Jérôme, de part sa double formation (formation principale en psychologie et diplôme de compétences complémentaires en informatique) nous a aidé à aborder.

L'attention est un filtre influant sur le traitement de l'information. Elle crée un lien entre le monde extérieur et les états internes de l'individu. Elle agit sur nos perceptions, nos représentations, nos raisonnements, notre mémoire et conditionne nos comportements et notre activité. Elle nous permet de ne pas être submergé par la très grande quantité de sollicitations sensorielles auxquelles nous sommes exposés. Elle nous permet de nous isoler d'un grand nombre d'informations afin de permettre le maintien de notre activité courante. Néanmoins, si la situation l'exige, l'attention est capable d'interrompre cette situation courante afin de pouvoir réagir à de nouveaux faits (bruit, variation d'intensité lumineuse, ...). Elle permet donc une résistance à la distraction sans pour autant interdire une révision de notre plan d'action en fonction de données nouvelles. Il existe deux axes de modélisation de l'attention :

1. l'attention comme une cause.
2. l'attention comme une conséquence.

L'attention comme une cause présuppose la présence d'un "superviseur" responsable de sélectionner les données sensorielles retenues. L'attention est considérée selon deux axes : intensif et sélectif. L'axe sélectif introduit la notion de "spotlight [ⓐ]" : l'attention spatialement localisée. Les événements perceptifs ne sont pris en compte que s'ils se trouvent à l'intérieur de ce faisceau. La direction et l'ouverture du faisceau restent à la charge du superviseur. L'axe intensif considère que l'attention est une ressource limitée (métaphore du réservoir contenant une quantité finie de ressources attentionnelles). Cette attention sert d'énergie pour le traitement de l'information. Suivant la compétence de l'individu et de l'effort investi, une tâche cognitive va consommer plus ou moins de ressources attentionnelles. La performance de la tâche dépendra alors de la quantité d'attention disponible et allouée. Ce modèle rend compte de la capacité d'exécuter des tâches en « parallèle ».

L'attention comme conséquence est le résultat d'études récentes en neurosciences. L'attention est vue comme une conséquence du fonctionnement neuronal et s'appuie sur la métaphore de la « compétition ». L'attention est définie comme une propriété émergente dûe au fait que lorsque différents stimuli luttent pour accéder aux ressources de traitement, un seul gagne. Ce gagnant est celui qui sera perçu comme l'objet de notre attention. L'attention comme conséquence permet en particulier d'apporter une explication au superviseur de l'attention comme cause.

8.4 Modèle attentionnel proposé



thèse de
Jérôme
Maisonnasse

Comme nous l'avons précisé à la fin du paragraphe 8.2, l'objectif de la thèse de Jérôme est de proposer un modèle *global* qui explicite les relations entre les utilisateurs et les périphériques. De l'étude de l'attention en psychologie cognitive, nous retiendrons :

- l'attention est un filtre sélectif permettant de sélectionner l'information,
- elle permet de maintenir une activité courante en résistant aux distractions externes tout en n'interdisant pas de réviser le choix de l'activité si la situation l'exige (on reste sensible aux événements annexes). Il y a donc deux composantes :
 1. une composante *endogène* : l'attention volontaire (notre activité courante), la résistance aux perturbations.
 2. une composante *exogène* : l'attention reflexe (guidée par la perception).
- L'attention comme cause propose un axe intensif (répartition d'une quantité limitée de ressources) et un axe sélectif (spatialisation de l'attention)

[ⓐ]. pinceau attentionnel

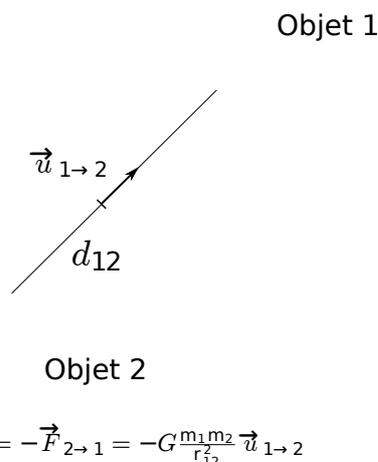


FIGURE 8.2: Définition de la force gravitationnelle

8.4.1 Métaphore gravitationnelle

Repartant des points précédents, la métaphore retenue pour le modèle attentionnel proposé par Jérôme est celle de l'attraction universelle. La loi de la gravitation formulée par Isaac Newton au 17^{ème} siècle stipule que deux objets dans l'univers exercent l'un sur l'autre une force gravitationnelle proportionnelle au produit de leur masse et inversement proportionnelle au carré de leur distance (voir figure 8.2). Le principe fondamental de la dynamique permet de déterminer l'accélération en fonction de la somme des forces. En particulier, le mouvement des objets de petite masse sera plus important vers les objets de masse supérieure et la trajectoire d'un objet possédant une vitesse initiale sera infléchi par la présence d'une masse à proximité (voir 8.3) Si l'on reconsidère le modèle attentionnel (composante exogène et composante endogène), la métaphore est donc la suivante :

- la vitesse initiale d'un objet soumis à aucune force définit une trajectoire rectiligne. Une accélération non nulle de cet objet va conduire à une modification de sa trajectoire. Plus la vitesse est grande, plus l'accélération devra être significative pour influencer notablement la trajectoire initiale. Le vecteur vitesse initial peut donc être vu comme la résistance à la distraction : c'est la composante **endogène** de l'attention. Elle correspond à l'intention : $\overrightarrow{Intention}$.
- L'accélération perturbe la trajectoire : elle représente donc la distraction, la composante **exogène** de l'attention. L'accélération correspond à la somme des forces. Dans le cadre de la loi gravitationnelle, la force est directement liée à la masse de l'objet. La masse est donc associée à la capacité de distraction : **la saillance** de l'objet. Cela peut être par exemple le niveau d'une sonnerie de téléphone, l'in-

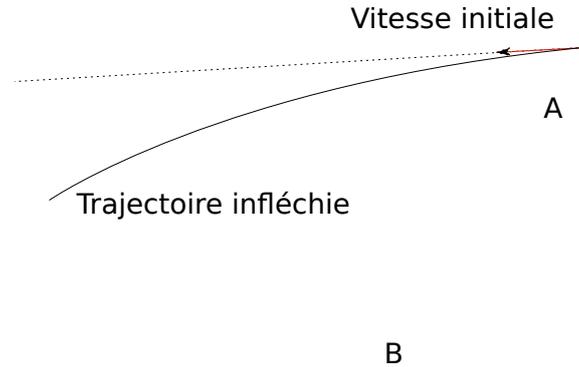


 FIGURE 8.3: Effet de la gravité sur la trajectoire d'une masse

tensité lumineuse d'un signal, sa fréquence etc. La somme des forces est le vecteur $\overrightarrow{Attraction}$.

L'intention d'un utilisateur ne peut pas être perçue directement mais doit être inférée en fonction d'indices (interactions inter-personnelles par la détection automatique de groupes de discussion, cf. (Brdiczka et al., 2009), interaction avec un système informatique etc.). La saillance est quant à elle fournie par l'utilisateur ou le concepteur. Un même objet peut posséder plusieurs saillances : un téléphone portable par exemple en mode vibreur ou en mode sonnerie.

La composante endogène et la composante exogène étant précisées, il faut maintenant les combiner pour construire le vecteur $\overrightarrow{Attention}$:

$$\overrightarrow{Attention} = \lambda \overrightarrow{Intention} + \mu e^{-2 \frac{\|\overrightarrow{Intention}\|}{max}} \overrightarrow{Attraction} \quad (8.1)$$

l'équation 8.1 n'a plus de lien avec la loi de la gravitation mais exprime simplement que l'attention est le résultat de la combinaison d'une composante endogène et exogène. *max* correspond à la valeur maximum que peut prendre le vecteur d'intention. Le facteur exponentiel permet de limiter les effets de l'attraction (distractions) si l'intention est proche de son maximum : si l'intention est forte, sa contribution à l'attention est prépondérante.

La compétition entre les deux vecteurs $\overrightarrow{Intention}$ et $\overrightarrow{Attraction}$ permet de modéliser la spatialisation de l'attention. La direction étant définie, il s'agit maintenant de distribuer les ressources attentionnelles aux objets présents.

8.4.2 Distribution des ressources attentionnelles

Comme nous l'avons présenté section 8.3, la répartition des ressources s'appuie sur la métaphore du pinceau attentionnel. Cette métaphore reprend les aspects sélectifs et intensifs. Nous allons considérer ici une personne i dont l'attention est déterminée par son intention propre et par un ensemble d'objets et de personnes saillantes j .

Focalisation du pinceau attentionnel

L'attention peut ainsi prendre la forme d'un pinceau extrêmement fin, focalisant une grande quantité des ressources attentionnelles sur les éléments présents directement dans l'axe du faisceau. A l'inverse, l'attention peut prendre également la forme d'un pinceau très large distribuant sur une grande surface, la même quantité de ressources mais de façon plus diffuse et moins concentrée. Cette focalisation du pinceau est modélisée par une famille de gaussiennes $F(\alpha, c)$ plus ou moins "aplaties", dont le sommet est fonction de la focalisation afin que la quantité de ressources distribuée reste la même (voir figure 8.4). Le paramètre c est fonction de la quantité d'attention disponible ($\|\overrightarrow{Attention}_i\|$). Une

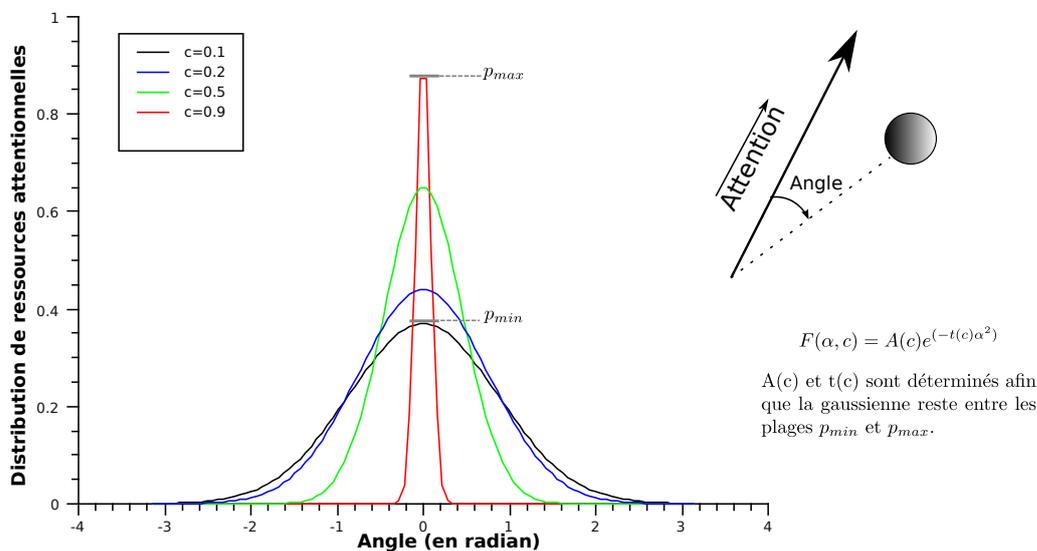


FIGURE 8.4: Distribution de la ressource attentionnelle en fonction de la focalisation c

attention importante, provoquée par une forte intensification ou une forte perturbation est généralement associée à une focalisation des ressources attentionnelles sur la cause.

Contribution à l'attention

La distribution des ressources attentionnelles doit se limiter aux objets ou personnes ayant participé à la détermination du vecteur $\overrightarrow{Attention}$. La contribution d'un artefact j à l'attention de la personne i est fournie par l'équation 8.2, où $\overrightarrow{a_{i \rightarrow j}}$ représente l'attraction de i par j .

$$contrib_{i,j} = \frac{\|\overrightarrow{a_{i \rightarrow j}}\|}{\|\overrightarrow{Attention_i}\|} \quad (8.2)$$

Distribution des ressources

Les ressources ne sont distribuées qu'auprès des personnes et non auprès des objets inertes. Les objets j ayant une saillance non nulle, ayant donc participé à la détermination du vecteur $\overrightarrow{Attention_i}$ sont ordonnés par angle croissant par rapport à ce vecteur. On parcourt ensuite cette liste triée. Pour tout objet j , tant que la quantité de ressource attentionnelle restante n'est pas nulle, la personne i en prélève une quantité $ressource_{i \rightarrow j}$ associée à l'objet j . Cette quantité est fonction de la contribution de l'objet j à l'attention de la personne i et de la focalisation du pinceau attentionnel (en particulier, de l'angle entre j et la direction de $\overrightarrow{Attention_i}$) :

$$ressource_{i \rightarrow j} = F(\alpha_{i,j}, c) * contrib_{i,j} \quad (8.3)$$

Cette répartition est illustrée Figure 8.5. En répétant cette opération pour l'ensemble des personnes présentes dans la scène, on obtient n vecteurs constituant la matrice attentionnelle.

Matrice attentionnelle

La matrice attentionnelle $[m_{ij}]$ est calculée à chaque instant d'une séquence d'information. Les personnes sont représentées en lignes alors que l'ensemble des entités sont représentées en colonnes. Les entités sont indifféremment des objets, des utilisateurs ou tout objet pouvant agir d'une façon ou d'une autre sur l'attention des utilisateurs. Chaque cellule de la matrice contient l'information : la personne i fait attention à l'entité j avec une quantité de ressources attentionnelles m_{ij} .

Chaque ligne représente une information égocentrée qui qualifie et quantifie la relation attentionnelle avec les autres objets et les personnes alentours par rapport à une personne cible. Cette relation signifie “*quelle attention une personne accorde aux objets alentours ?*”. Chaque colonne représente une information exocentrée et correspond à la question : “*comment les autres me portent-ils attention ?*”.

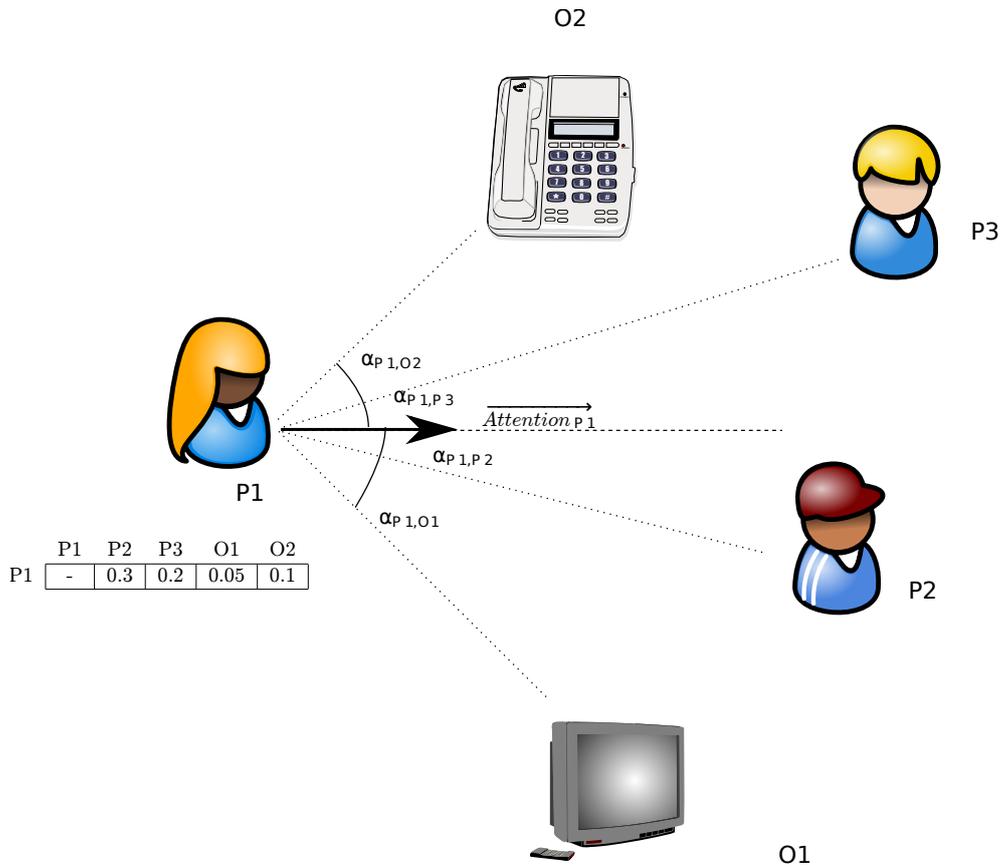


FIGURE 8.5: Distribution des ressources attentionnelles que P1 affecte à P2, P3, O1 et O2

8.4.3 Stratégies d'interaction

Comme nous l'avons indiqué dans la section précédente, la matrice attentionnelle peut être calculée à chaque instant en fonction des données perçues : intention des usagers, modification de la saillance des objets (télévision qui s'allume, téléphone qui sonne etc). Elle reflète alors l'état de l'attention des personnes présentes. Elle peut également être utilisée afin de prédire l'évolution de l'attention de l'ensemble des personnes présentes par rapport à un choix d'interaction. On peut également inversement l'utiliser afin de rechercher une stratégie d'interaction par rapport à un objectif fixé.

Considérons par exemple la matrice présentée dans la table 8.1. Un message urgent peut par exemple être communiqué à la personne $P3$ en utilisant l'objet $O1$ auquel il accorde déjà une grande attention (lecture egocentrée de la matrice : ligne violette). Si maintenant on souhaite délivrer un message confidentiel à $P3$, $O1$ n'est plus le bon choix car $P1$ lui accorde également une part importante de son attention (lecture exocentrée :

colonne verte). De même, la colonne bleue permet d’anticiper que distraire $P2$ aura comme conséquence de distraire également $P4$. Les valeurs présentes dans la matrice permettent

	P1	P2	P3	P4	O1	O2	O3
P1	-	0.15	0.02	0.02	0.30	0	0
P2	0.5	-	0.11	0.04	0.05	0.1	0
P3	0.04	0.2	-	0.12	0.54	0.07	0.08
P4	0.04	0.6	0.17	-	0.12	0.2	0.4

TABLE 8.1: Exemple de matrice attentionnelle

également d’estimer l’engagement d’un utilisateur dans une tâche donnée et de prévoir si l’engagement dans une nouvelle interaction risque de l’amener à interrompre sa tâche en cours faute de ressources supplémentaires disponibles. Par exemple si un utilisateur est déjà engagé dans une tâche de type résolution de problème nécessitant 80% de ses ressources attentionnelles, il ne lui restera pas suffisamment de ressources disponibles afin de gérer l’interruption provoquée par un service informatique : il ne sera pas en mesure de poursuivre sa tâche principale. Par contre, s’il est engagé dans une conversation de nature spontanée ne nécessitant que 30% de ses ressources, la nouvelle interaction pourra être intégrée sans qu’il ait à stopper sa conversation initiale bénéficiant après redistribution d’assez d’attention pour être maintenue. La table 8.2 propose un parallèle entre ressources attentionnelles et niveau de traitement cognitif.

8.5 Validation expérimentale

Nous n’avons pas à ce jour utilisé cette approche dans une application sensible au contexte. Nous avons uniquement cherché à vérifier sa ”validité” par rapport aux sciences sociales. La validation a été réalisée en comparant les résultats fournis par le modèle attentionnel par rapport à des résultats établis en psychologie. Nous nous sommes en particulier intéressé à l’effet de la spatialisation sur l’attention des individus impliqués dans des petits groupes de travail.

Un certain nombre d’études en psychologie ont analysé les interactions entre les individus dans les petits groupes de travail de type ”réunion”. Par exemple, dans le cadre d’un groupe de travail réuni en cercle, la fréquence d’échanges verbaux est maximum avec l’individu en face et décroît vers les individus adjacents. Ce phénomène est connu sous le nom d’effet *Steinzor*. Leyens et Rimé (Leyens & Rimé, 1974) ont mis en évidence expérimentalement que ce phénomène, comme le soupçonnait Steinzor, existe également dans le cadre d’une interaction non verbale. Nous avons modélisé 6 personnes assises

Ressources	Niveau de traitement cognitif	Niveau de disponibilité cognitive	Niveau de traitement	Niveau de traitement cognitif
0%	Nulle	Très forte	Inconscient	Aucune tâche n'est réalisée
5%	Nulle	Forte	Pré-attentives	Tâche de détection basée au niveau perceptif (couleur, bruit, orientation, mouvements etc.)
15%	Automatique / Reflexe	Moyenne	Inattentif	Tâches sur-apprises devenues reflexes et ne nécessitant plus de contrôle conscient (conduite, discussion, etc.)
25%	Règles	Limitée	Divisé	Tâche qui met en oeuvre des règles et nécessite un contrôle conscient mais assez bien maîtrisé autorisant l'exécution d'autres tâches en parallèle
50%	Connaissances	Très faible	Focalisé	Tâche qui requiert la remise en question des savoirs et savoirs faire. Nécessite un important effort cognitif comme la résolution de problème, la mémorisation etc.

TABLE 8.2: Correspondance entre ressources attentionnelles et niveau de traitement cognitif

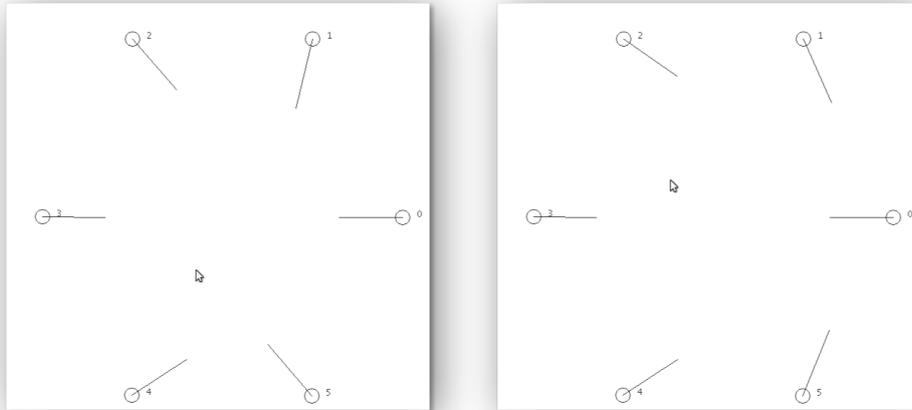


FIGURE 8.6: Effet Steinzor lors d'une réunion en cercle

autour d'une table. Ces personnes ne parlent pas, n'ont pas d'intention et ont toute la même saillance. La direction de l'attention de chaque personne est bien dirigée vers la personne située en face. (partie gauche de la figure 8.6). Le statut social peut être modélisé par le niveau de saillance. On peut modéliser par exemple le chef de la réunion en augmentant sa saillance (partie droite de la figure 8.6).

Deuxième partie

Point de vue utilisateur

Apprentissage supervisé

Nous allons dans un premier temps indiquer pourquoi nous nous sommes tourné vers l'apprentissage automatique dans le cadre de l'aide apportée à l'utilisateur final pour la mise en place de son application. Nous présenterons ensuite l'approche de type apprentissage supervisé mise en place dans le cadre de la thèse d'Oliver Brdiczka.

9.1 Pourquoi l'apprentissage automatique ?

Nous avons évoqué au cours de la section 1.4 l'assistant automatique proposé par Pattie Maes (Maes, 1994). Pattie Maes a en particulier conclu qu'il n'était pas envisageable de réaliser cette application en s'appuyant uniquement sur un développeur ou en demandant à l'utilisateur final de la spécifier entièrement. La solution qu'elle propose est d'apprendre les services à rendre dans chaque situation. Cet apprentissage a l'avantage d'être très évolutif. Le système a la capacité de s'adapter au changement de préférences de l'utilisateur sans que celui-ci ou le développeur n'ait besoin d'intervenir.

Byun et Cheverst (Byun & Cheverst, 2001), tout comme par exemple Godoy et Amandi (Godoy & Amandi, 2005) estiment également que le mode de vie de l'utilisateur est enclin à changer au cours du temps et que le système doit s'adapter à ces changements. Remagnino et Foresti (Remagnino & Foresti, 2005) pensent que l'avenir de l'Intelligence Ambiante réside dans la recherche en apprentissage automatique. Selon eux, les modèles pour les systèmes intelligents et les associations entre les sorties des capteurs et un comportement intelligent sont de loin trop complexes pour être câblés manuellement.

Nous nous sommes donc tournés vers les approches de type apprentissage automatique pour intégrer les contributions de l'utilisateur au fonctionnement de l'application.

9.2 Cadre général

L'application livrée par le développeur possède un modèle initial de contexte "général", devant convenir au plus grand nombre. L'utilisateur final peut donc la tester en situation afin d'éprouver ce modèle et fournir un retour pour l'adapter à ses besoins. Comme nous l'avons indiqué dans le paragraphe précédent, il ne s'agit pas pour l'utilisateur de reprogrammer le modèle de contexte, mais plutôt d'exprimer ses besoins de la manière la plus directe possible. Nous avons considéré deux approches :

- L'utilisateur revoit une séquence d'utilisation de l'application. Il exprime les corrections qu'il veut mettre en place en indiquant directement les actions souhaitées. Nous sommes dans un cadre d'apprentissage supervisé (abordé au cours de ce chapitre).
- L'utilisateur, lors de l'utilisation de l'application, exprime éventuellement son mécontentement ou sa satisfaction. Nous sommes dans un cadre d'apprentissage renforcé, abordé au cours du chapitre 10.

9.3 Retour de l'utilisateur

Dans un premier temps, l'utilisateur va jouer le rôle de superviseur (apprentissage supervisé). Il visualise une séquence d'utilisation de l'application. Il est considéré comme fiable : il fournit des indications cohérentes. Il peut fournir trois types de retour :

1. *Correction d'action* : l'action proposée par l'application n'est pas satisfaisante. L'utilisateur indique directement l'action qu'il aurait souhaitée pour la situation concernée. Remarque : ce point inclut également le cas où l'application ne propose aucune action
2. *Suppression d'action* : l'application propose une action mais l'utilisateur ne souhaite aucune action particulière.
3. *Aucune modification* : l'action proposée est correcte. L'utilisateur ne donne aucun retour au système.

L'étape suivante consiste à réaliser hors ligne un ensemble de modifications du modèle de contexte afin d'incorporer l'ensemble des remarques émises.

On retrouve ce type d'approche dans le système aCAPpella de Dey et al. (Dey et al., 2004). Ce système collecte dans un premier temps des données *en situation* à partir des capteurs déployés (caméras, microphones, ...). L'utilisateur peut ensuite, à l'aide d'une interface dédiée, rejouer les séquences enregistrées et visualiser les situations reconnues par le système. Il peut annoter l'enregistrement afin de décrire les situations qui lui semblent pertinentes ainsi que les actions associées. aCAPpella apprend à partir



de ces données (perceptions + annotations) un modèle de situation à l'aide de Réseaux Bayésiens Dynamiques. Ce processus de test-annotation-apprentissage est répété jusqu'à convergence. Cette approche dans son principe correspond à ce que l'on souhaite mettre en oeuvre. Le modèle de situation choisi (Réseau Bayésien Dynamique) ne nous semble par contre pas adapté. Il ne permet pas de fournir facilement une explication à l'utilisateur final sur son fonctionnement et les raisons de ses prises de décision, critère important pour l'établissement de la confiance.

On peut également citer l'approche de Mozer dans la "Neural Network House" (Mozer, 1998), une des premières applications d'intelligence ambiante, basée également sur l'apprentissage supervisé. Dans cette application, un réseau de neurones apprend à prédire l'occupation des différentes pièces afin d'optimiser le confort des habitants. Les informations sont fournies par l'utilisateur lors de la manipulation des systèmes de contrôle : si l'utilisateur baisse la luminosité de la pièce en entrant, cela signifie que la luminosité proposée n'est pas correcte et que la nouvelle valeur fournie doit servir pour l'apprentissage. Le formalisme choisi (réseaux de neurones) ne répond pas ici non plus au critère d'explicabilité. Cette approche présente néanmoins l'intérêt de recueillir les informations de l'utilisateur *in situ* et non pas hors ligne. Elle se rapproche de ce fait de l'apprentissage renforcé que l'on verra au chapitre suivant, même si cela reste de l'apprentissage supervisé.

9.4 Approche générale proposée

La base d'exemples pour l'apprentissage est constituée d'un ensemble de situations et de demandes de modifications associées :

$$B = \{(S_1, M_1), \dots, (S_n, M_n)\} \quad M_i \text{ étant soit une correction, soit une suppression}$$

Remarque : une situation S_i peut être présente plusieurs fois dans la base d'exemples (si cette situation s'est présentée plusieurs fois lors de la séquence annotée par l'utilisateur final).

Pour chacun de ces exemples, deux types de modifications du modèle de contexte sont possibles :

- l'exécution de la modification M_i pour la situation S_i (sans modifier la situation).
- le redécoupage de la situation S_i concernée.

Le premier cas, le plus simple, s'applique lorsque la même modification M est demandée pour l'ensemble des occurrences de S_i dans la base d'exemples B . Cette demande de modification est alors exécutée (remplacement ou suppression de l'action associée à S_i).

Le second cas s'applique lorsque des modifications différentes sont demandées pour l'ensemble des occurrences d'une situation S_i . Cela signifie que la situation est trop générale et n'incorpore donc pas des éléments de perception (rôles et relations) permettant de décider quelle action doit être exécutée. Ce problème est relativement courant. Lors de la mise en place du modèle initial de contexte, le développeur ne va pas chercher à représenter l'ensemble des situations possibles mais va plutôt naturellement créer des situations les plus générales possibles pour en diminuer le nombre total.

Une situation initiale trop générale pourrait par exemple être *travailler au bureau*. Il peut être nécessaire, après avis de l'utilisateur, de redécouper cette situation en trois sous-situations : *travailler sur le PC*, *lire un article* et *répondre au téléphone*. Ce redécoupage est illustré figure 9.1 Remarque : afin de réduire la complexité du problème, nous avons fait l'hypothèse qu'une seule sous situation peut être active à la fois.

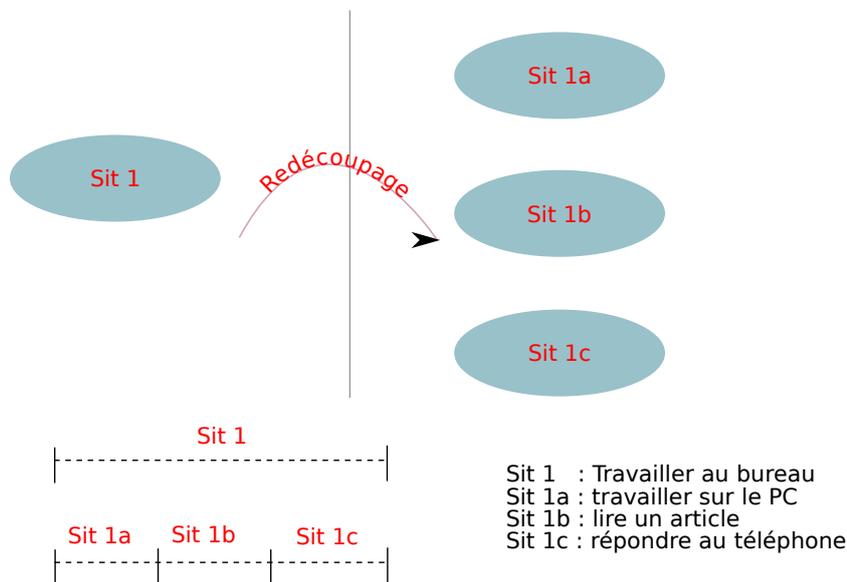


FIGURE 9.1: Exemple de redécoupage d'une situation en 3 sous-situations

Les situations sont construites à partir d'un ensemble prédéfini de rôles et de relations. Le problème de redécoupage de situations peut être abordé comme un problème de classification dont les classes sont les actions demandées (incluant la suppression d'une action) et les données d'entrée l'ensemble des rôles et des relations disponibles.

Comme nous l'avons déjà indiqué au cours du chapitre 7, il n'existe actuellement pas d'approche "universelle" susceptible de fonctionner pour n'importe quelle application et dans n'importe quel environnement. Nous avons donc remis en place une approche

générique permettant la mise en compétition des différents type de classifieur, afin de sélectionner celui adapté aux données en présence. Cette approche est rappelée figure 9.2.

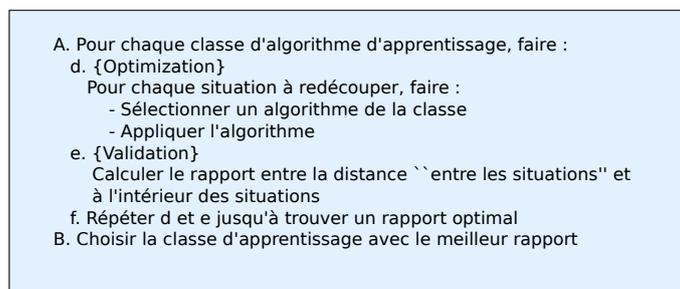


FIGURE 9.2: Algorithme général pour le redécoupage de situations

9.5 Evaluation

9.5.1 Scénario initial

Afin d'évaluer notre approche, nous sommes partis du scénario prédéfini suivant. Ce scénario se déroule dans un bureau équipé d'une caméra. Un tracker 2D permet de suivre les personnes. Des zones de détection ont été mises en place pour l'interprétation de la perception (rôles codés manuellement). Les autres données de perception sont fournies par l'agenda et par l'activité sur le PC. Le scénario est composé de 9 situations : $S0$: la pièce est vide, $S1$: une personne entre dans la pièce, $S2$: la personne se connecte/travaille sur le PC, $S5$: la personne connectée s'assoit sur le canapé, et $S8$: une présentation se déroule dans le bureau. Les trois autres situations sont associées à l'agenda. Elles concernent l'arrivée d'un rappel de rendez-vous alors que l'on est en train de travailler sur le PC ou que l'on est dans le canapé (respectivement $S3$ et $S6$ si le rendez-vous est au campus et $S4$ et $S7$ si le rendez-vous est dans le bureau). Les actions associées à ces situations sont entre autre mettre de la musique lorsque l'utilisateur est assis sur le canapé. Ce scénario est illustré figure 9.3.

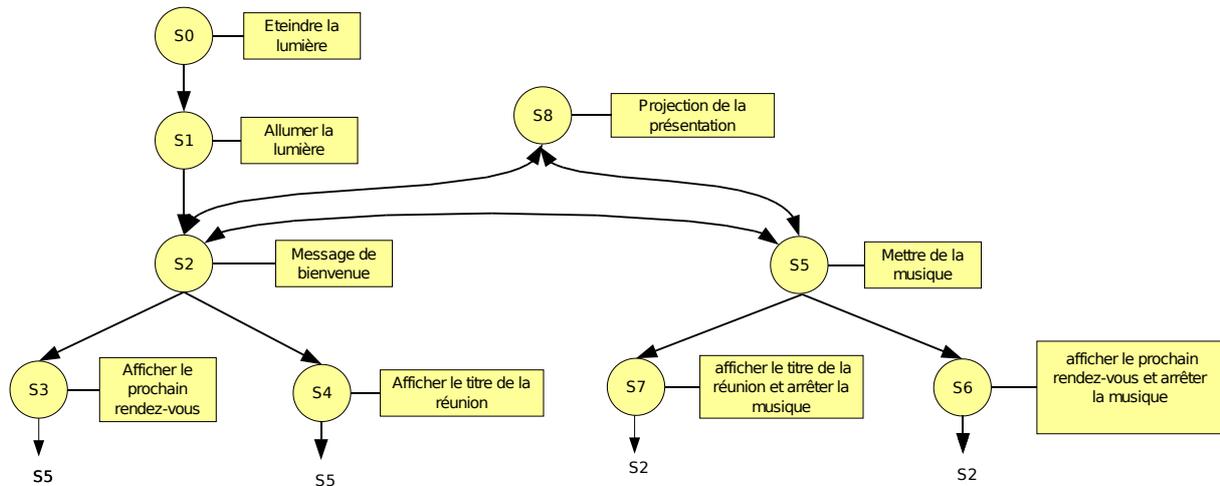


FIGURE 9.3: Scénario Initial

9.5.2 Retour de l'utilisateur

Nous allons nous intéresser uniquement aux demandes de l'utilisateur conduisant à un redécoupage de situations. Ces demandes concernent le contrôle de la musique (arrêt et démarrage) lorsqu'il est assis sur le canapé, Il souhaiterait en particulier que la musique s'arrête s'il se met à parler et reprenne lorsqu'il s'arrête. Ces demandes d'action (arrêt et redémarrage) surviennent lorsque le système est dans l'état $S1$ si l'utilisateur est allé directement s'asseoir après être entré, et $S5$ s'il s'est auparavant connecté au PC. La présence de plusieurs actions différentes de celles présentes indique la nécessité de redécouper ces deux situations, actuellement trop générales pour prendre en compte ces besoins.

9.5.3 Représentation d'une situation

Le nombre de rôles et de relations étant fixe, les situations peuvent être représentées sous forme d'un vecteur binaire de taille fixe :

$$(Rol_1, Rol_2, \dots, Rol_n, Rel_1(R_1, R_1), Rel_1(R_1, R_2), \dots, Rel_1(R_n, R_n), Rel_2(R_1, R_1), \dots)$$

Soit n le nombre de rôles. Si une relation Rel_i est symétrique, elle nécessite $\frac{n(n+1)}{2}$ positions dans le vecteur. Si elle n'est pas symétrique, elle nécessite n^2 positions.

9.5.4 Classes d'algorithmes d'apprentissage

Nous avons considéré deux classes d'algorithmes d'apprentissage :

- L'apprentissage de concepts : *Find-S* et *Candidate Elimination*
- L'apprentissage d'arbres de décision : ID3

Find-S ((Mitchell, 1997), chapitre 2.4) construit la représentation la plus spécifique de rôles et de relations afin d'expliquer les exemples d'apprentissage. L'idée générale est de partir d'un des exemples (situation + service) et de généraliser la description de la situation (en remplaçant un 0 ou un 1 par un $-$) afin de couvrir l'ensemble des autres exemples proposant le même service. Un exemple d'apprentissage est présenté Figure 9.4.

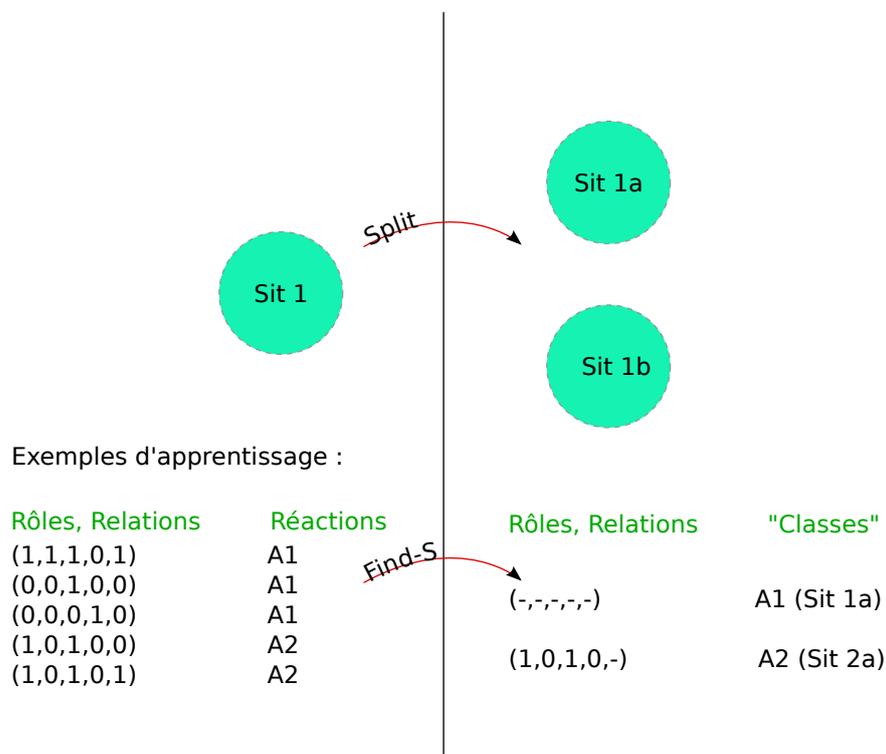


FIGURE 9.4: Exemple d'apprentissage par *find-S*

Les hypothèses produites étant les plus spécifiques, elles sont très sensibles au bruit (faible variation sur les rôles et les relations). Nous avons considéré une seconde famille d'algorithmes d'apprentissage conceptuel produisant une solution plus générale : *Candidate Elimination* ((Mitchell, 1997), chapitre 2.5). *Candidate Elimination* construit la représentation la plus générale, la représentation la plus spécifique et fait évoluer ces

représentations en spécialisant les premières et en généralisant les secondes jusqu'à ce qu'elles se rejoignent. La figure 9.5 reprend l'exemple précédent (Find-S) en appliquant cet algorithme. Les deux approches précédentes d'apprentissage conceptuel reposent sur

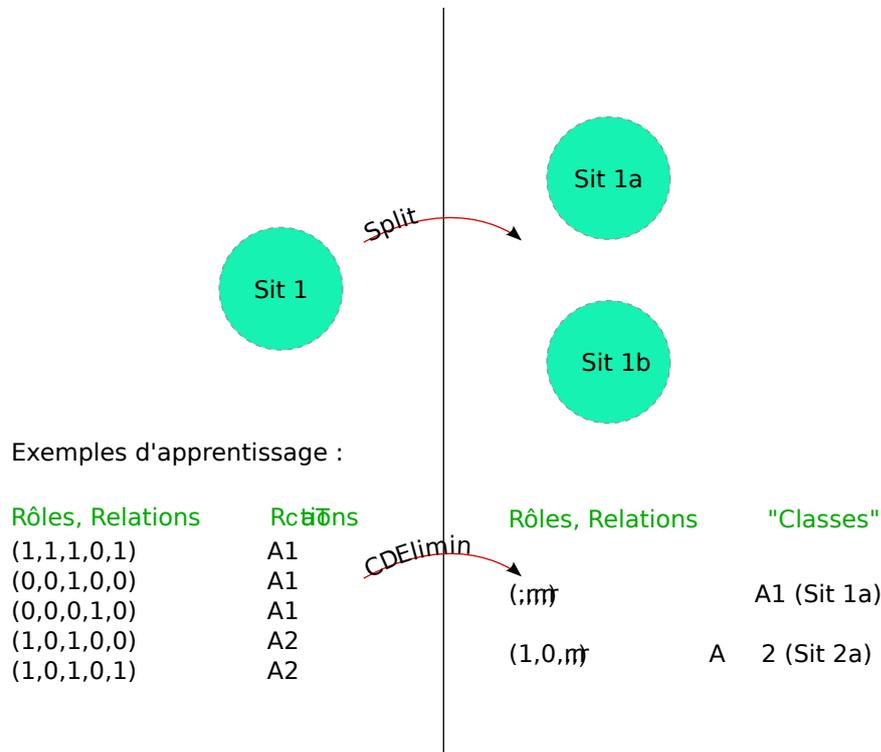


FIGURE 9.5: Exemple d'apprentissage par Candidate Elimination

un espace des hypothèses construit sur la conjonction des concepts (rôles et relations). Elles ne sont donc pas en mesure de représenter une sous-situation reposant sur une disjonction. Nous avons donc complété ces deux premières familles par une approche de type apprentissage d'arbres de décision : *ID3*. Les attributs de l'arbre sont les valeurs des rôles et des relations (0,1). Les feuilles correspondent aux classes (services). Le chemin de la racine vers une feuille fournit la représentation de la sous-situation associée à la feuille. Cette approche est illustrée sur l'exemple précédent Figure 9.6.

9.5.5 Résultat

Après apprentissage, les deux situations initiales *S1* et *S5* sont redécoupées en sous-situations. *S1* (une personne entre dans la pièce) inclut maintenant les sous-situations "assis sur le canapé", "parle en étant assis sur le canapé" et "se lève du canapé". Le

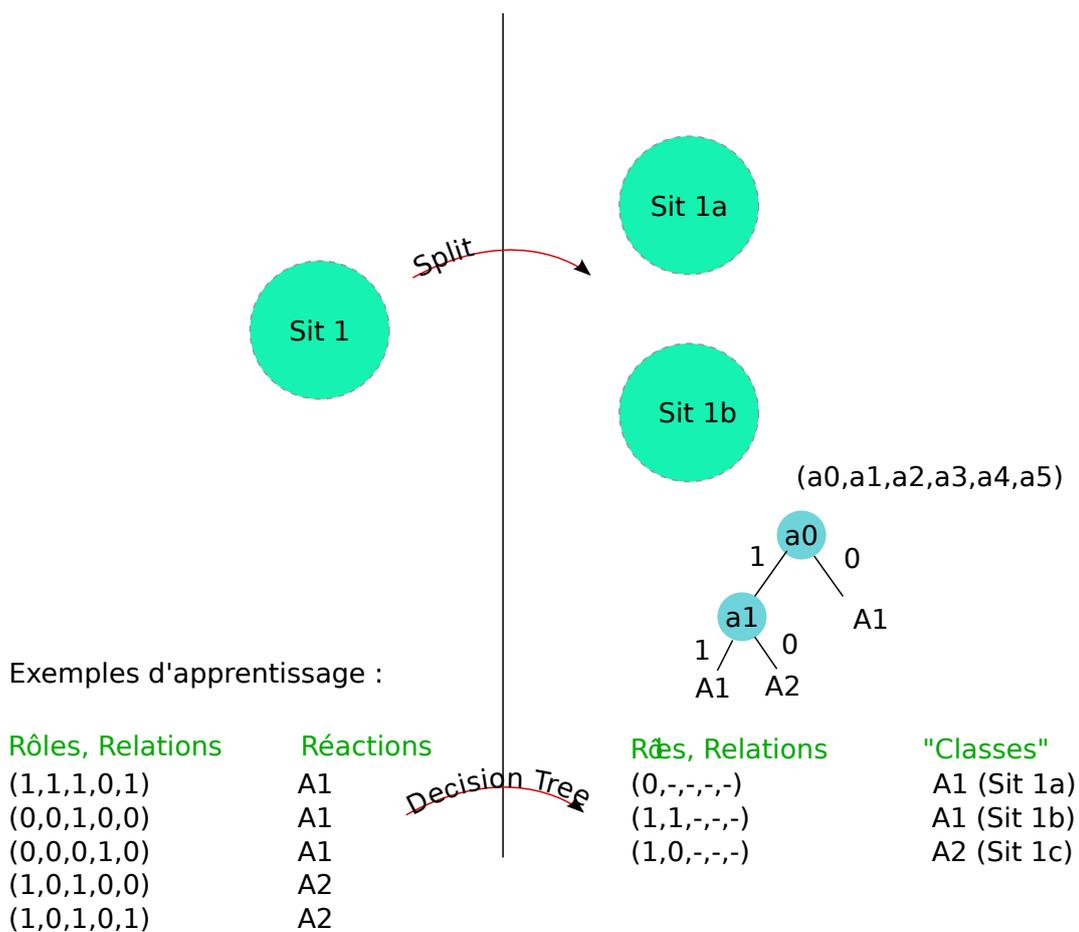


FIGURE 9.6: Exemple d'apprentissage par ID3

redécoupage est similaire pour *S5*. Les tables 9.1, 9.2 et 9.3 fournissent les matrices de confusion pour les trois familles d’algorithmes (*Find-S*, *Candidate Elimination* et *ID3*). *A0* correspond à l’action “ne rien faire”, *A8* démarre le lecteur audio et *A9* l’arrête.

	A0	A8	A9
A0	0.6014	0.1884	0.2101
A8	0.25	0.75	0
A9	0.25	0	0.75

TABLE 9.1: Matrice de confusion pour *Find-S*

	A0	A8	A9
A0	0.6232	0.1884	0.1884
A8	0.5	0.5	0
A9	0.625	0	0.375

TABLE 9.2: Matrice de confusion pour *Candidate Elimination*

	A0	A8	A9
A0	0.5985	0.1894	0.2121
A8	0	1	0
A9	0	0	1

TABLE 9.3: Matrice de confusion pour *ID3*

9.6 Bilan

Dans les exemples précédents, l’apprentissage supervisé a donc bien permis à l’utilisateur d’améliorer le comportement du système (modèle de contexte et actions associées) à l’aide d’exemples appropriés. Si, techniquement, cette approche est donc viable (tout au moins sur des exemples simples que l’on a considérés), elle n’est néanmoins pas réaliste. La collecte d’exemples est une tâche généralement fastidieuse. Autant il est envisageable de demander à un professionnel dont c’est le métier de prendre du temps pour collecter des exemples et les soumettre au système (voir chapitre 3), autant un utilisateur risque de rapidement abandonner le système.

Nous allons présenter au cours du chapitre suivant une approche permettant de recueillir les informations de l'utilisateur au cours de l'interaction : l'apprentissage renforcé.

L'apprentissage supervisé, tel que nous l'avons présenté au chapitre précédent, nécessite un retour après coup de la part de l'utilisateur pour annoter ce qui s'est passé. Cette annotation apporte une information précise quantitative (ce qu'il aurait fallu faire) mais "hors contexte" : on est plus en interaction. L'approche que nous allons présenter maintenant propose un retour "en situation" mais sous une forme qualitative : il s'agit de l'apprentissage renforcé. Cette étude est l'objet de la thèse de Sofia Zaidenberg.

10.1 Apprentissage renforcé

10.1.1 Principe

L'apprentissage renforcé (Sutton, 1988), (Sutton & Barto, 1998) est un modèle d'apprentissage où l'agent évolue dans un environnement et apprend de ses expériences à maximiser la quantité totale de récompenses reçues lors du choix de ses actions. L'information fournie par le superviseur (l'utilisateur final dans notre cas) n'est donc plus la réponse correcte mais une évaluation qualitative de la réponse proposée par l'agent : le renforcement. Le renforcement est généralement une fonction continue dont les valeurs sont comprises entre -1 et 1 . Une valeur de 1 indique la satisfaction, -1 la désapprobation et 0 l'absence d'avis.

Le problème de l'apprentissage renforcé peut se modéliser en s'appuyant sur les Processus de Décision Markoviens (MDP) (Bellman, 1957). Un MDP est défini par $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P} \rangle$:

- \mathcal{S} est l'ensemble des états de l'environnement.

- \mathcal{A} correspond à la liste des actions possibles.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ est la fonction de récompense. $\mathcal{R}(s, a, s')$ fournit un jugement sur l'opportunité d'avoir choisi l'action a dans la situation s .
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ est la fonction de transition modélisant l'environnement. $\mathcal{P}(s, a, s')$ est la probabilité d'être dans la situation s' en appliquant l'action a dans la situation s . Ce modèle d'environnement est stochastique. Il doit par contre être stationnaire.

Remarque : l'environnement est dit *stationnaire* si la distribution de probabilité modélisant la fonction de transition de l'environnement n'évolue pas au cours du temps.

Soit π une stratégie associée à l'agent : $\pi : \mathcal{S} \rightarrow \mathcal{A}$. Cette fonction propose pour chaque situation une action correspondante. Etant donné une stratégie π , on définit la fonction valeur d'un état associée à cette stratégie :

$$V^\pi(s) = E \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right\} \quad (10.1)$$

$V_\pi(s)$ représente le revenu espéré en partant de l'état s et en appliquant la stratégie π . L'objectif est de construire une politique optimale $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ qui maximise la fonction valeur $V^{\pi^*}(s)$ associée. La fonction valeur $V^{\pi^*}(s)$ est solution de l'équation de Bellman :

$$V^{\pi^*}(s) = \max_a \left(E \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right\} \right) \quad (10.2)$$

$$= \max_a \left(\sum_{s'} \mathcal{P}(s, a, s') (\mathcal{R}(s, a, s') + \gamma V^{\pi^*}(s')) \right) \quad (10.3)$$

L'équation 10.3 définit de manière récursive la fonction de valeur $V^{\pi^*}(\cdot)$. Une fois celle-ci calculée, la stratégie optimale associée peut être déterminée par :

$$\pi^*(s) = \operatorname{argmax}_a \left(\sum_{s'} \mathcal{P}(s, a, s') (\mathcal{R}(s, a, s') + \gamma V^{\pi^*}(s')) \right) \quad (10.4)$$

Si on dispose au départ d'un modèle complet de l'environnement (fonction de transition \mathcal{P} et de récompense \mathcal{R} du MDP), la stratégie optimale peut être trouvée en utilisant la programmation dynamique. Il s'agit alors plus d'une phase d'optimisation hors ligne, la stratégie pouvant être déterminée sans que l'agent interagisse avec son environnement.

Si on ne dispose par contre pas de modèles initiaux (transition et/ou récompense), on se retrouve alors dans le domaine de l'apprentissage renforcé. Il s'agit d'exploiter les conséquences des actions de l'agent sur son environnement afin de construire la stratégie optimale. Il existe plusieurs types d'approches :

Sans modèle (Model-Free) : l'apprentissage de la stratégie optimale est réalisé sans construire de modèle de l'environnement. Quelques exemples d'approches sans modèle :

- *Q-Learning* (Watkins & Dayan, 1992). Cette approche utilise les expériences successives de l'agent pour estimer l'utilité d'une action a dans un état donné s : $Q(s, a)$. La connaissance de cette fonction permet ensuite de choisir l'action pour chaque état. Il existe également des approches dérivées de Q-Learning telle que Sarsa (Rummery & Niranjan, 1994) (State, Action, Reward, State, Action).
- *Actor-Critic* : la fonction Q précédente contient à la fois la stratégie et la valeur de chaque état. Les approches de type Actor-Critic séparent ces deux aspects sous forme de deux fonctions séparées : la stratégie (*actor*) et la valeur (*critic*) estimant la stratégie courante. L'apprentissage repose sur les algorithmes $TD(\lambda)$ (Sutton, 1987).

Basées modèles (Model-Based) : ces approches essayent de reconstruire un modèle du système (phase apprentissage) afin de pouvoir réutiliser les résultats de la programmation dynamique pour construire la stratégie (phase planification).

10.1.2 Quelques exemples

L'apprentissage par renforcement a été utilisé en particulier en robotique (voir par exemple (Berns et al., 1992), (Millán & Torras, 1994), (Smart & Kaelbling, 2002)). Nous citerons plutôt les travaux autour des agents logiciels proactifs interagissant avec l'utilisateur. Robyn Kozierok et Pattie Maes (Kozierok & Maes, 1993) ont proposé dès 1993 un agent d'interface capable d'aider l'utilisateur à gérer son agenda (règles de prises de rendez-vous). Une des particularités de cette approche est de fournir le contrôle à l'utilisateur sur la délégation graduelle à l'agent des tâches relatives à l'agenda. Ceci permet de construire progressivement une relation de confiance de l'utilisateur envers l'agent, point important dans l'acceptation de ces approches (voir section 1.5). La montée en compétence de l'agent est basée sur une approche regroupant deux méthodes d'apprentissage : l'apprentissage par analogie[ⓐ] (Stanfill & Waltz, 1986) et l'apprentissage renforcé. Dans un premier temps, l'agent observe l'utilisateur en enregistrant tout ce qu'il fait en tant que couples situation-action. Dans une situation donnée, l'agent cherche la situation enregistrée la plus proche et peut ainsi prédire l'action à effectuer. L'apprentissage par renforcement intervient dans le cas où la prédiction est incorrecte. Cette approche est très intéressante car elle combine les propriétés que l'on souhaite dans une application de type Intelligence Ambiante : mise en place d'un lien de confiance avec l'utilisateur, intelligibilité du modèle choisi. En cas d'erreur de prédiction, le système explique à l'utilisateur les raisons de son choix et l'utilisateur peut expliquer pourquoi c'est incorrect.

ⓐ. Memory-Based Learning

Elle s'appuie par contre sur un modèle de "situation" très simple (initiateur d'un rendez-vous, participants et sujet) mais adapté au problème. La mise en place d'un modèle de situation générique risque d'amener une sollicitation trop importante de l'utilisateur et un rejet du système.

Walker et al. ont proposé ELVIS, un système de gestion d'emails par téléphone (Walker, 2000). L'agent doit apprendre à interagir vocalement de manière optimale avec l'utilisateur afin de lui fournir rapidement la bonne information. Par rapport à l'exemple précédent, l'espace d'état est complexe (13 variables discrètes pouvant prendre entre 2 et 4 valeurs chacune). L'apprentissage sur cet espace est long et complexe. Les auteurs préconisent de réduire cet espace d'état afin d'améliorer de manière significative les performances du système. La réduction est réalisée en recherchant les variables non pertinentes pour l'apprentissage.

10.1.3 Environnement partiellement observable

La modélisation d'un problème sous forme d'un MDP [Ⓟ] présuppose une perception complète de l'état de l'environnement. Si l'agent ne peut réaliser qu'une observation partielle, on remplace alors les MDP par des *Partially Observable Markov Decision Process* (POMDP) (Aström, 1965). Dans le cadre d'un POMDP, L'agent ne connaît pas l'état réel s dans lequel se trouve l'environnement, mais il possède seulement une observation o . Il agit alors en fonction d'un état estimé b (*belief state*) défini par une distribution de probabilités sur S (l'ensemble des états). Un POMDP est un processus non Markovien qui peut se ramener à un processus Markovien (MDP) sur un espace d'état continu, cet espace d'état continu étant l'espace des croyances (*belief space*). L'espace des croyances est l'espace qui, à partir de la probabilité d'être dans chacun des états et d'une action, donne la nouvelle probabilité d'être dans chacun des états (voir par exemple le tutorial (Cassandra, 1999)). Le MDP provenant du POMDP étant sur un espace continu, cela rend sa résolution beaucoup plus complexe.

Les POMDP sont notamment utilisés dans le cadre des systèmes multi-agents. Dans un système multi-agent, chaque agent accède à son état interne, à l'état de l'environnement mais n'a généralement pas accès à l'état interne des autres agents. Or, ces autres agents agissent également et sont donc partie prenante des modifications de l'environnement. Chaque agent n'a donc bien qu'une vue partielle de l'état du système complet (voir Littman (Littman, 1994) par exemple) Remarque : dans le cas d'un système multi-agents où chaque agent est modélisé sous forme d'un POMDP, on parle alors de DEC-POMDP (Decentralized Partially Observable Markov Decision Process).

[Ⓟ]. Markov Decision Process

10.2 Apprentissage renforcé indirect

10.2.1 Principe

L'apprentissage par renforcement est connu pour être lent. Leslie Kaelbling ([Kaelbling, 2004](#)) prend comme exemple le problème de l'ascenseur (trouver la trajectoire optimale) qui a nécessité 60000 heures de simulation, et celui du jeu Backgammon qui a appris sur plus de 1.5 millions de parties. Le principe de l'apprentissage renforcé est de laisser le système se construire en expérimentant son environnement. Lorsque cet environnement est virtuel (cf. partie de Backgammon), la parallélisation des algorithmes et l'accroissement de la vitesse des processeurs peut permettre dans une certaine mesure de réduire ces durées considérables. Lorsque l'environnement est réel et que, de plus, l'utilisateur en fait partie (en tant par exemple que source des valeurs de satisfaction), la vitesse du processeur n'est plus le facteur limitant dans le temps de convergence. Les nombreux pas d'apprentissage avant convergence vont impliquer de nombreux retours de l'utilisateur qui va rapidement se lasser. Il est donc important de prendre en compte cet aspect pour nos agents proactifs.

Le principe de l'apprentissage par renforcement indirect est d'ajouter aux expériences réelles des expériences virtuelles. Les interactions avec le monde réel ne suffisent pas pour apprendre la politique car elles sont trop rares. De plus, si l'environnement est complexe, alors l'espace d'états est vaste et certains états ne seront que très rarement visités. La politique risque donc d'être mauvaise pour ces états. L'apprentissage par renforcement indirect introduit un monde virtuel, un modèle du monde réel, dans lequel l'agent peut jouer autant d'expériences que nécessaire afin d'apprendre rapidement, sans impliquer l'utilisateur ni l'environnement réel (les méthodes classiques d'apprentissage par renforcement sont appliquées à ces expériences mentales). Autrement dit, le modèle du monde est utilisé pour générer des expériences imaginaires, telles que les a introduites Kenneth Craik en 1943 ([Craik, 1943](#)). Craik développe la proposition que l'Homme utilise des modèles mentaux, qu'il construit des "modèles réduits" de la réalité employés pour raisonner, anticiper les événements et servir de base aux explications.

La construction du modèle du monde est un processus continu et à long terme. C'est ce qui donne vie au système, ce qui lui permet de réagir aux changements du monde. La planification (apprentissage renforcé à partir des expériences mentales) est un processus secondaire rapide qui s'effectue en arrière plan.

L'approche DYNA a été proposée par Richard Sutton en 1991 ([Sutton, 1991](#)). Cette approche se décompose en trois parties, répétées en permanence :

1. Apprendre un modèle du monde.

2. Utiliser ce modèle pour déterminer une bonne politique (phase de planification : apprentissage par renforcement sur le modèle du monde).
3. Déterminer une bonne politique sans utiliser le modèle du monde (apprentissage par renforcement dans le monde réel).

Ces trois parties sont asynchrones. La partie 1 s'exécute en parallèle et indépendamment des deux autres. Les parties 2 et 3 s'exécutent en alternance (voir Figure 10.1).

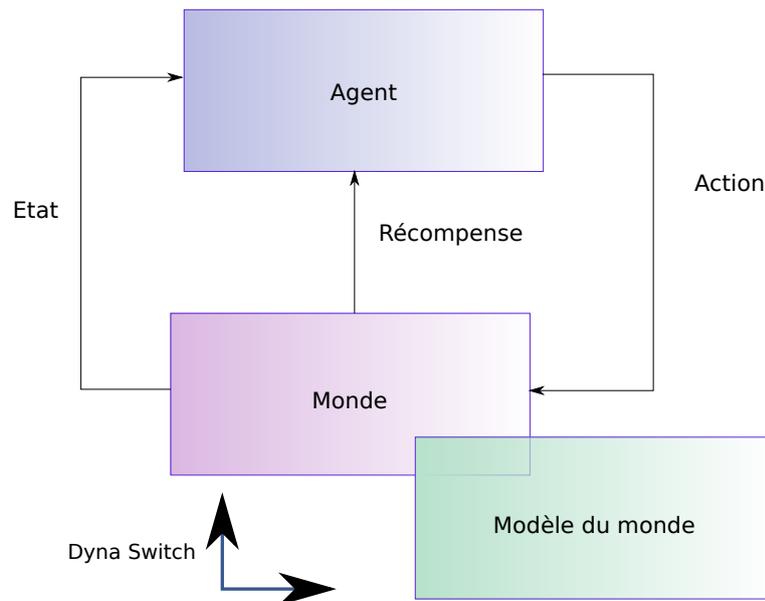


FIGURE 10.1: Méthode d'apprentissage renforcé indirect : DYNA. L'apprentissage s'effectue tour à tour avec le monde réel et le modèle du monde.

10.3 Notre approche

10.3.1 Choix du formalisme

La première question à laquelle nous devons répondre est : "quelle est la frontière de notre environnement?". Dans un système classique basé sur l'apprentissage par renforcement, seules les actions de l'agent provoquent une modification de l'environnement. Notre assistant personnel est immergé dans un environnement (physique et informatique) modifié par des éléments extérieurs sur lesquels il n'a aucun contrôle. L'utilisateur est en particulier un de ces éléments extérieurs. Ses allées et venues, les e-mails qu'il reçoit, son



activité courante, les rappels de son agenda, etc. sont autant d'éléments non déterministes provoquant des changements d'état. La question qui se pose alors est la suivante : l'utilisateur doit-il être intégré à l'environnement ou doit-il rester à l'extérieur ?

La réponse la plus naturelle est d'intégrer l'utilisateur à l'environnement : il en est un élément important. Ses actions sont fonction en partie de son état interne (cf. chapitre 1). L'état complet de l'environnement est donc inaccessible à l'agent (cf. POMDP). Comme l'explique Olivier Buffet dans sa thèse (Buffet, 2003), on est face à un problème non-Markovien stationnaire. Comme nous l'avons vu lors de la section 10.1.3, le problème non-Markovien peut se transformer en problème Markovien. Le problème est stationnaire car toute la non-stationnarité introduite par l'utilisateur est prise en compte dans la partie non-observable. L'environnement lui-même peut être considéré comme stationnaire.

L'autre approche consiste à considérer que l'utilisateur est extérieur à l'environnement. L'environnement peut alors être complètement observé par l'agent. On se retrouve alors dans un cadre de problème Markovien (MDP) non stationnaire. L'utilisateur est l'élément perturbateur qui crée la non stationnarité.

L'apprentissage dans le cadre des POMDP est reconnu comme un problème difficile (Jaulmes et al., 2005). Nous avons préféré rester dans le cadre mieux maîtrisé des MDP et donc adopter la seconde approche : le problème Markovien non stationnaire. La non stationnarité est aussi une difficulté pour l'apprentissage mais dans notre cas, nous sommes confrontés à une non-stationnarité que l'on peut supposer lente, cette non-stationnarité étant le résultat de l'évolution du comportement de l'utilisateur. L'hypothèse est que l'apprentissage renforcé converge avec une vitesse bien supérieure à la vitesse de la non-stationnarité afin que la politique apprise soit capable de suivre les évolutions de comportement de l'utilisateur.

10.3.2 Vue d'ensemble

Conformément à l'architecture Dyna, notre système est constitué d'une partie en ligne et d'une partie hors ligne [⊙], lancée régulièrement en tâche de fond.

La partie en ligne est à l'écoute des événements capteurs. L'arrivée d'un événement est interprétée comme un changement d'état de l'environnement. L'assistant choisit une action correspondante en se basant sur la politique courante. L'action est exécutée. L'utilisateur donne éventuellement un renforcement. Les transitions (état précédent, action et nouvel état) ainsi que le renforcement *sont sauvegardés* pour l'apprentissage supervisé du modèle du monde. On met donc en place une base de données contenant l'historique des interactions.

[⊙]. non interactive

La partie non interactive (tâche de fond) est composée de deux parties. La première consiste à jouer des épisodes de Q-Learning sur le modèle du monde au lieu de l'environnement : c'est la phase de planification. La seconde partie consiste à apprendre le modèle de transition du monde et le modèle de renforcement à partir des exemples collectés lors des interactions avec le monde réel. Ces différentes étapes sont résumées dans l'algorithme 2.

Algorithme 2 : L'algorithme global de l'assistant.

Input : Modèles de transition et de récompense initiaux, $n =$ nombre de pas entre deux mises à jour du modèle du monde

Output : Le modèle de contexte de l'utilisateur.

Exécuter un épisode (algorithme 5);

$i = 0$;

répéter

Recevoir le nouvel état s_i ;

Sauvegarder l'exemple suivant dans la base de données : $\{s_{i-1}, a_{i-1}, s_i\}$;

Choisir une action en utilisant la politique courante $a_i = \pi(s_i)$ et retourner l'action à l'assistant afin qu'il l'exécute dans l'environnement réel;

Afficher s_i et a_i pour l'utilisateur;

Si l'utilisateur donne une récompense, alors la sauvegarder dans la base de données : $\{s_i, a_i, r_i\}$;

si i est un multiple de n **alors**

 Exécuter l'apprentissage supervisé du modèle de transition (algorithme 3);

 Exécuter l'apprentissage supervisé du modèle de récompense (algorithme 4);

$i = i + 1$;

jusqu'à ∞ ;

En parallèle, à des intervalles de temps réguliers, exécuter un épisode (algorithme 5). La nouvelle politique remplace la politique courante pour le choix des actions dans le monde réel;

Première remarque : l'algorithme Dyna propose une mise à jour de la politique à la fois par la planification (utilisation du modèle du monde) mais également par l'utilisation du Q-learning lors des phases dans le monde réel. Ceci conduit à une modification permanente du comportement de l'assistant. Cette absence de stabilité peut être très déconcertante pour l'utilisateur final. Nous avons donc préféré ne faire évoluer la politique que de temps en temps, lors des phases de planification.

Seconde remarque : la première étape "exécuter un épisode" (non interactif) permet d'injecter des connaissances initiales. Cela permet de ne pas commencer avec une Q-table vide et donc un comportement d'assistant totalement aléatoire. Cette connaissance initiale prend la forme d'un modèle d'environnement et d'un modèle de récompense. Le

modèle de récompense (qui sera précisé section 10.3.5) fournit le renforcement attendu en fonction du choix de l'action dans une situation donnée. Ce modèle de renforcement peut donc être pré-construit à partir du modèle de contexte proposé par le développeur. Cette première étape "exécuter un épisode" permet donc de transcrire un modèle de contexte initial en une représentation de type Q-table adaptée à l'apprentissage par renforcement.

L'algorithme principal est mis en place. Il faut maintenant choisir une représentation d'un état de l'environnement permettant de construire la Q-table. Il faut également s'intéresser aux récompenses proposées par l'utilisateur. Ces deux derniers éléments permettent de mettre en place l'apprentissage renforcé. Pour l'apprentissage renforcé indirect, il faut également choisir un modèle de représentation de l'environnement simulé et mettre en place les algorithmes permettant de le construire. Le monde simulé étant prêt, il ne reste plus qu'à l'exploiter par la phase de planification. Nous allons maintenant aborder tous ces points dans les sections suivantes.

10.3.3 Etat

Modélisation d'un état

Comme nous l'avons vu précédemment, il est important de proposer une représentation intelligible afin de gagner la confiance de l'utilisateur. Nous avons choisi une représentation des états sous forme de prédicats du premier ordre. Ces prédicats et leurs arguments sont directement choisis en fonction des capteurs dont on dispose. La liste est fournie table 10.1.

La valeur des variables peut être le symbole particulier `<null>` afin d'indiquer qu'aucune valeur n'est disponible. Un état du système est constitué par la conjonction de l'ensemble de ces prédicats. Chaque prédicat est présent une et une seule fois. Un prédicat dont tous les arguments sont `<null>` signifie que ce prédicat est faux dans l'état courant. Exemple :

```
alarm(minute=<null>, title=<null>, hour=<null>)
```

signifie qu'aucune alarme de rappel d'événement n'est actuellement programmée.

Réduction de l'espace d'états

Etant donné le nombre de prédicats et le nombre de valeurs possibles des variables associées, la dimension de notre espace d'état, et donc de la Q-table correspondante, est

Prédicat	Arguments	Description
alarm	title, hour, minute	Le dernier rappel émis par l'agenda de l'utilisateur ; Lorsqu'il n'y a pas de rappel, les arguments ont des valeurs nulles.
xActivity	machine, isActive	Indique s'il y a une activité clavier/souris sur un ordinateur donné.
inOffice	user, office	Indique, s'il est connu, le bureau dans lequel se trouve l'utilisateur, null sinon.
absent	user	Déclare que l'utilisateur n'est pas dans son bureau. Il pourrait être dans un autre bureau ou bien le système ne sait pas où il se trouve.
hasUnreadMail	from, to, subject, body	Le dernier mail reçu par l'utilisateur.
entrance	isAlone, friendlyName, btAddress	Indique que quelqu'un vient d'entrer dans le bureau de l'utilisateur. Cette personne est identifiée grâce à son dispositif bluetooth. L'argument <i>isAlone</i> indique si l'utilisateur était seul dans son bureau avant cet événement.
exit	isAlone, friendlyName, btAddress	De la même manière que précédemment, indique que quelqu'un vient de quitter le bureau de l'utilisateur.
task	taskName	La tâche sur laquelle l'utilisateur est entrain de travailler.
user	login	Ce prédicat n'est pas sensé être modifié, il désigne l'utilisateur principal de l'assistant.
userOffice	office, login	De même, ce prédicat identifie le bureau de l'utilisateur principal.
userMachine	machine, login	De même, ce prédicat identifie l'ordinateur de l'utilisateur principal.
computerState	machine, isScreenLocked, isMusicPaused	Décrit l'état de l'ordinateur de l'utilisateur en indiquant si son écran est verrouillé ou non et si la musique est en pause ou non.

TABLE 10.1: Prédicats de description de l'environnement.

beaucoup trop importante pour une mise en oeuvre réaliste. Les problèmes proviennent d'une part de la quantité de mémoire nécessaire mais surtout du nombre d'essais qu'il va falloir réaliser pour pouvoir peupler cette Q-table de manière satisfaisante. Il va donc falloir réduire (factoriser) cet espace d'état. On peut par exemple se demander s'il est vraiment nécessaire de garder deux états distincts selon que c'est Mr Dupont ou Mr

Durant^⑥ qui entrent dans le bureau : l'action à exécuter a de fortes chances d'être la même dans les deux cas. Il est donc judicieux de généraliser ces deux états en un seul "quelqu'un entre dans le bureau". Lorsque l'on observe pour la première fois un visiteur, le système apprend une estimation du comportement à avoir dans le cas où n'importe qui entre dans le bureau. Ceci est plus efficace car il ne faut pas attendre de voir chaque personne pour agir de manière cohérente.

Un état généralisé est obtenu en remplaçant les valeurs des variables fournies par les capteurs par deux jokers : $\langle + \rangle$ et $\langle * \rangle$. Le joker $\langle + \rangle$ remplace une valeur non nulle. Le joker $\langle * \rangle$ remplace une valeur nulle.

La Q-table ne contient que des états généralisés, ce qui réduit considérablement ses dimensions. Cette idée est similaire à celle des systèmes de classeurs décrits par Sigaud et Gérard (Sigaud & Gérard, 2005). Dans ces systèmes, ce ne sont pas les couples état-action qui sont associés dans la Q-table, mais les couples condition-action, une condition étant un ensemble de contraintes pouvant être vérifiées par plusieurs états, ce qui permet la généralisation.

Remarque : la généralisation des états permet de réduire l'espace d'état et d'accélérer l'apprentissage. Elle peut par contre être parfois excessive. Dans certains cas, il peut être pertinent de distinguer les valeurs concrètes des arguments pour choisir l'action. Par exemple, l'utilisateur pourrait vouloir être averti d'un e-mail si l'expéditeur est son supérieur hiérarchique (from=directeur@entreprise.com), mais pas s'il s'agit d'un bulletin (from=newsletter@lemonde.fr). Il est donc nécessaire de prévoir un mécanisme de spécialisation d'états génériques. La Q-table contient alors des états génériques et des états partiellement ou totalement spécifiés (dont les variables possèdent des valeurs autre que $\langle + \rangle$ ou $\langle * \rangle$). Lors de la recherche d'une action à exécuter, les états spécifiques sont mis en correspondance en priorité avec l'état perçu. S'il n'y a pas de correspondance, alors les états génériques sont considérés. La question qui se pose alors est : "quand est ce qu'un état générique doit être spécialisé?". Nous n'avons pas mis en place de solution pour cette question. Une des pistes envisagée est la détection "d'aliasing perceptif" en se basant sur l'historique sauvegardé. Si, pour un état donné, les renforcements reçus oscillent (tantôt positifs, tantôt négatifs par exemple), cela signifie que l'état perçu par l'agent est trop général pour percevoir les détails qui plaisent ou déplaisent à l'utilisateur. Il faut mettre en place une description plus fine et donc, spécialiser l'état.

10.3.4 Collecte du renforcement

L'apprentissage se base sur les renforcements que l'utilisateur fournit pour les actions que l'assistant décide d'exécuter. L'apprentissage est d'autant meilleur (rapide et

⑥. dans la mesure où Mr Dupont et Mr Durant ont un statut social équivalent

précis) que les renforcements reçus sont fréquents et fiables. Malheureusement, comme le notent Richard et Yamada (Richard & Yamada, 2007), les utilisateurs sont souvent réfractaires à fournir des informations précises ou une récompense explicite à un système d'apprentissage. Cette tâche est fastidieuse et ils sont souvent trop occupés pour y penser. De plus, comme le souligne Isbell et al, (Isbell et al., 2001), les récompenses fournies par l'utilisateur peuvent souvent être incohérentes et se décaler dans le temps :

”Individual users may be inconsistent in the rewards they provide (even when they implicitly have a fixed set of preferences), and their preferences may change over time (for example, due to becoming bored or irritated with an action). Even when their rewards are consistent, there can be great temporal variation in their reward pattern.”

Thomaz et al, (Thomaz et al., 2006) ont spécifiquement étudiés l'apprentissage par renforcement guidé par des utilisateurs humains non experts. La conclusion de cette étude est que, pour les humains, l'entraînement est un processus à double sens. Les humains ont tendance à vouloir communiquer avec le système, à voir l'entraînement comme un enseignement, un partenariat. En particulier, les usagers sont enclin à encourager ”l'élève” en lui donnant un renforcement positif, même si l'action choisie est incorrecte. Les participants ont également montré une tendance à considérer les renforcements plutôt comme une guidance, une indication sur le comportement à avoir dans le futur et non pas une appréciation de la dernière action effectuée. Les utilisateurs attendaient également une amélioration immédiate du comportement suite à un renforcement négatif. Dans la pratique, la modification du comportement n'est pas immédiate, laissant un sentiment à l'usager que ses remarques sont ignorées.

Les conclusions de cette étude sont très intéressantes mais nous n'avons actuellement pas de piste pour les prendre en compte. Nous avons retenu les contraintes suivantes : le renforcement fourni par l'utilisateur est rare et il peut être incohérent. Ces contraintes nous ont amené à considérer également le renforcement implicite. Le renforcement implicite consiste à recueillir des indices permettant de déduire la satisfaction ou la non satisfaction de l'utilisateur. Par exemple, si l'assistant ouvre une fenêtre de réponse à un email et que l'usager l'utilise, on peut considérer qu'il est satisfait. Si par contre, il la referme ou l'ignore, l'action était inappropriée. Les renforcements implicites, de part leur caractère incertain, ne doivent pas avoir un impact trop important ((Richard & Yamada, 2007) et (Maes, 1994)).

10.3.5 Modèle de l'environnement

L'apprentissage renforcé indirect se base sur un modèle de l'environnement afin de réaliser des ”expériences mentales”. Ce modèle doit être en mesure de fournir à partir de l'état courant, l'état suivant ainsi que le renforcement reçu en fonction d'une action ou

d'un événement (voir figure 10.2). Leslie Kaelbling (Kaelbling, 2004) précise qu'un bon

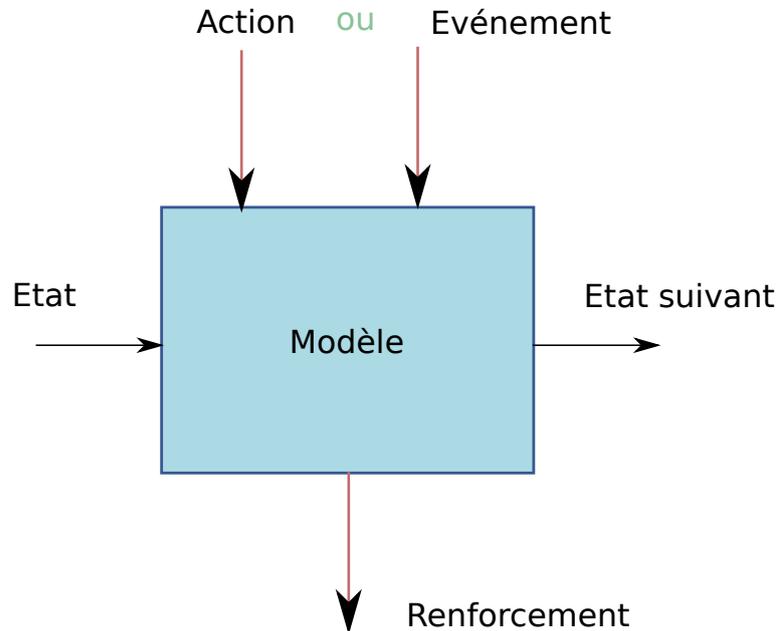


FIGURE 10.2: Modèle de l'environnement

modèle de monde doit être compact, pratique pour la planification et facile à apprendre. Elle suggère en particulier l'utilisation de règles probabilistes, formalisme bien adapté à la représentation d'objets dont les propriétés changent au cours du temps et à la modélisation d'environnements dont les états ne sont que partiellement modifiés après chaque action.

Notre modèle d'environnement est constitué de deux parties : une partie dédiée au calcul de l'état suivant (*modèle de transition* \mathcal{P}) et une partie dédiée au calcul du renforcement (*modèle de récompense* \mathcal{R}). Ces deux parties sont apprises par apprentissage supervisé à partir des exemples sauvegardés dans l'historique lors des interactions de l'utilisateur avec l'environnement (voir section 10.3.6).

Fonction de transition

La fonction de transition $\mathcal{P} : \mathcal{S} \times \mathcal{O} \times \mathcal{S} \rightarrow [0, 1]$ est une fonction stochastique Markovienne notée $\mathcal{P}(s'|s, o)$, o étant soit une action, soit un événement. Elle fournit la probabilité d'atteindre l'état s' à partir de l'état s étant donné l'action ou l'événement o .

Cette fonction de transition est définie comme un ensemble de transformations. Chaque transformation est composée de :

- un état de départ s^t . Cet état peut être générique (valeur $\langle * \rangle$ ou $\langle + \rangle$ pour les variables).
- une occurrence o^t ;
- une probabilité p^t (la probabilité que cette transformation s’applique) ;
- les modifications à appliquer sur l’état de départ observé afin d’obtenir l’état d’arrivée M^t . Ces modifications ont pour rôle de fixer la valeur des variables des prédicats de l’état suivant. Ces valeurs peuvent être soit des constantes, soit provenir des variables de l’état initial, de l’événement ou de l’action.

Exemple : une transformation peut par exemple spécifier que, quel que soit l’état initial, un événement `entrance(user)` a pour effet de positionner la valeur de la variable `user` du prédicat `inOffice(user,office)` à la valeur de la variable `user` de l’événement `entrance`.

Lorsque le modèle remplace le monde réel dans l’apprentissage, le modèle de transition reçoit un état s et une occurrence o et doit retourner l’état suivant s' . Il recherche pour ce faire une transformation compatible avec les paramètres en entrée (s et o) et calcule le nouvel état en sortie s' en appliquant les modifications de cette transformation. Si plusieurs transformations sont applicables, on tire au sort la transformation retenue en respectant leurs probabilités associées. Si aucune transformation n’est applicable, on est alors dans une zone non modélisée du monde. On est sorti de la compétence du modèle.

Fonction de récompense

La fonction de récompense est une fonction $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathfrak{R}$. Elle fournit une valeur de récompense (nombre réel) en fonction du choix de l’action a dans une situation s . Cette fonction est définie par extension à l’aide d’un ensemble de triplets (s, a, r) . s peut être un *état générique*. Lorsque le modèle remplace le monde réel dans l’apprentissage, le modèle de récompense reçoit un état s et une action a et doit retourner le renforcement associé. Si plusieurs triplets correspondent à la situation s , le renforcement retourné est la moyenne des renforcements proposés par chacun d’entre eux.

Nous allons maintenant détailler la construction de ces deux fonctions par apprentissage supervisé sur l’historique de l’interaction de l’usager.

10.3.6 Apprentissage du modèle de l’environnement

L’environnement est trop complexe pour être modélisé manuellement. De plus, il évolue au cours du temps. Le modèle de l’environnement est donc appris par apprentissage

supervisé. Cet apprentissage est réalisé sur la base d'exemples stockés dans l'historique. Il est régulièrement ré-exécuté : plus l'utilisateur interagit, plus l'historique grandit et plus il y a d'exemples pour affiner le modèle. Le réapprentissage régulier permet également de s'adapter aux évolutions de l'environnement.

L'évolution possible de l'environnement nécessite que l'apprentissage soit en mesure d'intégrer progressivement de nouveaux exemples venant contredire le modèle courant. Cette intégration doit être progressive afin d'éviter une déstabilisation brutale du modèle courant par ajout d'un exemple incorrect. Cet exemple incorrect peut provenir d'une mesure capteur fautive dans le cadre du modèle de transition, et d'une récompense utilisateur inappropriée dans le cadre du modèle de récompense (voir section 10.3.4).

Apprentissage de la fonction de transition

Cet apprentissage s'effectue sur les exemples de transitions enregistrés durant les interactions réelles. Ces transitions sont de la forme $\{s, o, s'\}$, où o est une occurrence pouvant être une action a ou bien un événement e . Les états s et s' sont des états ayant été perçus durant les interactions. L'algorithme d'apprentissage est l'algorithme 3.

Algorithme 3 : L'algorithme d'apprentissage supervisé du modèle de transition.

Input : Un ensemble d'exemples $\{s, o, s'\}$

Output : \mathcal{P}

pour chaque *exemple* $\{s, o, s'\}$ **faire**

si une transformation t permettant d'obtenir s' à partir de s avec l'occurrence o , peut être trouvée **alors**

 └ Augmenter la probabilité de t ;

sinon

 Créer une transformation, partant de s , ayant l'action a , ou bien un événement générique créé à partir de e , et se terminant dans s' , avec une faible probabilité;

 Diminuer la probabilité de toute autre transformation t' qui correspond à l'état de départ s et à l'occurrence o mais dont l'état d'arrivée est différent de s' ;

Apprentissage de la fonction de récompense

Le modèle des récompenses est appris sur les récompenses que l'utilisateur a données lors d'interactions réelles du système avec l'environnement. Durant ces interactions

Algorithme 4 : L'algorithme d'apprentissage supervisé du modèle de renforcement.

Input : Un ensemble d'exemples $\{s, a, r\}$

Output : \mathcal{R}

pour chaque *exemple* $\{s, a, r\}$ **faire**

si une entrée du modèle $e = \{s_e, a_e, r_e\}$ telle que $s \sim s_e$ et $a = a_e$ peut être trouvée **alors**

 Mettre à jour e , assigner $r_e = \text{mix}(r, r_e)$, où mix est une fonction de combinaison;

sinon

 Ajouter une nouvelle entrée $e = \{s, a, r\}$ au modèle de récompense;

l'assistant enregistre tous les exemples de la forme $\{s, a, r\}$ et l'algorithme 4 permet d'apprendre de manière supervisée un modèle à partir de ces exemples.

10.3.7 Apprentissage non interactif (planification)

La phase de planification se base sur le modèle de l'environnement dans l'état actuel afin de mettre à jour la politique par utilisation de l'apprentissage renforcé. L'algorithme 5 résume les différentes étapes.

Chaque épisode débute dans un état initial. Cet état initial peut être choisi de trois manières : un état initial "neutre" dont toutes les variables valent `<null>`, un état généré aléatoirement, ou enfin un état tiré au hasard parmi les états enregistrés dans la base de données (historique), c'est-à-dire un état déjà observé. La seconde option permet d'augmenter l'exploration de l'espace d'états et d'essayer de passer par des états inconnus afin d'acquérir une estimation du comportement à avoir lorsqu'ils seront rencontrés réellement. Ceci a un sens car notre modèle de transition a une capacité de généralisation (le modèle appris est plus général que les exemples d'interactions réelles enregistrés et sur lesquels il a été appris). La troisième option permet de renforcer l'expérience vécue.

Un épisode se termine classiquement au bout de k itérations, ou bien lorsqu'un état final a été atteint. Dans notre cas, il n'y a pas d'états finaux car l'assistant n'a pas de but spécifique à remplir, seulement la maximisation des récompenses. Il est tout de même possible de trouver un autre critère d'arrêt prématuré d'un épisode basé sur la mise à jour de la Q-table. Lorsque la différence des Q-valeurs entre deux pas n'est plus suffisamment significative, nous pouvons arrêter l'épisode en cours.

Algorithme 5 : Un épisode de Q-learning utilisé pour l'étape de planification par l'agent d'AR.

Input : \mathcal{P} , \mathcal{R} , k le nombre d'itérations dans un épisode

pour i de 1 à k **faire**

 Choisir un événement e ;

 Envoyer l'état courant s et l'événement e au modèle de l'environnement et obtenir une prédiction sur l'état suivant s' : $s' \leftarrow \max_{s' \in \mathcal{S}} \mathcal{P}(s'|s, e)$;

si s' ne peut être fourni (zone inconnue du modèle) **alors**

 └ Interrompre l'épisode

sinon

 Choisir une action $a = \pi(s')$;

 Envoyer s' et a au modèle de l'environnement et obtenir des prédictions sur l'état suivant s'' et la récompense r :

$s'' \leftarrow \max_{s'' \in \mathcal{S}} \mathcal{P}(s''|s', a)$, $r \leftarrow \mathcal{R}(s', a)$;

 Appliquer une méthode d'apprentissage par renforcement à l'expérience hypothétique $\langle s', s'', a, r \rangle$:

$Q(s', a) \leftarrow Q(s', a) + \alpha(r + \gamma \max_{a'} Q(s'', a') - Q(s', a))$;

$i = i + 1$;

10.3.8 Enquête grand public

Parallèlement à la mise en place de cette approche basée sur l'apprentissage renforcé, nous avons réalisé une enquête "grand public" sous la direction de Nadine Mandran du LIG afin de mieux cerner les attentes des utilisateurs potentiels. Cette enquête a été réalisée auprès de 26 sujets non-informaticiens. La répartition des sujets en groupe d'âge et de sexe est fournie dans le tableau 10.2. Les sujets ont également été équilibrés selon les critères de type lieux d'habitation, nombre d'enfants et type d'habitat. L'enquête s'est

Classe d'âge \ Sexe	Sexe		Total
	Femme	Homme	
18-25	3	6	9
26-40	4	3	7
40-60	4	3	7
60+	1	2	3
Total	12	14	26

TABLE 10.2: Répartition des sujets de l'enquête selon l'âge et le sexe.

déroulée sous forme d'entretiens d'une heure environ. Les sujets étaient confrontés à une maquette du système sous forme d'un powerpoint interactif.

Parmi les réponses, on peut noter le souhait des usagers de s'impliquer dans l'apprentissage de leur agent si cela est gage de fiabilité. Ils sont prêts à accepter ses erreurs et à le corriger si cela peut conduire à une amélioration de son comportement. Cette amélioration doit par contre arriver relativement rapidement (de l'ordre de 2 à 3 semaines) :

« Je laisserai un délai mais pas énorme. Si au début il se trompe, pourquoi pas, je me prêterais au jeu »

Un apprentissage progressif est également préféré à une phase initiale longue pouvant être rapidement fastidieuse. Le choix d'un retour simple de l'utilisateur (de type renforcement) a également été confirmé. Les utilisateurs ont également indiqué qu'ils apprécient que le système soit en mesure d'expliquer ce qu'il fait : ils veulent garder le contrôle.

10.3.9 Validation Expérimentale

Une première évaluation de l'assistant a été réalisée. Elle inclut l'utilisation d'un simulateur permettant de rejouer plusieurs fois les mêmes scénarios dans des conditions similaires. Une évaluation "en vraie grandeur" va être réalisée cette année.

Dans un système d'apprentissage par renforcement, le résultat de l'apprentissage est une politique π qui, en se basant sur une représentation de la fonction de valeur d'action, donne un comportement au système. Notre premier critère d'évaluation est donc le comportement obtenu, et, plus précisément, à quel point celui-ci se rapproche du comportement désiré. On s'intéresse également au temps nécessaire pour que le système atteigne ce comportement.

Nous allons, pour réaliser cette évaluation, associer une note à une Q-table. Le critère principal pour cette note est le nombre d'états corrects atteints : $nbCorrect$. Il s'agit du nombre de cases de la Q-table pour lesquelles le comportement proposé est le comportement souhaité. Ceci est le critère principal d'évaluation. Le nombre total de cases visitées $nbTotal$ dans la Q-table caractérise l'exploration de l'espace d'état qui a été réalisée. Enfin, on s'intéresse au pourcentage de Q-valeurs non nulles $pNonNul$. Ce critère reflète le nombre d'états pour lesquels l'assistant a une proposition d'action non aléatoire. La note retenue est donc :

$$note = \frac{1}{13}(10 \times nbCorrect + 2 \times pNonNul + pTotal) \quad (10.5)$$

Apprentissage initial

Cette évaluation a pour objectif de mesurer l'influence des différents paramètres de la phase initiale. Cette phase a pour but de transférer les connaissances d'un modèle d'environnement vers une première Q-table. La figure 10.3 montre l'influence du choix de l'état initial pour un épisode. L'état initial aléatoire est une solution potentiellement in-

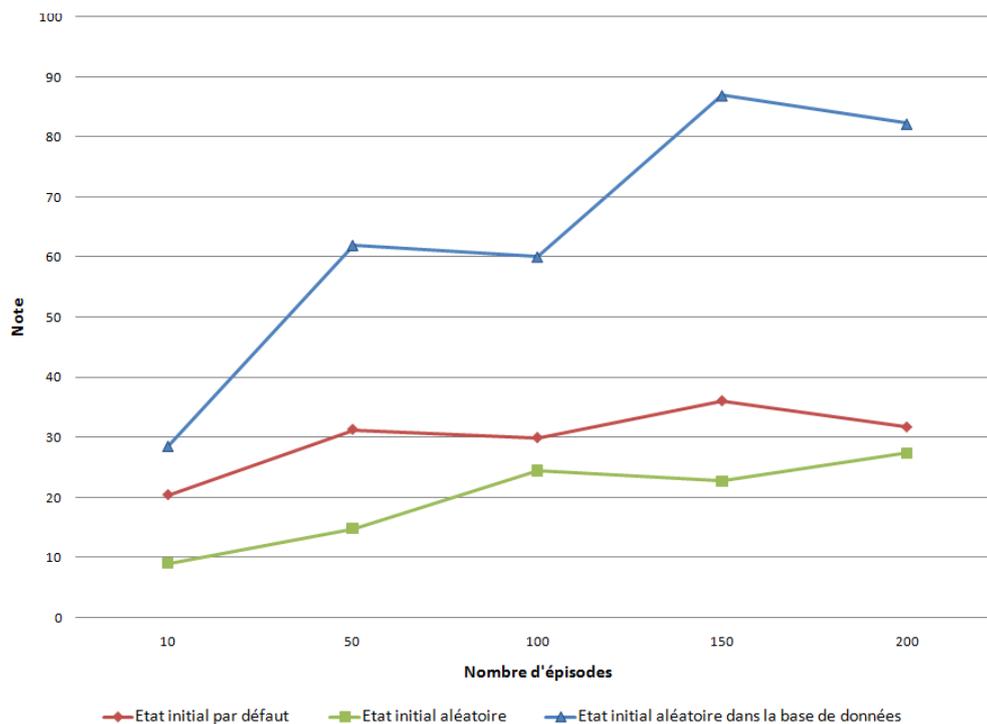


FIGURE 10.3: Influence du choix de l'état initial avec 100 itérations par épisode et des événements tirés au hasard dans la base de données

téressante car elle permet une plus grande exploration. Elle donne néanmoins de mauvais résultats pour un apprentissage initial. Ceci est dû au fait que le modèle initial d'environnement est minimal. L'état aléatoire a donc de fortes chances d'être dans une zone non prise en charge ce qui arrête l'épisode d'apprentissage.

La figure 10.4 étudie l'influence du nombre d'itérations par épisode. Sans surprise, le résultat est d'autant meilleur que le nombre d'itérations est important (meilleure exploration de l'espace d'états). Il faut néanmoins noter que l'amélioration de la note n'est pas proportionnelle au nombre d'itérations ou au nombre d'épisodes. Par exemple, les configurations suivantes permettent toutes d'obtenir la note 20 :

- 22 épisodes de 10 itérations, soit 220 itérations au total ;
- 14 épisodes de 25 itérations, soit 350 itérations au total ;

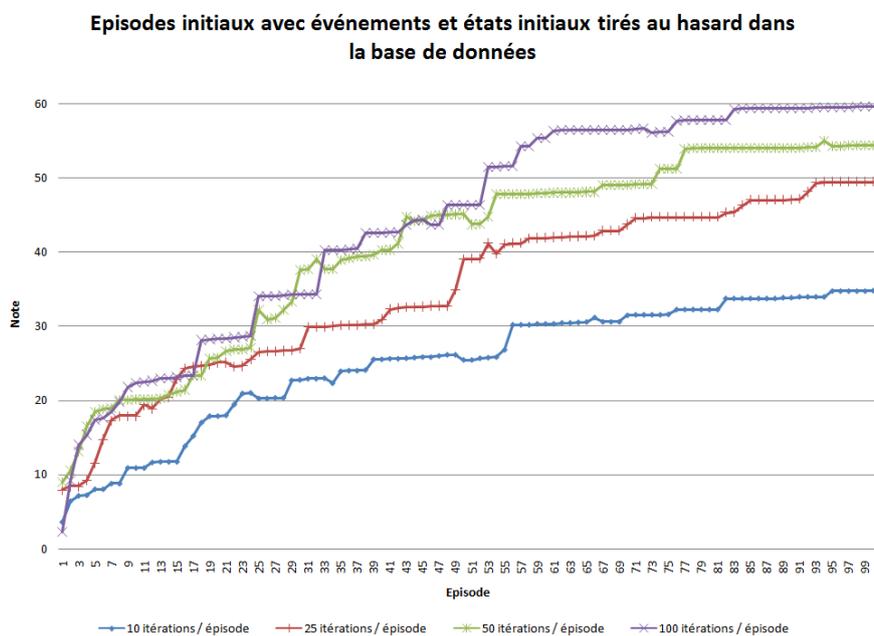


FIGURE 10.4: Influence du nombre d'itérations par épisode avec état initial et événements tirés au hasard dans la base de données.

- 9 épisodes de 50 itérations, soit 450 itérations au total ;
- 9 épisodes de 100 itérations, soit 900 itérations au total ;

Il peut donc être plus intéressant, afin de réduire le coût de calcul, de réduire le nombre d'itérations par épisode et d'augmenter le nombre d'épisodes.

Système complet

Il s'agit d'évaluer le système complet, incluant l'apprentissage supervisé périodique d'un modèle de l'environnement, des phases périodiques de planification pour exploiter ce modèle et l'utilisation de la politique issue de la planification. De manière à accélérer l'expérience, les événements pouvaient être déclenchés "manuellement" par l'utilisateur. L'apprentissage supervisé est mis en oeuvre tous les 6 événements reçus (ce qui correspond donc à 6 nouveaux exemples dans l'historique). Les épisodes d'apprentissage renforcé étaient déclenchés toutes les minutes. Ces fréquences sont bien entendu trop élevées pour un système "hors test". Il faut plutôt envisager une fréquence journalière pour la remise à jour des modèles et la planification.

La figure 10.5 montre l'évolution des notes des Q-tables en fonction du nombre d'épisodes. Le comportement appris est de plus en plus adéquat aux préférences de l'utilisateur. Les deux zones dans lesquelles la courbe est fortement croissante (correspondant approximativement aux intervalles $periode \in [0, 20]$ et $periode \in [70, 110]$) correspondent aux périodes où l'expérimentateur générait de nouveaux événements et donnait des renforcements, fournissant ainsi à l'assistant beaucoup de matériel à intégrer dans la Q-table. La zone dans laquelle la courbe est plate (pour les abscisses approximatives $x \in [20, 70]$) correspond à un moment où l'expérimentateur s'est absenté et où il n'y avait donc aucun événement ni renforcement. Ces épisodes n'ont pas fait évoluer l'apprentissage car la Q-table avait alors intégré toutes les informations disponibles à cet instant.

La courbe montrée Figure 10.6 représente approximativement la même expérience. L'expérience démarre après la phase d'apprentissage initial, et intègre tous les nouveaux événements fournis par l'expérimentateur. Aux alentours de l'épisode 90, l'expérimentateur a décidé de *changer d'avis* en donnant désormais un renforcement contradictoire à la logique suivie précédemment. On constate sur la courbe que la note diminue d'une manière significative (la Figure 10.6 montre un agrandissement de cette zone d'intérêt). Cette baisse correspond aux états dans lesquels la meilleure action ne correspond plus à l'action désirée par l'expérimentateur, alors qu'elle correspondait jusqu'à présent. Puis, la note remonte progressivement : les changements sont appris et intégrés d'abord au modèle de renforcement, puis au comportement.

Ces expériences préliminaires ont permis de valider le principe de l'approche par renforcement indirect pour l'apprentissage d'un assistant. La collecte des récompenses est un point qu'il faut améliorer avant de pouvoir réaliser un essai "grandeur nature"

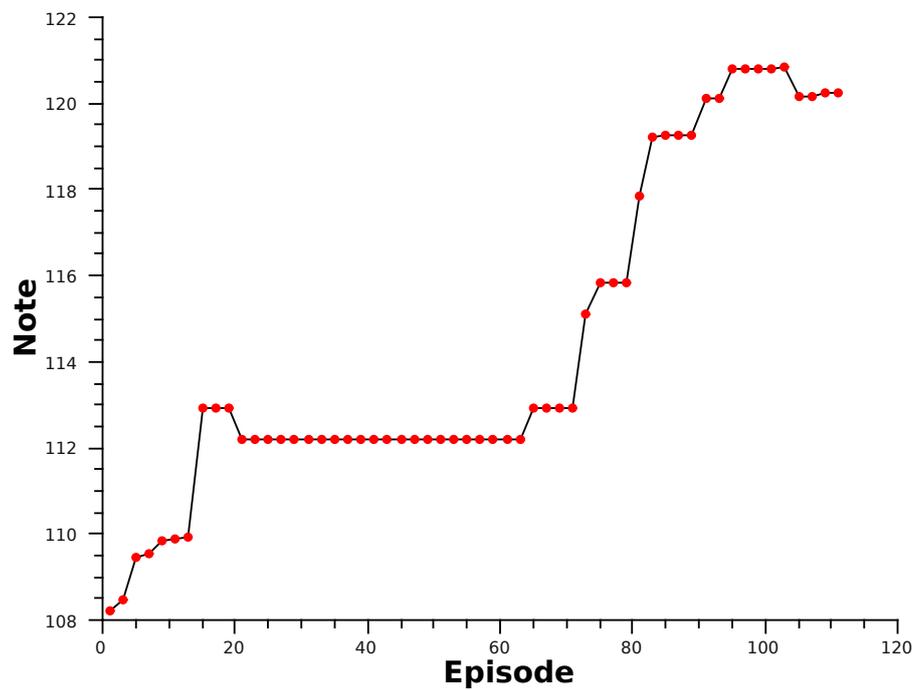


FIGURE 10.5: Notes des Q-tables produites par chaque épisode d'apprentissage par renforcement exécuté en parallèle d'interactions avec l'assistant.

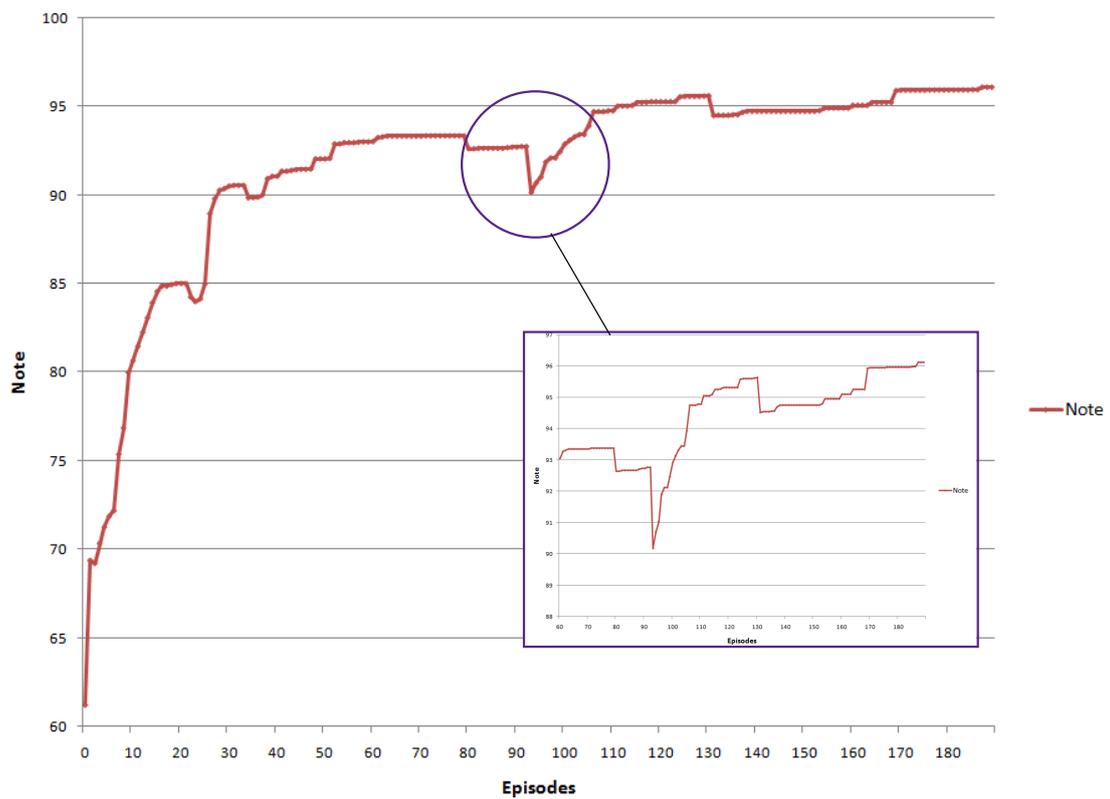


FIGURE 10.6: Notes des Q-tables produites par chaque épisode d'apprentissage par renforcement. La baisse correspondant à un changement d'avis de l'utilisateur. La remontée correspond à l'adaptation du système.

auprès d'utilisateurs. Il faut également mettre en place l'interface d'explication permettant un retour auprès de l'utilisateur sur le fonctionnement interne du système.

Troisième partie

And now something totally different
Jones et al. (n.d.)

Direction de recherche et perspectives

Ce dernier chapitre est l'occasion, en guise de conclusion, de faire un bilan sur ces 8 dernières années de recherche. C'est en particulier le moment de revenir sur les choix qui ont guidé ces travaux, sur les hypothèses de départ qui ont été un fil conducteur pour les recherches que nous avons menées. A partir de ce bilan, je proposerai quelques directions vers lesquels je veux maintenant me diriger.

11.1 Bilan

Les hypothèses initiales qui ont orienté mes recherches dans le cadre de l'intelligence ambiante sont :

- *la pro-activité*. L'agent intelligent agit sans que l'utilisateur ait besoin de recourir à des ordres explicites.
- *l'intelligibilité et la compréhensibilité*. L'agent doit pouvoir expliquer ses choix et l'utilisateur doit pouvoir *regarder à travers l'agent*.
- *la dualité des acteurs* pour la mise en oeuvre des applications. Le développement d'une application nécessite l'intervention d'un professionnel pour la mise en place de connaissances initiales et de l'utilisateur final pour la spécialisation.
- *Le respect de la vie privée* : je n'ai jamais vraiment considéré ce point, n'étant guidé que par la faisabilité plus que l'éthique.

Est-ce que ces hypothèses (ou absence d'hypothèse pour la vie privée) tiennent toujours ?

Nous allons tout d'abord reconsidérer la première (pro-activité) et ré-examiner dans la suite de ce chapitre les trois suivantes sous l'angle de la protection des données et de la vie privée.

11.1.1 Scénarios, pro-activité, engagement utilisateurs

Weiser, dans sa vision initiale, parlait de l'ordinateur du 21^{ème} siècle. Nous sommes actuellement au 21^{ème} siècle. Avons nous atteint cette vision ? De l'avis général, ce n'est pas le cas et la plupart des publications parlent de l'informatique ubiquitaire comme d'un avenir proche, comme le faisait déjà Weiser en 1991. C'est donc un domaine qui est toujours tout proche (*just around the corner*) depuis 15 ans, mais qui est toujours repoussé à demain, qu'on n'atteint finalement jamais. Ce point de vue est développé par Geneviève Bell et Paul Dourish dans leur article (Bell & Dourish, 2007) écrit en 2005 et paru en 2007 dans le journal *Personal and Ubiquitous Computing* et intitulé *Yesterday's tomorrows : notes on ubiquitous computing's dominant vision*. L'article affirme que cette vision a, en réalité, été atteinte, mais à des détails près, ces différences étant le fruit de nos attentes technologiques qui sont illusoire. Les auteurs s'appuient en particulier sur les exemples de Singapour[ⓐ] et de la Corée du sud, porteuse d'un plan technologique visant à transformer le pays en une société ubiquitaire en 2010, communément appelé *U-Korea*.

La différence est le fruit de nos attentes technologiques qui sont illusoire. Beaucoup de scénarios illustrant l'intelligence ambiante sont trop futuristes, intéressants du point de vue technique, mais souvent inutiles dans la pratique car ils simplifient trop la nature mondaine de notre vie de tous les jours (Barton & Pierce, 2006). Barton et Pierce proposent de les évaluer en quantifiant "la magie" qu'ils contiennent. Les scénarios inappropriés résultent en des systèmes non réalistes que personne ne va adopter, en des évaluations biaisées et en le risque de manquer le potentiel réel d'une technologie.

Dans la vision de Weiser, les infrastructures devaient être continues, sans heurt. Ce n'est pas le cas aujourd'hui et ceci est peut être la raison pour laquelle nous repoussons toujours l'informatique ubiquitaire à demain. D'après les auteurs, les infrastructures sont par nature un fouillis. Elles doivent en permanence être maintenues et renégociées. Leur conclusion est que l'informatique ubiquitaire est une réalité dans la société d'aujourd'hui (ou dans certaines sociétés du moins), mais qu'elle n'est pas aussi propre et ordonnée que ce que nous attendions. Les appareils la composant ne sont pas discrets et invisibles comme Weiser les avait imaginés, mais les usagers les souhaitent au contraire très présents et visibles (on peut notamment penser aux nouveaux téléphones tels que l'*iPhone*). La bonne approche est peut-être de laisser les gens inventer leurs usages de l'informatique ubiquitaire, plutôt que d'essayer de leur en imposer sans qu'ils ne s'en aperçoivent. Il faut leur donner le contrôle.

Cette idée est également développée par Yvonne Rogers (Rogers, 2006). L'auteur résume les efforts fournis depuis une quinzaine d'années pour réaliser des applications d'« informatique calme » mais constate qu'ils n'égalent toujours pas la vision originale.

ⓐ. qui porte le surnom *d'Intelligent Island*

La raison en est simplement qu'atteindre cet objectif implique de résoudre des problèmes d'intelligence artificielle trop difficiles. De plus, le comportement des gens dans la « vraie vie », leurs humeurs et intentions, sont trop complexes pour être correctement intégrés dans un système informatique. L'auteur évoque également le fait que l'informatique calme, qui s'efface et est transparente pour les utilisateurs, n'est peut-être pas ce qu'ils souhaitent. Ceux-ci ne désirent pas dépendre des ordinateurs dans leurs tâches quotidiennes, mais tiennent, pour la plupart, à être actifs et garder le contrôle. L'auteur propose alors d'élargir la portée du domaine de l'informatique ubiquitaire au regard de l'évolution des pratiques constatées ces dernières années et qui est que les technologies *engagent* les utilisateurs. Les applications populaires ne sont pas celles qui sont transparentes et invisibles, mais celles qui engagent la créativité et l'imagination des utilisateurs. C'est vers cette direction, d'après l'auteur, que doit désormais se tourner l'informatique ubiquitaire. Cela signifie entre autre que la pro-activité n'est pas ce qui doit être recherché à tout prix, mais plutôt, comme le dit Rui José (José, 2008) :

« *Proactive people rather than Proactive systems* »

11.1.2 Ethique, respect de la vie privée

L'étude de la faisabilité des scénarios d'intelligence ambiante a été ma principale priorité. De manière un peu simpliste, on pourrait résumer cette approche par : "voyons d'abord si ça marche, on se demandera ensuite si c'est éthiquement correct".

Les problèmes d'éthique et de respect de la vie privée ont été abordés très tôt dans le domaine de l'intelligence ambiante (voir (Bellotti & Sellen, 1993) par exemple). L'évolution rapide de notre société et de nos comportements face aux technologies de l'information me conduit à reconsidérer mes priorités.

Pour la plupart des gens, le terme "privé" désigne le droit de chacun de décider quelle information le concernant est connue et par qui. La notion de "privée" est une notion qui a évolué au cours de l'histoire en s'adaptant en particulier aux évolutions technologiques (Langheinrich, 2001). On retrouve des premières traces en 1361 à travers un arrêt de la justice anglaise à l'encontre des voyeurs et des personnes écoutant aux portes ou aux fenêtres^③. Au 18^{ème} siècle, le parlementaire Anglais William Pitt définit le droit de chacun à ce que sa maison soit une zone privilégiée où même le roi et ses gardes ne peuvent entrer. On parle de *territorial privacy*. Au 19^{ème} siècle, le développement de la presse écrite a apporté de nouvelles dérives. Samuel Warren et Louis Brandeis, dans leur article de référence "The Right of Privacy" (Warren & Brandeis, 1890) définissent "the right to be let alone". Ils dénoncent en particulier la possibilité, grâce aux progrès de la photographie, de pouvoir prendre quelqu'un à la volée sans sa permission. Initialement,

③. peeping toms and eaves-droppers

la prise en photo d'une personne nécessitait qu'elle reste assise sans bouger un long moment, sinon, la photo était floue. Dans les années 30 est apparu la *communication privacy* dans le cadre du procès Olmstead contre le gouvernement des Etats-Unis. Il s'agissait de déterminer la légalité de la mise en place d'enregistrements téléphoniques. Depuis les années 60 – 70, avec l'arrivée et la généralisation des outils informatiques pour le stockage et l'automatisation du traitement des données (par les gouvernements puis par les acteurs privés), nous sommes maintenant dans l'*information privacy*.

Le respect des données privées est-il important ? Est-ce une préoccupation générale ? Scott McNealy, PDG de SUN, résume ainsi une attitude que l'on rencontre fréquemment face aux développements très rapides de la technologie : « *You already have zero-privacy anyway, get over it.* ». Peter Cochrane, ancien dirigeant du laboratoire avancé de recherche de British Telecom soutient que la vie serait actuellement meilleure sans vie privée. Il argumente en particulier les aspects pratiques qui en découleraient :

« *We have never enjoyed total anonymity in the past world of paper, so why should we expect it to be remotely possible in a world of bits ? Not only might it be infeasible to put into effect most of the well intended privacy legislation, it might actually do more harm than good : Should I be knocked unconscious in a road traffic accident in New York – please let the ambulance have my medical record* »

Amitai Etzioni se place sur le plan du bien pour la société :

« *If the FBI is able to decipher secret email messages, it can better prevent terrorists from planning their operations. If newborns are tested for HIV, immediate treatment can significantly increase their life expectancy while revealing information about their parents that those would rather avoid* »

Cet argument présuppose la présence d'un gouvernement dont le contrôle démocratique est suffisant pour garantir qu'il n'agit que dans l'intérêt de ses citoyens et qu'il n'abuse pas de son pouvoir.

Ces positions extrêmes illustrent bien que la défense des données privées ne va pas forcément de soi, d'autant plus que les usagers eux-mêmes sont prêts à y renoncer si le bénéfice qu'ils en retirent est suffisant (cet argument est repris dans la section 11.2.2).

Il y a bien évidemment une implication de l'informatique dans la problématique du respect des données privées. Cette implication s'est considérablement renforcée avec la généralisation d'internet. De part ses caractéristiques, l'intelligence ambiante ajoute encore une part non négligeable de risque dans ce domaine. Comme le souligne Philip Brey (Brey, 2005), les objets intelligents^① construisent généralement un profil de leurs utilisateurs (préférences, comportement, etc). Ils sont par principe connectés au réseau. Le risque de voir circuler librement des données personnelles est donc important.

①. Smart Objects

Selon le principe de l'informatique ubiquitaire, ces objets sont appelés à se multiplier et à devenir invisibles, au risque de nous faire oublier les conséquences de leur présence. Ils perçoivent leur environnement avec une qualité de plus en plus importante (son, vidéo) et deviennent capables de mémoriser et indexer nos activités pour une consultation ultérieure.

Comme le souligne Brey (Brey, 2005), l'intelligence ambiante offre également un grand potentiel pour les entreprises commerciales, leur promettant la transparence totale du marché et un accès direct aux consommateurs. Elle leur offre des moyens de connaître leurs clients et de les atteindre avec des publicités ciblées. Par exemple, la plupart des hypermarchés remplacent actuellement leurs cartes de fidélité par des cartes à puce faisant également office de cartes bleues. Ils ont donc maintenant accès à nos habitudes de consommation, non seulement dans leurs magasins (cas des anciennes cartes de fidélité), mais également chez les autres commerçants, réduisant d'autant notre espace de liberté. Pire encore, si les objets communicants deviennent des agents intelligents de confiance, les entreprises n'auront plus besoin de convaincre directement les consommateurs avec leurs arguments mais pourront s'appuyer sur l'autorité de ces agents intelligents pour convaincre les utilisateurs de faire des achats. Les objets dynamiques pourraient devenir des intermédiaires entre les entreprises et les consommateurs pour influencer leur jugement. Pour arriver à cela, il est important que ces entreprises puissent influencer les algorithmes par lesquels les objets communicants tirent leurs conclusions. Une telle influence pourrait déjà être exercée lors de la phase de conception, lorsque les développeurs travaillent avec des entreprises commerciales pour mettre en place ces nouveaux modèles économiques.

Ce tableau assez sombre pourrait conduire dans une vision pessimiste à rejeter l'informatique ambiante. Faut-il jeter tout le panier si quelques fruits sont pourris ? Ce même constat pessimiste pourrait être fait pour l'Internet en général. Mais les bénéfices d'Internet pour la société sont tels qu'il ne faut pas reculer face aux risques encourus mais plutôt adopter une approche volontariste pour les prendre en compte et essayer de les maîtriser.

En résumé, comme l'indique Langheinrich, la prise en compte de la protection de la vie privée est de notre responsabilité en tant que chercheur :

« We cannot rely on lawmakers and sociologists to be fully aware of the vast possibilities and implications that the technology so obviously presents to us. It is us who need to understand the potential and danger of our advancements, and develop sound conventions and guidelines according to well-established principles that will help us drive technology into a responsible and socially acceptable direction. »

11.2 Quelle direction ?

11.2.1 Principes

Répartition des rôles

Notre hypothèse de départ était que la construction d'une application ambiante repose sur un développeur et un utilisateur. Le rôle du développeur est de programmer les "briques" élémentaires (les services) et de fournir une pré-connaissance initiale sous forme d'un modèle de contexte générique. Ce modèle est ensuite spécialisé par l'utilisateur final. Ceci conduit à des applications fortement "colorées" par le concepteur initial et répondant à un besoin supposé de l'utilisateur.

Je souhaite maintenant déplacer cette frontière entre les deux acteurs. Le développeur fournit une liste de services de perception, d'action et un ou plusieurs moteurs génériques de reconnaissance de contexte. Il ne fournit par contre plus de connaissance initiale. Ceci permet de prendre en compte le constat d'Yvonne Rogers ou de José Ruiz appelant à augmenter la pro-activité de l'utilisateur plutôt que de la substituer par une pro-activité du système. En impliquant plus l'utilisateur dans la co-construction de son système, il réduit d'autant la part délicate d'interprétation de ses besoins par la machine. Diminuer l'entrée initiale de connaissance par le développeur permet également de diminuer les risques d'influence de compagnies externes cités lors de la section précédente.

L'utilisateur passe d'un rôle d'adaptation à celui plus impliquant de construction de scénarios. Cette construction concerne le modèle de contexte mais peut également concerner la phase d'interprétation des données de perception. Ce point est repris au paragraphe 11.2.3. Ce déplacement des rôles va nous conduire à nous intéresser plus fortement à la communauté « *end-user programming* » .

Scénario

Un des objectifs de l'intelligence ambiante est d'apporter de l'aide contextuelle aux usagers. Comme nous l'avons souligné au cours du paragraphe 11.1.1, la pro-activité du système ne doit pas disparaître mais encourager et assister la pro-activité de l'utilisateur. La réalisation d'une application ambiante "tout en un", capable d'assurer des scénarios complets, est d'une incroyable complexité qui risque de ne pas tenir face au bénéfice **réel** que l'utilisateur pourrait en retirer. Une telle application risque de plus d'être très liée à un scénario type, le coût humain et financier pour la faire évoluer étant

très élevé. L'informatique ambiante n'est pas calme et transparente mais hétérogène et "fouillée".

Parmi les grandes familles d'applications pour l'intelligence ambiante, celle que je souhaite développer concerne la réduction de la charge cognitive des usagers (cf. article de Pattie Maes (Maes, 1994)). Nous sommes entourés de services informatiques fournissant de l'information (personnels ou professionnels de type agenda, carnets d'adresses, todo lists, emails, ... ou généralistes de type météo, information, localisation ...). Ces services sont autant de sources d'information. Ils sont généralement externes et ne se connaissent pas. Une première façon de traiter ces données est classiquement de créer une page de résumé présentant les flux RSS associés. Ces pages sont reconstruites côté serveurs et affichées dans un navigateur Web. L'utilisateur construit interactivement sa page en juxtaposant des flux de données. Ces services sont proposés par exemple par google (iGoogle)[Ⓓ], Yahoo (MyYahoo)[Ⓔ] ou Netvibes[Ⓒ]. Au lieu d'une simple juxtaposition, ces informations peuvent également être combinées au sein d'un programme : ce sont des Mashups. Pour reprendre une définition de Duane Merrill (Merrill, 2006) :

« a mashup Web site is characterized by the way in which it spreads roots across the Web, drawing upon content and functionality retrieved from data sources that lay outside of its organizational boundaries... it was borrowed from the pop music scene, where a mashup is a new song that is mixed from the vocal and instrumental tracks from two different source songs (usually belonging to different genres). »

Ces Mashups sont généralement fournis par des développeurs. Il existe des environnements graphiques de construction pour l'utilisateur final (voir Yahoo Pipes[Ⓔ] par exemple).

La sélection des informations pertinentes et leur traitement est très dépendant de l'utilisateur mais également et surtout de son contexte courant. Il existe des travaux s'intéressant à l'intégration du contexte aux Mashups (voir par exemple (Li et al., 2008) ou (Brodt et al., 2008)). Ce concept de combinaison contextuelle d'information est mis en avant dans ce que Thomas Vander Wal appelle le « *Personal InfoCloud* ». Le Personal InfoCloud est défini comme un changement de paradigme dans l'accès au Web : on ne va plus chercher "leur" information (centré sur le fournisseur de contenu) à travers un navigateur unique mais on remet l'utilisateur au centre du processus. Il construit sa propre information en attirant des sources d'information provenant d'une variété de dispositifs. Il repose sur l'apprentissage de l'utilisation faite par l'utilisateur de cette information et sur les contextes d'usage associés.

Les premiers scénarios que je souhaite étudier concerne la mise en place d'un « Personal InfoCloud » reposant sur les informations du Web ainsi que sur toute information

Ⓓ. <http://www.google.fr/ig>

Ⓔ. <http://fr.my.yahoo.com/>

Ⓒ. <http://www.netvibes.com>

Ⓔ. <http://pipes.yahoo.com>

provenant des services disponibles dans l'environnement. L'objectif est de donner à l'utilisateur final la possibilité de construire interactivement et surtout opportunistiquement ses assistants contextuels de combinaison d'information.

11.2.2 Confidentialité de mes données

La sécurité et le respect de la confidentialité des données dans les applications d'Intelligence Ambiante ne doivent pas être envisagés après coup mais faire partie de la réflexion initiale. On se place dans le cadre du *Privacy by Design*.

La sécurité totale n'est techniquement pas réaliste et serait certainement trop contraignante pour l'utilisateur final. David Brin (Brin, 1999) argumente qu'une réaction trop forte aux problèmes actuels conduisant à "verrouiller" l'ensemble des données (société du secret) peut conduire à une situation pire : l'illusion de la protection de ses données privées. Les personnes ou les gouvernements mal intentionnés trouveront toujours le moyen d'y accéder sans aucun contrôle. Il est à la limite préférable de laisser un accès ouvert total : on peut surveiller ceux qui nous surveillent.

« if everybody has access to the same information, it ceases to be a weapon in the hands of a few well-informed »

Brin préconise une approche intermédiaire où on investit dans la protection des données réellement privées et sensibles (religion, orientation sexuelle etc) et où les données *semi-publiques* (préférences commerciales etc) sont laissées en libre accès.

Langheinrich propose la protection de la vie privée à partir des propriétés suivantes :

Information : un agent intelligent doit annoncer tout ce qu'il enregistre, à l'instar de ce que propose la spécification P3P[Ⓟ] (Cranor & al., n.d.) du W3C.

Choix et consentement ; l'utilisateur doit consentir explicitement à l'enregistrement de ses données personnelles. Il doit avoir le contrôle.

Anonymat et Pseudonyme : il faut recourir autant que possible à l'anonymat mais cela peut être une sérieuse limitation dans les environnements intelligents. Les pseudonymes permettent de préserver une part d'anonymat tout en proposant des services personnalisés.

Proximité : un "smart object" ne devrait pouvoir enregistrer des données que si son propriétaire est à côté. Cela évite de s'en servir pour "espionner" les autres.

L'utilisation d'un pseudonyme n'est généralement pas suffisante : l'identité de la personne peut être retrouvée par recoupement entre toutes les données. Le choix et consentement est sur le principe une bonne approche mais elle pose néanmoins plusieurs problèmes : la *liberté de choix* et est-ce que *ce choix est éclairé* ?

Ⓟ. Platform for Privacy Preference

La liberté de choix : un refus peut entraîner le refus d'accès à un service qui m'est nécessaire. Imaginons que les bâtiments administratifs deviennent "intelligents" et que pour accéder aux services offerts, je dois accepter d'être identifié et que mon activité soit tracée. Si refuser ce suivi signifie ne pas pouvoir accéder au bâtiment, est ce que je vais pouvoir refuser ? Je sais que lorsque je paye avec ma carte bleue, je suis localisé. Est-ce que je peux néanmoins me permettre de ne plus utiliser ma carte bleue ?

Un Choix éclairé : il n'est pas certain que lorsqu'un usager accepte de fournir des informations, il réalise pleinement ce que cela signifie. Cohen (Cohen, 1994) cite par exemple le cas de personnes laissant un volume de leur disque Mac accessible à tout le monde. Elles sont troublées lorsqu'elles peuvent voir qui accède à leur volume partagé sur leur disque dur, et en particulier lorsqu'elle constate le nombre d'inconnus qui y accèdent. Plus généralement, les utilisateurs ne mesurent généralement pas tous les risques et toutes les conséquences associés à la fourniture de leurs données. Un million d'utilisateurs de la Royal Bank of Scotland n'imaginaient pas par exemple en confiant leur données personnelles qu'elles se retrouveraient sur le disque dur d'un ordinateur vendu par erreur sur eBay (45 Euros!) (Blog, n.d.). Les données que l'on juge peu importantes maintenant et que l'on donne donc sans problème ne peuvent-elles pas devenir sensibles plus tard, dans un contexte différent ? La durée de rétention des données est un élément sensible. La directive Européenne 2002/58/CE (Européen, n.d.) précise que les données personnelles ne doivent pas être gardées plus que nécessaire, ce qui reste difficilement vérifiable et applicable. Google a accepté de passer d'une durée illimitée à 2 ans puis 9 mois. Ce n'est pas encore la durée de 6 mois préconisée par l'ensemble des CNILs Européennes. Google affiche une éthique par rapport aux données personnelles "don't be evil". Ce slogan a déjà montré quelques failles lorsque des intérêts économiques importants sont en jeu. Et qu'en sera-t'il de ce slogan lorsque la direction de Google changera ? Indépendamment de mesurer la portée de leurs actes, la plupart des usagers sont prêts de renoncer consciemment au caractère privé de leurs données s'ils jugent le bénéfice suffisant (Bellotti & Sellen, 1993). Google est là encore un très bon exemple. Nombre de mes collègues, malgré une politique forte de sensibilisation à la confidentialité des données par leur tutelle, sont prêts à utiliser Google Doc[Ⓒ] pour monter des projets de recherche... On peut également citer le cas de Facebook où on accepte l'installation d'applications "ludiques" sur son compte en confirmant bien que ces applications tiers vont avoir accès à toutes nos données.

De nombreux travaux de recherche sont actuellement menés autour du respect de la confidentialité des données dans le contexte de l'Intelligence Ambiante. Il y a les approches basées sur l'encryptage des données (voir (Pearson, 2002) ou (Huberman & Hogg, 2002) par exemple). Il existe également de nombreux travaux dans la communauté multi-agents autour de la notion de confiance et de réseau de confiance (voir (Castelfranchi & Falcone, 1998) pour la formalisation de la confiance, (Melaye & Demazeau, 2005) pour

Ⓒ. <http://docs.google.com/>

une approche bayésienne de ce modèle). Ces directions sont très prometteuses. Je souhaite compléter cet aspect sous un autre angle illustré dans la section suivante.

11.2.3 Schéma général

On peut considérer deux grandes catégories d'architecture pour les assistants ambiants : les assistants de type "*concierge*" (terminologie utilisée en particulier par James Crowley) et les assistants de type "*ange gardien*".

Les assistants de type concierge (cf. concierge de grand hôtel), sont attachés à un lieu géographique. Ils sont présents dans le système informatique de ce lieu et peuvent répondre à des requêtes des usagers. Ils fournissent des services personnalisés et possèdent pour cela un modèle de chaque utilisateur.

Les assistants de type "ange gardien" sont des assistants privés, personnels. Ils sont au plus près de l'utilisateur et l'accompagnent en permanence. Les téléphones mobiles intègrent de plus en plus de données et deviennent de fait de plus en plus personnels. Ces assistants peuvent trouver une place naturelle dans ces systèmes.

Ce deuxième paradigme est celui que je souhaite développer car il me semble plus adapté au problème de contrôle de mes données personnelles. Cet assistant ne doit pas "sortir" du téléphone. Il utilise les services proposés par l'environnement comme autant de sources d'information. Il utilise l'ensemble des données perçues pour inférer des connaissances de plus haut niveau sur son "propriétaire". Il réalise lui-même ces inférences sans les déléguer à un système externe et ne publie pas ses résultats. Il peut être néanmoins nécessaire pour certains scénarios de réaliser des traitements combinant les informations provenant de plusieurs utilisateurs. Ce problème difficile est étudié dans le cadre des recherches sur les algorithmes de fouille de données répartis préservant les contraintes privées (voir (Sakuma et al., 2008) par exemple).

Quelle modèle de contexte utiliser ? Une de nos hypothèses initiales, reprise des travaux de Bellotti et Edwards (Bellotti & Edwards, 2001), est que l'assistant doit être intelligible et transparent afin de gagner la confiance de l'utilisateur. La transparence nous a en particulier conduit à utiliser une représentation des états sous forme de prédicats pour l'apprentissage renforcé. La transparence est-elle une nécessité ? la communication et la collaboration entre l'assistant et l'utilisateur repose sur le partage de concepts communs, sur l'intelligibilité mutuelle. Cette intelligibilité mutuelle n'implique pas une représentation explicite partagée entre l'assistant et l'utilisateur. On peut laisser le système construire sa propre représentation interne, éventuellement "obscur". La seule contrainte est qu'il soit capable, sur requête explicative de l'utilisateur, de lui répondre en s'appuyant sur des concepts communs.

Cette perte de transparence peut entraîner une diminution de la confiance. En reprenant l'argument de Patty Maes (Maes, 1994), le fait que l'utilisateur soit le seul maintenant impliqué dans la co-construction de son assistant (plus de comportement initial en provenance du développeur) peut en contre-partie apporter une confiance supplémentaire. Cette perte de transparence peut également être positive du point de la protection de mes données privées. Il n'est pas accessible directement à la compréhension d'un lecteur externe mais répond à des requêtes ciblées.

Mon objectif est de co-construire un assistant personnel partageant un sens commun avec son utilisateur. Il s'agit de revenir à "l'intimate computing" tel qu'évoqué en 1994 par Lamming et Al. (Lamming & Flynn, 1994). Je souhaite en particulier étudier cette co-construction de sens sous l'angle de l'approche biologique de la cognition : l'énaction (Varela et al., 1992). L'énaction stipule que le comportement d'un système autonome est façonné par l'histoire de son interaction avec l'environnement. Il existe un courant de l'intelligence artificielle qui s'inspire de ces principes : l'intelligence artificielle énaactive (Froese & Ziemke, 2009). Ce courant est en particulier présent en robotique dans la mise en place de systèmes sensori-moteurs. Je souhaiterai maintenant étudier ces approches dans le cadre de l'intelligence ambiante.

Références bibliographiques

- Abowd, G. D., Atkeson, C. G., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N. & Tani, M. (1996). Teaching and learning as multimedia authoring : the classroom 2000 project, *Proceedings of the fourth ACM international conference on Multimedia*, ACM, Boston, Massachusetts, United States, pp. 187–198.
URL: <http://portal.acm.org/citation.cfm?doid=244130.244191>
- Abowd, G. D. & Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing, *ACM Trans. Comput.-Hum. Interact.* **7**(1) : 29–58.
URL: <http://portal.acm.org/citation.cfm?id=344988>
- Adamczyk, P. D. & Bailey, B. P. (2004). If not now, when? : the effects of interruption at different moments within task execution, *CHI '04 : Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, pp. 271–278.
- Adams, N., Gold, R., Schilit, B. N., Tso, M. & Want, R. (1993). An infrared network for mobile computers, *Proceedings USENIX Symposium on Mobile & Location-independent Computing*, pp. 41–52.
- Allen, J. F. (1983). Maintaining knowledge about temporal intervals, *Commun. ACM* **26**(11) : 832–843.
URL: <http://doi.acm.org/10.1145/182.358434>
- Allen, J. F. (1984). Towards a general theory of action and time, *Artificial Intelligence* **23**(1) : 123–154.
URL: [http://dx.doi.org/10.1016/0004-3702\(84\)90008-0](http://dx.doi.org/10.1016/0004-3702(84)90008-0)
- Aström, K. J. (1965). Optimal control of markov decision processes with incomplete state estimation, *Journal of Mathematical Analysis and Applications* **10** : 174–205.
- Balaguer J.F. (1993). *Virtual Studio : Un système d'animation en environnement virtuel*, PhD thesis, EPFL DI-LIG.
- Ballet, P., Rodin, V. & Tisseau, J. (1997). Edge detection using a multiagent system, *SCIA '97, IAPR Scandinavian Conference on Image Analysis*, Lappeenranta (Finlande), pp. 621–629.

- Ballet, P., Tisseau, J. & Harrouet, F. (1997). A multiagent system to model an human secondary immune response, *IEEE Transactions on Systems, Man, and Cybernetics*, Orlando (USA).
- Barton, J. & Pierce, J. (2006). Quantifying magic in ubicomp systems scenarios, *Position Paper for UbiSys 2006 Workshop at Ubicomp2006*, Orange County, California.
URL: <http://www.magic.ubc.ca/ubisys/positions/barton.pdf>
- Bell, G. & Dourish, P. (2007). Yesterday's tomorrows : notes on ubiquitous computing's dominant vision, *Personal Ubiquitous Comput.* **11**(2) : 133–143.
URL: <http://portal.acm.org/citation.cfm?id=1229069>
- Bellman, R. (1957). A markov decision process, *Journal of Mathematical Mechanics* **6** : 679–684.
- Bellotti, V. & Edwards, K. (2001). Intelligibility and accountability : human considerations in context-aware systems, *Hum.-Comput. Interact.* **16**(2) : 193–212.
URL: <http://portal.acm.org/citation.cfm?id=1463108.1463113>
- Bellotti, V. & Sellen, A. (1993). Design for privacy in ubiquitous computing environments, *Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work*, Kluwer Academic Publishers, Milan, Italy, pp. 77–92.
URL: <http://portal.acm.org/citation.cfm?id=1241940>
- Berns, K., Dillmann, R. & Zachmann, U. (1992). Reinforcement Learning for the Control of an Autonomous Mobile Robot, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, USA, pp. 1808–1815.
- Bilmes, J. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, *Technical Report Technical Report ICSI-TR-97-021*, University of Berkeley, CA, USA.
- Blog (n.d.). Un million de données bancaires vendues sur eBay!, <http://www.generation-nt.com/ebay-donnees-bancaires-perte-royal-bank-scotland-actualite-142421.html>.
URL: <http://www.generation-nt.com/ebay-donnees-bancaires-perte-royal-bank-scotland-actualite-142421.html>
- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers, in A. Press (ed.), *In Proceedings of the fifth annual workshop on Computational learning theory (COLT 92)*, number ISBN 0-89791-497-X, New-York, USA, pp. 144–152.
- Bourdon, F. (1997). The interactional semantics of knowledge, *International Joint Conference on Artificial Intelligence, Poster Session*, Nagoya (Japan), p. 15.
- Brdiczka, O., Maisonnasse, J., Reignier, P. & Crowley, J. L. (2009). Detecting small group activities from multimodal observations, *Applied Intelligence* **30**(1) : 47–57.
URL: <http://portal.acm.org/citation.cfm?id=1485066>
- Brey, P. (2005). Freedom and privacy in ambient intelligence, *Ethics and Inf. Technol.* **7**(3) : 157–166.
URL: <http://portal.acm.org/citation.cfm?id=1133862>
- Brin, D. (1999). *The Transparent Society : Will Technology Force Us to Choose Between Privacy and Freedom ?*, Perseus Books Group.

- URL:** <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0738201448>
- Brodt, A., Nicklas, D., Sathish, S. & Mitschang, B. (2008). Context-Aware mashups for mobile devices, *Web Information Systems Engineering - WISE 2008*, pp. 280–291.
URL: http://dx.doi.org/10.1007/978-3-540-85481-4_22
- Brown, P. (1996). The stick-e document : a framework for creating context-aware applications, *Proceedings of EP'96, Palo Alto*, also published in it EP-odd, pp. 272, 259.
URL: <http://www.cs.kent.ac.uk/pubs/1996/396>
- Brumitt, B., Meyers, B., Krumm, J., Kern, A. & Shafer, S. (2000). EasyLiving : technologies for intelligent environments, *Handheld and Ubiquitous Computing*, pp. 97–119.
URL: http://dx.doi.org/10.1007/3-540-39959-3_2
- Brutzman D.P (1994). *A virtual world for an autonomous underwater vehicle*, PhD thesis, Naval Postgraduate School, Monterey, California.
- Buffet, O. (2003). *Une double approche modulaire de l'apprentissage par renforcement pour des agents intelligents adaptatifs*, PhD thesis, Université Henri Poincaré, Nancy 1. Laboratoire Lorrain de recherche en informatique et ses applications (LORIA).
- Byun, H. E. & Cheverst, K. (2001). Exploiting user models and context-awareness to support personal daily activities, *Workshop in UM2001 on User Modeling for Context-Aware Applications*.
- Caporossi, A., Hall, D., Reignier, P. & Crowley, J. L. (2004). Robust visual tracking from dynamic control of processing, *PETS '04 - Performance and Evaluation of Tracking and Surveillance*, PETS, Czech technical University.
- Carlsson C. & Hagsang O. (1993). Dive – a platform for multi-user virtual environment, *Computer and Graphics* pp. 663–669.
- Cassandra, A. R. (1999). POMDPs for dummies : Page 5, <http://www.cs.brown.edu/research/ai/pomdp/tutorial/pomdp-solving.html>.
URL: <http://www.cs.brown.edu/research/ai/pomdp/tutorial/pomdp-solving.html>
- Castelfranchi, C. & Falcone, R. (1998). Principles of trust for MAS : cognitive anatomy, social importance, and quantification, *Proceedings of the 3rd International Conference on Multi Agent Systems*, IEEE Computer Society, p. 72.
URL: <http://portal.acm.org/citation.cfm?id=852234>
- Cheverst, K., Byun, H., Fitton, D., Sas, C., Kray, C. & Villar, N. (2005). Exploring issues of user model transparency and proactive behaviour in an office environment control system, *User Modeling and User-Adapted Interaction* **15**(3) : 235–273.
URL: <http://dx.doi.org/10.1007/s11257-005-1269-8>
- Clemente, P. J., Conejero, J. M., Hernández, J. & Sánchez, L. (2009). HAAIS-DSL : DSL to develop home automation and ambient intelligence systems, *Proceedings of the Second Workshop on Isolation and Integration in Embedded Systems*, ACM, Nuremberg, Germany, pp. 13–18.
URL: <http://portal.acm.org/citation.cfm?id=1519130.1519133>
- Codella C.F, Jalili R., Koved L. & Lewis J.B. (1993). A toolkit for developing multi-user, distributed virtual environments, *IEEE Virtual Reality Annual International*

Symposium.

- Coen, M. H. (1998). Design principles for intelligent environments, *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, American Association for Artificial Intelligence, Madison, Wisconsin, United States, pp. 547–554.
URL: <http://portal.acm.org/citation.cfm?id=295733>
- Cohen, J. (1994). Out to lunch : Further adventures monitoring background activity.
- Cook, D., Youngblood, M., Heierman, E., Gopalratnam, K., Rao, S., Litvin, A. & Khawaja, F. (2003). MavHome : an agent-based smart home, *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pp. 521–524.
- Cortes, C. & Vapnik, V. (1995). Support-vectors networks, *Machine Learning* **20**(3) : 273–297.
- Craik, K. (1943). *The Nature of Exploration*, Cambridge University Press, Cambridge, England.
- Cranor, L. & al. (n.d.). P3P : the platform for privacy preferences, <http://www.w3.org/P3P/>.
URL: <http://www.w3.org/P3P/>
- Crowley, J., Coutaz, J., Rey, G. & Reignier, P. (2002). Perceptual components for context aware computing, *UbiComp 2002 : Ubiquitous Computing*, pp. 117–134.
URL: http://dx.doi.org/10.1007/3-540-45809-3_9
- Davies, N., Cheverst, K., Mitchell, K. & Friday, A. (1999). 'Caches in the air' : Disseminating tourist information in the guide system, *Mobile Computing Systems and Applications, IEEE Workshop on*, Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, p. 11.
- Dey, A. K. (2001). Understanding and using context, *Personal and Ubiquitous Computing* **5**(1) : 4–7.
URL: <http://dx.doi.org/10.1007/s007790170019>
- Dey, A. K., Hamid, R., Beckmann, C., Li, I. & Hsu, D. (2004). a CAPpella : programming by demonstration of context-aware applications, ACM, Vienna, Austria, pp. 33–40.
URL: <http://portal.acm.org/citation.cfm?id=985692.985697>
- Dourish, P. (2004). What we talk about when we talk about context, *Personal Ubiquitous Comput.* **8**(1) : 19–30.
- Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J. & Burgelman, J.-C. (2001). Scenarios for ambient intelligence in 2010, *Technical report*, IST Advisory Group (ISTAG), IPTS-Seville.
URL: <http://cordis.europa.eu/ist/istag-reports.htm>
- Duval, T., Morvan, S., Reignier, P., Harrouet, F. & Tisseau, J. (1997). Arévi : une boîte à outils 3d pour des applications coopératives, *Calculateurs Parallèles* **9**(2) : 239–250.
- Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J. & Welch, B. (1992). Liveboard : a large interactive display supporting group meetings, presentations, and remote collaboration, *Pro-*

- ceedings of the SIGCHI conference on Human factors in computing systems*, ACM, Monterey, California, United States, pp. 599–607.
URL: <http://portal.acm.org/citation.cfm?id=142750.143052>
- Erickson, T. (2002). Some problems with the notion of context-aware computing, *Commun. ACM* **45**(2) : 102–104.
URL: http://portal.acm.org/ft_gateway.cfm?id=503154
- Européen, P. (n.d.). EUR-Lex - 32002L0058 - FR, <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:FR:HTML>. Directive 2002/58/CE du Parlement européen et du Conseil du 12 juillet 2002 concernant le traitement des données à caractère personnel et la protection de la vie privée dans le secteur des communications électroniques (directive vie privée et communications électroniques).
URL: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:FR:HTML>
- FAME (2001). Fame : Facilitating agent for multi-cultural exchange (wp4), European Commission project IST-2000-28323 .
- Fisher, R. (1938). The statistical utilization of multiple measurements, *Annals of Eugenics* **8** : 376–386.
- Fisher, R. & al. (n.d.). Caviar : Context aware vision using image-based active recognition.
URL: <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>
- Froese, T. & Ziemke, T. (2009). Enactive artificial intelligence : Investigating the systemic organization of life and mind, *Artif. Intell.* **173**(3-4) : 466–500.
URL: <http://portal.acm.org/citation.cfm?id=1501318>
- Georgalas, N., Ou, S., Azmoodeh, M. & Yang, K. (2007). Towards a Model-Driven approach for Ontology-Based Context-Aware application development : A case study, pp. 21–32.
- Gobbetti E. (1993). *Virtuality Builder II, vers une architecture pour l'interaction avec des mondes synthétiques*, PhD thesis, EPFL DI-LIG.
- Godoy, D. & Amandi, A. (2005). User profiling for web page filtering, *Internet Computing, IEEE* **9**(4) : 56–64.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1463162
- Grabish, M. (2006). L'utilisation de l'intégrale de choquet en aide multicritère à la décision, *Newsletter of the European Working Group Multicriteria Aid for Decisions* **3**(14).
- Green M., Shaw C. & White L. (1993). Minimal reality toolkit, version 1.3, *Technical report*, Department of Computing Science, University of Alberta.
- Greenberg, S. (2001). Context as a dynamic construct, *Human-Computer Interaction* **16** : 257–268.
URL: http://dx.doi.org/10.1207/S15327051HCI16234_09
- Harrouet, F., Cozien, R., Reignier, P. & Tisseau, J. (1997). oris, un langage pour simulations multi-agents, *Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents*, La Colle-sur-Loup.

- Hartman J. & Wernecke J. (1996). *The VRML 2.0 Handbook : Building Moving Worlds on the Web*, Addison-Wesley Publishing Company.
- Horvitz, E., Kadie, C., Paek, T. & Hovel, D. (2003). Models of attention in computing and communication : from principles to applications, *Commun. ACM* **46**(3) : 52–59.
- Hsu, C.-W., Chang, C.-C. & Lin, C.-J. (n.d.). A practical guide to support vector classification, *Technical report*, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.
- Huberman, B. A. & Hogg, T. (2002). Protecting privacy while revealing data, *Nat Biotech* **20**(4) : 332.
URL: <http://dx.doi.org/10.1038/nbt0402-332>
- Isbell, C., Shelton, C. R., Kearns, M., Singh, S. & Stone, P. (2001). A social reinforcement learning agent, *AGENTS '01 : Proceedings of the fifth international conference on Autonomous agents*, ACM Press, New York, NY, USA, p. 377–384.
URL: <http://portal.acm.org/citation.cfm?id=375735.376334>
- Jaulmes, R., Pineau, J. & Precup, D. (2005). Learning in Non-Stationary partially observable markov decision processes, *ECML Workshop on Reinforcement Learning in Non-Stationary Environments*, Porto, Portugal.
URL: <http://www.cs.mcgill.ca/jpineau/publications.html>
- Jones, T., Palin, M., Idle, E., Cleese, J., Chapman, G. & Gilliam, T. (n.d.). Monthly python's flying circus.
- José, R. (2008). UbiComp 2.0 : From envisioning a future to being part of a new reality, Plenary Talk at the UCAmI'08 conference.
- Kaelbling, L. P. (2004). Life-Sized learning. Published : Lecture at CSE Colloquia.
URL: <http://www.uwtv.org/programs/displayevent.aspx?rID=3566>
- Kay, J., Kummerfeld, B. & Lauder, P. (2003). Managing private user models and shared personas, *UM03 Workshop on User Modeling for Ubiquitous Computing*, p. 1–11.
- Kozierok, R. & Maes, P. (1993). A learning interface agent for scheduling meetings, *IUI '93 : Proceedings of the 1st international conference on Intelligent user interfaces*, ACM Press, New York, NY, USA, pp. 81–88.
URL: <http://portal.acm.org/citation.cfm?id=169908>
- Lamming, M. & Flynn, M. (1994). Forget-me-not : intimate computing in support of human memory, *Proceedings FRIEND21 Symposium on Next Generation Human Interfaces*.
URL: <http://citeseer.nj.nec.com/lamming94forgetmenot.html>
- Langheinrich, M. (2001). Privacy by design - principles of Privacy-Aware ubiquitous systems, *Proceedings of the 3rd international conference on Ubiquitous Computing*, Springer-Verlag, Atlanta, Georgia, USA, pp. 273–291.
URL: <http://portal.acm.org/citation.cfm?id=741336>
- Leyens, J. & Rimé, B. (1974). L'effet des facteurs écologiques et de signaux non verbaux sur les structures affectives dans les petits groupes, *L'année psychologique* **74**(2) : 487–500.
URL: http://www.persee.fr/web/revues/home/prescript/article/psy_0003-

- 5033_1974_num_74_2_28058
- Li, Y., Fang, J. & Xiong, J. (2008). A Context-Aware services Mash-Up system, *Grid and Cooperative Computing, International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 707–712.
- Littman, M. L. (1994). Markov games as a framework for Multi-Agent reinforcement learning, *IN PROCEEDINGS OF THE ELEVENTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING*, pp. 157–163.
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.8623>
- Lueg, C. (2002). Looking under the rug : On Context-Aware artifacts and socially adept technologies, *Proceedings of Workshop "The Philosophy and Design of Socially Adept Technologies"*, *ACM SIGCHI Conference on Human Factors in Computing Systems*.
- Macedonia M.R. (1995). *A Network software Architecture For Large Scale Virtual Environments*, PhD thesis, Naval Postgraduate School, Monterey, California.
- Maes, P. (1994). Agents that reduce work and information overload, *Commun. ACM* **37**(7) : 30–40.
URL: <http://portal.acm.org/citation.cfm?id=176792>
- Martin, M. (1997). *Application d'un simulateur multi-agents à la modélisation de populations d'invertébrés en milieu bocager*, Master's thesis, Ecole Nationale Supérieure Agronomique de Rennes.
- Matthews, T., Dey, A. K., Mankoff, J., Carter, S. & Rattenbury, T. (2004). A toolkit for managing user attention in peripheral displays, *UIST '04 : Proceedings of the 17th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, pp. 247–256.
- Mcfarlane, D. C. (2002). Comparison of four primary methods for coordinating the interruption of people, *Human-Computer Interaction* **17** : 63–139.
- Melaye, D. & Demazeau, Y. (2005). Bayesian dynamic trust model, *Multi-Agent Systems and Applications IV*, pp. 480–489.
URL: http://dx.doi.org/10.1007/11559221_48
- Merrill, D. (2006). Mashups : The new breed of web app.
URL: <http://www.ibm.com/developerworks/library/x-mashups.html>
- Millán, J. & Torras, C. (1994). Efficient Reinforcement Learning of Navigation Strategies in an Autonomous Robot, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Munich, Germany.
- Mitchell, T. (1997). *Machine Learning*, international edition edn, McGraw Hill, New York, USA.
- Mozer, M. C. (1998). The neural network house : An environment that adapts to its inhabitants, *Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, AAAI Press, Menlo Park, CA, pp. 110–114.
- Muehlenbrock, M., Brdiczka, O., Snowdon, D. & Meunier, J.-L. (2004). Learning to detect user activity and availability from a variety of sensor data, *In Proceedings*

- of *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 13–24.
- Muir, B. M. & Moray, N. (1996). Trust in automation. part II. experimental studies of trust and human intervention in a process control simulation, *Ergonomics* **39**(3) : 429.
URL: <http://www.informaworld.com/10.1080/00140139608964474>
- Murata, T. (1989). Petri nets : Properties, analysis and applications, *Proceedings of the IEEE*, Vol. 4, pp. 541–580.
- Nardi, B. (1996). *Context and consciousness : activity theory and human-computer interaction*, MIT Press, Cambridge Mass.
- Nardi, B. A. (1995). Studying context : a comparison of activity theory, situated action models, and distributed cognition, *Context and consciousness : activity theory and human-computer interaction*, Massachusetts Institute of Technology, pp. 69–102.
URL: <http://portal.acm.org/citation.cfm?id=223830>
- Nilsson, N. J. (1982). *Principles of Artificial Intelligence*, Springer.
- Norman, D. A. (1999). *The Invisible Computer*, MIT Press.
URL: <http://mitpress.mit.edu/catalog/item/default.asp?tttype=2&tid=5160>
- Pearson, S. (2002). Trusted agents that enhance user privacy by Self-Profiling, *AAMAS WORKSHOP - SPECIAL TRACK ON PRIVACY*, Vol. 2631, p. 113.
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.9840>
- Petri, C. A. (1962). *Kommunikation mit Automaten*, PhD thesis, Institut fuer Instrumentelle Mathematik, Bonn, Germany.
- Pires, L. F., Maatjes, N., Sinderen, M. & Costa, P. D. (2008). Model-Driven approach to the implementation of Context-Aware applications using rule engines, *Constructing Ambient Intelligence*, pp. 104–112.
URL: http://dx.doi.org/10.1007/978-3-540-85379-4_13
- Poggi, A. & Adorni, G. (1996). A multi-language environment to develop multi-agents applications, *ECAI*, Budapest (Hungary), pp. 325–340.
- Pousman, Z. & Stasko, J. (2006). A taxonomy of ambient information systems : four patterns of design, *AVI '06 : Proceedings of the working conference on Advanced visual interfaces*, ACM, New York, NY, USA, pp. 67–74.
- Puzicha, J., Hofmann, T. & Buhmann, J. (1997). Non-parametric similarity measures for unsupervised texture segmentation and image retrieval, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 267–272.
URL: <http://dx.doi.org/10.1109/CVPR.1997.609331>
- Qian, R., Sezan, M. & Mathews, K. (1998). Face tracking using robust statistical estimation, *Proceedings of Workshop on Perceptual User Interfaces*, San Francisco.
- Remagnino, P. & Foresti, G. L. (2005). Ambient intelligence : A new multidisciplinary paradigm, *Systems, Man and Cybernetics, Part A, IEEE Transactions on* **35**(1) : 1–6.
URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1369340&isnumber=29969

- Resnick, M. (96). Starlogo : An environment for decentralized modeling and decentralized thinking, *CHI*, pp. 11–12.
- Ribeiro, P. & Victor, J. S. (2005). Human activity recognition from video : modeling, feature selection and classification architecture, *Proceedings of International Workshop on Human Activity Recognition and Modelling (HAREM)*, pp. 61–70.
- Richard, N. & Yamada, S. (2007). Advances in ambient intelligence, Vol. 164 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, Nieuwe Hemweg 6B, 1013 BG Amsterdam, The Netherlands, p. 123–142.
URL: <http://www.iospress.nl/loadtop/load.php?isbn=9781586038007>
- Rodden, T., Chervest, K., Davies, N. & Dix, A. (1998). Exploiting context in HCI design for mobile systems, in *Workshop on Human Computer Interaction with Mobile Devices*.
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.1279>
- Rogers, Y. (2006). Moving on from weiser’s vision of calm computing : Engaging UbiComp experiences, *UbiComp 2006 : Ubiquitous Computing*, pp. 404–421.
URL: http://dx.doi.org/10.1007/11853565_24
- Roohlf J. & Helman J. (1994). Iris performer : A high performance multiprocessing toolkit for real-time 3d graphics, *ACM SIGGRAPH*, pp. 381–393.
- Rummery, G. A. & Niranjana, M. (1994). On-line q-learning using connectionist systems, *Technical report*, Cambridge University Engineering Department.
- Sakuma, J., Kobayashi, S. & Wright, R. N. (2008). Privacy-preserving reinforcement learning, *Proceedings of the 25th international conference on Machine learning*, ACM, Helsinki, Finland, pp. 864–871.
URL: <http://portal.acm.org/citation.cfm?id=1390265>
- Salber, D., Dey, A. K. & Abowd, G. D. (1999). The context toolkit : aiding the development of context-enabled applications, *Proceedings of the SIGCHI conference on Human factors in computing systems : the CHI is the limit*, ACM, Pittsburgh, Pennsylvania, United States, pp. 434–441.
URL: <http://portal.acm.org/citation.cfm?id=303126>
- Salembier, P. & Zouinar, M. (2004). Intelligibilité mutuelle et contexte partagé. inspirations théoriques et réductions technologiques, *@CTIVITES* 1(2).
- Salovaara, A. & Oulasvirta, A. (2004). Six modes of proactive resource management : a user-centric typology for proactive behaviors, *Proceedings of the third Nordic conference on Human-computer interaction*, ACM, Tampere, Finland, pp. 57–60.
URL: <http://portal.acm.org/citation.cfm?id=1028022>
- Schilit, B., Adams, N. & Want, R. (1994). Context-aware computing applications, *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, pp. 85–90.
- Schilit, B. & Theimer, M. (1994). Disseminating active map information to mobile hosts, *Network, IEEE* 8(5) : 22–32.
- Schmidt, A. (2005). Interactive Context-Aware systems interacting with ambient intelligence, in G. Riva, F. Vatalaro, F. Davide & M. Alcaniz (eds), *Ambient Intelligence*, IOS Press, chapter 9, p. 159–178.

- Schmidt, A., Beigl, M. & w Gellersen, H. (1998). There is more to context than location, *COMPUTERS AND GRAPHICS* **23** : 893—901.
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.2933>
- Shell, J. S., Selker, T. & Vertegaal, R. (2003). Interacting with groups of computers, *Commun. ACM* **46**(3) : 40–46.
- Shell, J. S., Vertegaal, R., Cheng, D., Skaburskis, A. W., Sohn, C., Stewart, A. J., Aoudeh, O. & Dickie, C. (2004). Ecslasses and eyepliances : using attention to open sociable windows of interaction, *ETRA '04 : Proceedings of the 2004 symposium on Eye tracking research & applications*, ACM, New York, NY, USA, pp. 93–100.
- Shi, Y., Xie, W., Xu, G., Shi, R., Chen, E., Mao, Y. & Liu, F. (2003). The smart classroom : merging technologies for seamless tele-education, *Pervasive Computing, IEEE* **2**(2) : 47–55.
- Sigaud, O. & Gérard, P. (2005). L'apprentissage par renforcement indirect dans les systèmes de classeurs, *Journal électronique d'intelligence artificielle* **4**(19).
URL: <http://jedai.afia-france.org/detail.php?PaperID=19>
- Smart, W. D. & Kaelbling, L. P. (2002). Effective reinforcement learning for mobile robots, *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2002)*, Vol. 4, p. 3404–3410.
URL: <http://www.cse.wustl.edu/wds/papers.php?detail#icra2002>
- Stanfill, C. & Waltz, D. (1986). Toward memory-based reasoning, *Commun. ACM* **29**(12) : 1213–1228.
URL: <http://portal.acm.org/citation.cfm?doid=7902.7906>
- Strauss P.S. & Carey R. (1992). An object-oriented 3d graphics toolkit, *ACM SIGGRAPH*, pp. 341–347.
- Sugeno, M. (1977). *Fuzzy Automata and Descision Process*, Amsterdam, Holland, chapter Fuzzy Measures and Fuzzy Integrals, pp. 89–102.
- Sutton, R. (1987). Implementation Details of the TD(λ) Procedure for the Case of Vector Predictions and Backpropagation, *Technical Report TN87-509.1*, GTE Laboratories.
- Sutton, R. (1988). Learning to Predict by the Methods of Temporal Differences, *Machine Learning* **3** : 9–44.
- Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting, *SIGART Bull.* **2**(4) : 160–163.
URL: <http://portal.acm.org/citation.cfm?id=122377>
- Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning : An Introduction (Adaptive Computation and Machine Learning)*, The MIT Press.
- Tahani, H. & Keller, J. (1990). Information fusion in computer vision using the fuzzy integral, *IEEE Transactions on Systems, Man and Cybernetics* **20**(3) : 733–741.
- Thomaz, A. L., Hoffman, G. & Breazeal, C. (2006). Reinforcement learning with human teachers : Understanding how people want to teach robots, *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, p. 352–357.

- URL:** http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4107833
- Varela, F. J., Thompson, E. & Rosch, E. (1992). *The embodied mind*, MIT Press.
- Vaufreydaz, D., Emonet, R. & Reignier, P. (2006). A lightweight speech detection system for perceptive environments, *3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Washington DC, USA.
- URL:** <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/VER06/>
- Vertegaal, R., Mamuji, A., Sohn, C. & Cheng, D. (2005). Media eyepliances : using eye tracking for remote control focus selection of appliances, *CHI '05 : CHI '05 extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, pp. 1861–1864.
- Walker, M. A. (2000). An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email, *Journal of Artificial Intelligence Research* **12** : 387–416.
- Want, R., Hopper, A., Falcão, V. & Gibbons, J. (1992). The active badge location system, *ACM Trans. Inf. Syst.* **10**(1) : 91–102.
- URL:** <http://portal.acm.org/citation.cfm?id=128756.128759>
- Ward, A., Jones, A. & Hopper, A. (1997). A new location technique for the active office, *Personal Communications, IEEE* **4**(5) : 42–47.
- Warren & Brandeis (1890). The right to privacy, *Harvard Law Review* **4** : 193–220.
- URL:** http://groups.csail.mit.edu/mac/classes/6.805/articles/privacy/Privacy_brand_warr2.htm
- Watkins, C. & Dayan, P. (1992). Q-Learning, *Machine Learning* **8**(3-4) : 279–292. ISSN 0885-6125.
- Weiser, M. (1991). The computer for the 21st century, *Scientific American* .
- URL:** <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>
- Xie, W., Shi, Y., Xu, G. & Xie, D. (2001). Smart classroom - an intelligent environment for tele-education, *Advances in Multimedia Information Processing — PCM 2001*, pp. 662–668.
- URL:** http://dx.doi.org/10.1007/3-540-45453-5_85
- Zadeh, L. (1965). Fuzzy Sets, *Information and Control* **8** : 338–353.
- Zelevnik R.C., Conner D.B., Wlocka M.M., Aliaga D.G., Wang N.T., Hubbard P.M., Knepp B., Kaufman H., Hugues J.F. & van Dam A. (1991). An object-oriented framework for the integration of interactive animation techniques, *ACM SIGGRAPH*, pp. 105–112.
- Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I. & Lathoud, G. (2004). Multimodal group action clustering in meetings, *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*, ACM, New York, NY, USA, pp. 54–62.
- URL:** <http://portal.acm.org/citation.cfm?doid=1026799.1026810>

Quatrième partie

Annexes

A.1 Avant Propos

Mon travail de recherche en thèse portait sur la navigation réactive pour la robotique mobile. J'ai été nommé Maître de Conférences en 1995 à l'Ecole Nationale d'Ingénieurs de Brest. Je me suis alors consacré à la Réalité Virtuelle et plus particulièrement aux environnements de développement pour le prototypage interactif dans le cadre de la réalité virtuelle. J'ai été muté en 2000 à l'Université Joseph Fourier de Grenoble. J'ai co-encadré au cours de mon passage à Brest la thèse de Fabrice Harrouet dont les éléments principaux se retrouvent dans l'article ci-dessous, publié lors des journées Françaises de l'Informatique Graphique (AFIG) à Rennes, en 1997.

A.2 Introduction



thèse de
Fabrice
Harrouet

Notre étude se situe dans le cadre d'un nombre important de recherches visant à l'élaboration de boîtes à outils pour la réalité virtuelle et la simulation distribuée (MR Toolkit ([Green M. et al., 1993](#)), VR-DECK ([Codella C.F et al., 1993](#)), DIVE ([Carlsson C. & Hagsang O., 1993](#)), NPSNET/DI ([Brutzman D.P, 1994](#)), ([Macedonia M.R., 1995](#))), ou l'animation et l'interaction 3D (UGA ([Zeleznik R.C. et al., 1991](#)), VB-II ([Gobbetti E., 1993](#)), Virtual Studio ([Balaguer J.F., 1993](#))).

En parallèle à ces études, depuis ces dernières années, un effort important a été réalisé par les constructeurs et les éditeurs pour améliorer les algorithmes de rendu et élaborer des moteurs graphiques efficaces. Certains produits commerciaux sont présents sur le marché : World Toolkit (Sense8, 1991), dVISE (Division) ou Clovis (MEDIALAB, 1995).

En normalisation, nous assistons à l'émergence d'un standard de description d'applications de réalité virtuelle : V.R.M.L. 2.0 [ISO/IEC WD 14772] ([Hartman J. & Wernecke J., 1996](#)), même si les aspects comportementaux ou distribués ne sont pas encore réelle-

ment pris en compte. Côté communication, la prochaine version du protocole IP (IP v6) [RFC 1883 (décembre 1995)] introduit la notion de flot et inclut en natif le multipoint, essentiel en réalité virtuelle distribuée.

De ce fait, une proposition pour une nouvelle architecture de boîte à outils pour la réalité virtuelle se doit d'être très largement ouverte et le plus possible indépendante des bibliothèques de rendu pour pouvoir utiliser les solutions les plus optimales (Roohlf J. & Helman J., 1994), (Strauss P.S. & Carey R., 1992), elle devra pouvoir décrire des environnements de plus en plus larges, comportant des agents aux comportements de plus en plus complexes.

La boîte à outil Arévi, résultat de nos recherches, est une plate-forme logicielle de développement d'applications de réalité virtuelle distribuée. Ecrite en C++, elle possède un noyau indépendant des bibliothèques 3D. Elle peut être utilisée pour développer rapidement des applications de visualisation 3D "temps réel", avec ou sans immersion des opérateurs humains, des applications de travail coopératif ou des applications de réalité virtuelle distribuée intégrant des univers peuplés d'entités aux comportements plus ou moins complexes.

Nous détaillerons plus particulièrement au cours de cet article les aspects comportement de notre plate-forme. Après avoir indiqué les contraintes propres à la réalité virtuelle, nous présenterons plus en détail Arévi : son architecture logicielle ainsi que le langage multi-agents Oris autour duquel elle est articulée. Nous présenterons ensuite un exemple illustrant les possibilités du système. Nous concluons sur les perspectives d'évolution de cette plate-forme.

A.3 Les besoins d'un système de réalité virtuelle

Une application de réalité virtuelle distribuée de type travail coopératif a pour objectif de permettre à un groupe de personnes distantes de travailler ensemble sur un projet commun (prototypage interactif par exemple). Afin de laisser les utilisateurs se concentrer sur le problème qu'ils cherchent à résoudre, il faut que le système qu'ils utilisent soit le moins contraignant possible : il ne faut pas qu'ils s'adaptent à l'outil, il faut que l'outil soit capable d'accepter tout ce qu'ils ont besoin de faire.

A.3.1 Modularité

La notion de prototypage interactif implique au minimum la possibilité d'ajouter, de retirer ou de déplacer des éléments dans le système. Il faut donc que le fonctionnement de l'ensemble du système dispose d'une certaine souplesse afin de supporter des modifications substantielles de sa structure.

L'approche multi-agents semble la plus appropriée dans l'accomplissement de cet objectif. Elle focalise l'attention du concepteur d'un système sur le fonctionnement individuel de chacun de ses composants. Ceux-ci doivent être dotés de moyens de percevoir et d'agir sur leur environnement (les autres composants du système). Si l'on dispose d'un moyen de décrire le fonctionnement interne d'un composant ainsi que ses interactions avec son environnement, alors celui-ci pourra participer au fonctionnement global d'un système constitué d'éléments conçus de la même façon. Ce fonctionnement global sera le résultat des actions conjuguées de chacun des composants du système et variera selon les interventions de l'utilisateur qui manipulera à sa guise les divers composants.

A.3.2 Immersion par le langage

La différence principale entre la simulation et la réalité virtuelle est que tout ne peut être prévu lors du lancement d'une session. Une personne à l'origine d'un monde virtuel aura en effet prévu de le peupler avec certaines catégories d'entités, aura également prévu les interactions possibles. Tout utilisateur se connectant doit le percevoir mais ne doit pas être restreint à cela. Il faut lui donner la possibilité d'apporter en cours d'exécution ses propres entités, même si elles étaient inconnues du concepteur initial. Cet apport d'entités doit pouvoir être fait sans avoir besoin d'arrêter le système et de tout reconfigurer. De la même manière, un utilisateur peut avoir besoin pour sa tâche d'interagir avec l'univers d'une manière non prévue par le concepteur. Il faut qu'il ait la plus grande liberté d'action et d'interaction possible. Tout ce qu'un agent virtuel peut faire, il faut qu'un opérateur humain puisse également le faire afin d'être au même niveau qu'eux et avoir la même liberté qu'eux. L'opérateur doit parler le langage des agents (on parle alors d'immersion par le langage). Ce langage doit donc être dynamique, c'est à dire accepter des requêtes et des modifications en cours d'exécution.

A.3.3 Généricité

Les applications de la réalité virtuelle sont multiples et concernent des domaines d'activités très éloignés les uns des autres (par exemple : ergonomie des locaux techniques, étude de populations d'insectes, des mouvements de foules, navigation dans un site archéologique reconstitué, chirurgie assistée par ordinateur, etc). Il est important que le système utilisé n'intègre pas de contraintes sur le type d'univers.

En résumé, intégrer dans une plate-forme de réalité virtuelle un langage multi-agents, dynamique et générique permet d'obtenir un outil laissant une grande liberté aux utilisateurs quant aux types d'entités qu'ils souhaitent mettre en œuvre et sur les interactions qu'ils souhaitent réaliser.

A.4 ARéVi 4.0

Nous allons présenter au cours de cette section la plate-forme ARéVi. Cette plate-forme est le résultat de nos travaux de recherche.

A.4.1 Présentation générale

ARéVi repose sur les concepts *d'entités*, de *scènes* et de *viewers*.

Les entités sont des agents repérés dans l'espace et possédant une représentation en 3 dimensions. Cette représentation inclut les animations (plusieurs représentations se succédant dans le temps) ainsi que les niveaux de détails (réduction du nombre de facettes en fonction de la distance à la caméra). Les entités peuvent ensuite être dérivées (au sens de la programmation objet) en des caméras, des lumières ou tout autre type propre à l'utilisateur.

Afin de réduire les coûts d'affichage, les entités sont regroupées en scène : lorsqu'on est dans une scène, ne sont visualisées que les entités en faisant partie.

Les entités d'une scène sont perçues par une caméra présente dans celle-ci (soit elle est fournie par l'utilisateur, soit elle est créée par défaut). Les images prises par cette caméra sont envoyées à un viewer (fenêtre 2D) pour affichage.

L'ensemble de ces points est illustré figure A.1.

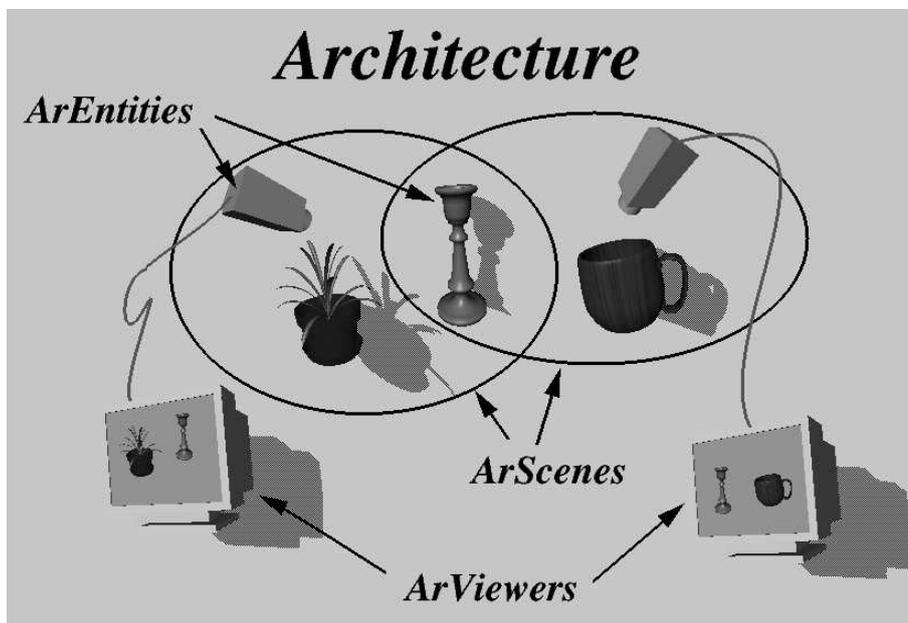


FIGURE A.1: Principe d'ARéVi 4.0

A.4.2 Architecture logicielle

Le noyau d'ARéVi se compose d'un ensemble de classes C++. La figure A.2 représente une partie du modèle OMT associé. L'ensemble des fonctionnalités du système, à

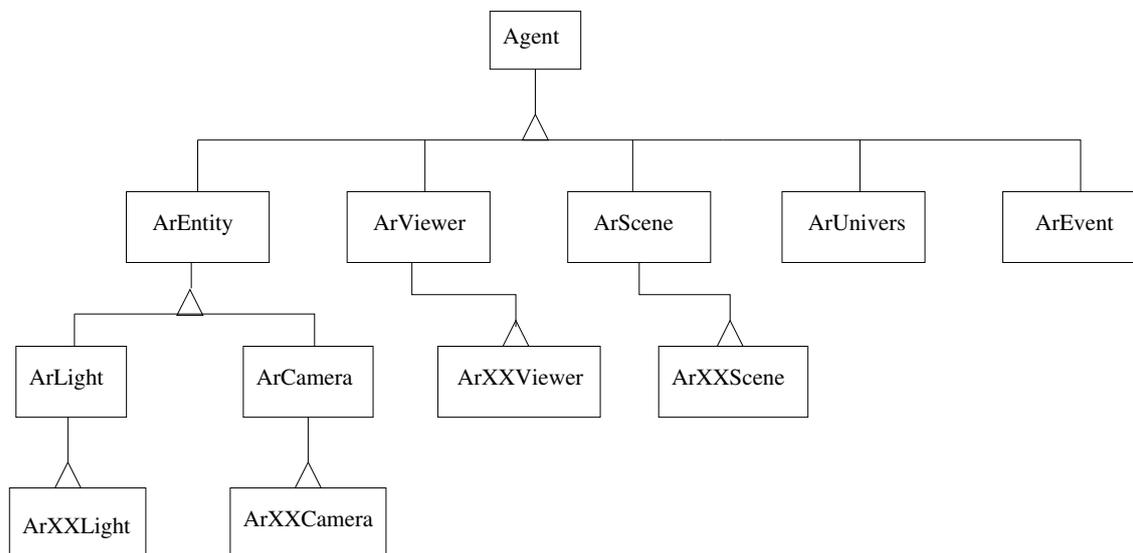


FIGURE A.2: Extrait du diagramme OMT du noyau d'ARéVi

l'exception de l'affichage, a été réalisé sous la forme de classes abstraites ou non. On retrouve en particulier les classes rattachées aux notions d'entité et ses dérivées (**ArEntity**, **ArLight**, **ArCamera**), de scène (**ArScene**) et de viewer (**ArViewer**). La dérivation commune à partir de la classe **Agent** sera expliquée lors de la section A.4.3.

Une **ArEntity** possède une géométrie (**ArGeometry**) chargée de gérer la position dans l'espace ainsi que la cinématique de l'entité associée.

Le lien avec une bibliothèque particulière de rendu est réalisé par dérivation des classes concernées. Par exemple, une **ArCamera** sera dérivée en **ArXXCamera** où **XX** représente la librairie concernée. Actuellement, deux librairies sont supportées : OpenInventor (SGI) et Vision (Ilog). Le portage sous OpenGL est en cours de réalisation.

A.4.3 Le langage oRis

Comme nous l'avons indiqué lors de la section A.3, le formalisme multi-agents ainsi que les caractéristiques dynamiques et la généralité d'un langage sont des propriétés très intéressantes dans le cadre d'un système de réalité virtuelle.

Parmi les langages interprétés (pour les aspects dynamiques) existants, nous considérerons deux catégories :

1. Les langages multi-agents.
2. Les langages “généralistes” (et plus particulièrement les langages objets).

A.4.3.1 Les langages multi-agents

La plupart des systèmes multi-agents existants sont des systèmes développés pour une application très précise. Il en existe néanmoins quelques-uns de plus généraux. Nous allons décrire brièvement deux d’entre eux.

Starlogo ([Resnick, 96](#)) est un simulateur et un langage permettant de décrire un univers multi-agents. Chaque agent de cet univers est équivalent à une tortue Logo dont le comportement est facilement spécifié par l’utilisateur à l’aide du langage. Ces agents évoluent dans un univers plan matérialisé par des cases pouvant elles-mêmes avoir un comportement semblable à celui des tortues (mis à part la faculté de se déplacer). Cet outil a permis entre-autre de simuler la circulation routière, des colonies de termites, la collecte de nourriture par les fourmis, l’agrégation des cellules, les éco-systèmes proie/prédateur. L’objectif d’un tel outil est de mettre en évidence une autre manière de penser les problèmes informatiques (approche massivement parallèle). Les simulations à l’aide de Starlogo mettent en œuvre quelques centaines voire milliers d’agents.

L’environnement HOMAGE ([Poggi & Adorni, 1996](#)) permet de décrire des systèmes multi-agents en mettant en œuvre différents langages adaptés à chaque niveau de description. En ce qui concerne la prise de décision, le langage ALL à base de règles permet de décrire le comportement des agents alors que des langages comme C++, CommonLisp et Java permettent de décrire le déroulement bas niveau de chaque procédure élémentaire de comportement. La communication entre agents peut être synchrone ou asynchrone mais un agent ne peut pas contrôler complètement un autre (le détruire par exemple). Les applications connues de ce langage ne mettent malheureusement en œuvre qu’un très petit nombre d’agents.

A.4.3.2 Les langages généralistes

Les langages généralistes par définition sont suffisamment ouverts pour permettre de représenter n’importe quel univers. Si le langage inclut les concepts objets, il est alors relativement simple de passer au niveau agent en reprogrammant un “ordonnanceur” chargé de réveiller de manière cyclique chaque objet devenu agent.

Ils présentent pour certains (Java en particulier) des propriétés dynamiques intéressantes. Il est ainsi possible à tout instant, sans arrêter l’interpréteur, de recevoir une portion de code à travers le réseau et de l’exécuter. Ce code peut contenir des actions (activation de méthodes) mais également la définition de nouvelles classes ou la surdéfinition de méthodes de classes déjà existantes. Cet aspect dynamique du langage nous permettrait en partie de réaliser ce que l’on souhaite : modifier le comportement d’entités ou ajouter de nouvelles entités en cours d’exécution. La granularité de ces actions se situe au niveau classe mais nous souhaiterions, pour plus de liberté, avoir une gra-

nularité au niveau instance : on veut redéfinir une méthode non pas au niveau de la classe et donc pour toutes les instances simultanément mais pour une instance seule qui en quelque sorte se détache un peu de sa classe d'origine. Considérons par exemple un univers virtuel constitué d'une route et de voitures faisant la course sur cette route. Les voitures peuvent être des instances d'une même classe fournie par le concepteur de cet univers. Une autre personne se connectant à la session peut observer l'évolution de la course et la juger peu satisfaisante. Il a alors la possibilité, sans arrêter le déroulement, de choisir une voiture, de proposer pour cette voiture seulement un nouvel algorithme de conduite et d'observer comment son véhicule se comporte par rapport aux autres. Si la nouvelle approche est jugée meilleure, l'algorithme de pilotage peut alors passer du statut de méthode d'une instance au statut de méthode de classe afin que toutes les instances en profitent. Un autre exemple d'utilisation de la surdéfinition de méthodes d'instance sera fourni section [A.5.3](#).

Les langages cités précédemment ne répondant pas complètement à nos contraintes, nous avons développé notre propre langage : le langage oRis ([Harrouet et al., 1997](#)).

A.4.3.3 Caractéristiques principales d'oRis

oRis est un langage multi-agents dynamique. C'est d'abord un langage objet : utilisation de classes regroupant des attributs et méthodes. La syntaxe est proche du C++. Il intègre les notions d'héritage multiple. C'est également un langage agent : tout objet possédant une méthode `void main(void)` devient un agent. Cette méthode est exécutée cycliquement par l'ordonnanceur du système et comprend donc le comportement de l'entité.

oRis est un langage dynamique : il est capable en cours d'exécution d'accepter un nouveau code, de définir une nouvelle classe, de surdéfinir des méthodes au niveau de la classe mais également au niveau de l'instance. Il est enfin possible d'ajouter au niveau d'une instance la définition d'une nouvelle méthode n'existant pas au sein de la classe d'origine.

oRis permet un couplage profond avec le C++ : le programmeur peut spécifier une correspondance entre une classe oRis et une classe C++, l'objet C++ représentant l'agent oRis au sein de l'interpréteur étant celui proposé par l'utilisateur (dérivant de l'objet C++ Agent, objet de base pour les agents oRis). L'appel des méthodes oRis déclarées natives déclenche les méthodes C++ associées.

oRis a été utilisé avec succès dans des applications aussi diverses que le traitement d'images ([Ballet, Rodin & Tisseau, 1997](#)), l'immunologie ([Ballet, Tisseau & Harrouet, 1997](#)), les réseaux informatiques ([Bourdon, 1997](#)) ou les populations d'insectes ([Martin, 1997](#)).

A.4.3.4 Intégration à ARéVi

Grâce au couplage `oRis` \leftrightarrow C++, ARéVi propose à l'utilisateur des classes `oRis` prédéfinies (méthodes natives) correspondant aux classes C++ exposées section A.4.2 : `ArEntity`, `ArScene`, `ArViewer` ... La figure A.3 représente le diagramme OMT des principales classes prédéfinies. Ce diagramme est le reflet fidèle des classes C++ (figure A.2).

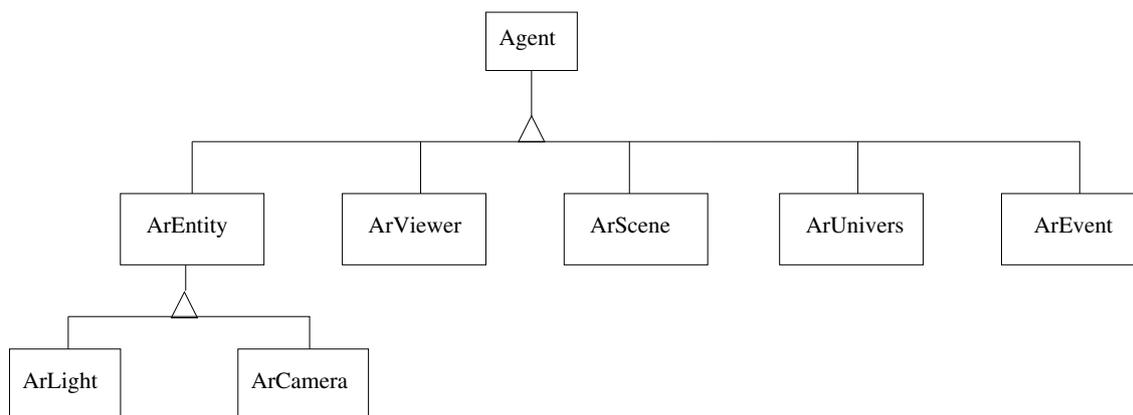


FIGURE A.3: Diagramme OMT des classe prédéfinies `oRis` pour ARéVi

Remarque : nous avons choisi de représenter tous les éléments manipulés par ARéVi sous la forme d'agents. Cela comprend donc non seulement les objets "physiques" (les `ArEntity`) faisant partie de l'univers mais également les structures nécessaires à la mise en place de la simulation. Le principal avantage de cette approche est qu'on peut alors facilement par dérivation leur donner un comportement. Par exemple, l'agent `ArScene` n'est à l'origine qu'une liste d'agents qui sont visualisés lorsqu'on regarde dans cette scène. Il est possible grâce à `oRis` de créer une `ProxyScene`, dérivée d'une `ArScene` fournissant un comportement au travers d'une méthode `main()`. Ce comportement peut être par exemple d'intégrer automatiquement à la scène tout objet virtuel (donc tout agent dérivant d'`ArEntity`) pénétrant dans la zone géographique concernée. De même, un viewer ARéVi (`ArViewer`) a accès au nombre d'images générées par seconde. On peut très facilement créer un viewer dont le comportement est de dégrader l'image générée (fil de fer etc) si la fréquence de rafraîchissement devient trop faible, et donc d'asservir le niveau de rendu à la puissance du moteur graphique.

A.5 Exemples

Nous allons au cours de cette section illustrer les capacités multi-agents et dynamiques d'ARéVi. Nous présenterons tout d'abord la modélisation d'un mécanisme simple. Nous présenterons ensuite une version multi-agents des réseaux de Petri. Enfin, nous montrerons comment on peut très simplement coupler les deux univers précédents grâce à la surdéfinition de méthodes d'instance.

A.5.1 Les trains d'engrenages

Nous souhaitons modéliser un train d'engrenages. Le problème ici n'est pas de réaliser une modélisation physique (contact entre les dents ...) mais une modélisation fonctionnelle : deux engrenages en contact tournent en sens contraire, le rapport des vitesses angulaires étant identique au rapport des rayons.

Le problème peut être abordé d'un point de vue global : le train comporte quatre engrenages : l'engrenage 1 tourne dans le sens des aiguilles d'une montre, le 2 dans le sens inverse etc. Il peut être abordé de manière locale : un engrenage E est un objet (passif). Il connaît son successeur et son prédécesseur dans le mécanisme. Il peut être sollicité (appel de méthodes) par son prédécesseur P . P indique alors à E de combien il vient de tourner ainsi que ses caractéristiques. E a alors pour charge de déduire sa propre rotation et la transmet à son successeur, le processus se répercutant de proche en proche. On crée ensuite un agent moteur (c'est le seul agent de cet exemple) dont le comportement (la méthode `main`) est de tourner. Il ne reste plus qu'à "connecter" le moteur au premier pignon pour que toute la chaîne se mette en marche. Dans le code du destructeur de la classe Pignon, on demande au pignon tué de se déconnecter de son prédécesseur. Ainsi, si en cours d'exécution, un pignon est tué par un utilisateur, cela ne provoquera pas d'erreur (le pignon précédent ne transmet plus de mouvements) et les pignons suivants s'arrêteront automatiquement, n'étant plus sollicités par personne. On obtient donc bien un système dont le comportement est conforme à son équivalent réel.

L'exemple précédent met en œuvre des engrenages passifs, qui sont des objets mais pas des agents. Afin d'obtenir un univers plus réaliste, on peut leur fournir un comportement :

- Scruter l'univers.
- Lorsque qu'un engrenage E est détecté à proximité (la somme de son rayon et du rayon de E = la distance séparant les deux entités) :
 - Se connecter à E
 - ⇒ On se met automatiquement à tourner.
- Si on est connecté à un pignon et que la distance à celui-ci devient trop importante
 - Se déconnecter
 - ⇒ On s'arrête automatiquement de tourner.

On peut donc maintenant interactivement en déplaçant les engrenages, à l'aide d'un périphérique d'interface (souris ou autre), modifier le mécanisme alors que celui-ci est en mouvement (voir figure A.4).

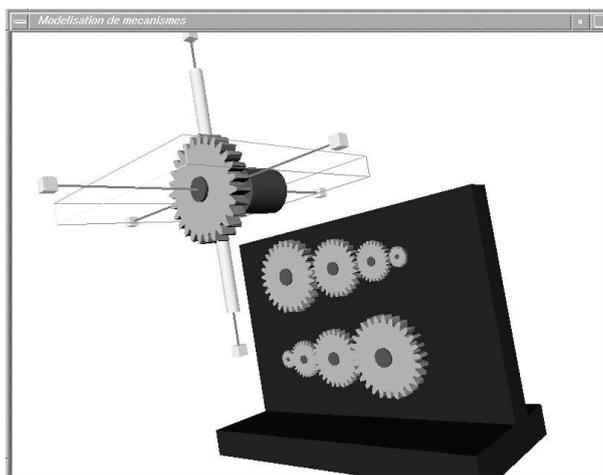


FIGURE A.4: Train d'engrenages dynamique.

A.5.2 Les réseaux de Petri

Le deuxième exemple d'univers que nous allons présenter concerne les réseaux de Petri. Ce type de système ne possède habituellement pas de représentation graphique mais nous lui en avons néanmoins donné une afin de visualiser son fonctionnement (voir figure A.5).

Remarque : il existe plusieurs approches pour la programmation de réseaux de Petri. Utilisant oRis, nous avons opté pour une approche agent : les places les liens et les marques sont des objets, les transitions sont des agents. Elles décident elles-mêmes quand elles doivent s'activer.

A.5.3 Couplage engrenages et réseau de Petri

L'étape suivante consiste maintenant à regrouper les deux exemples précédents afin que le réseau de Petri serve de partie commande aux engrenages qui joueront donc le rôle de partie opérative.

Les deux univers ont été conçus séparément et donc ne se connaissent pas. Le couplage est réalisé en trois étapes :

1. Chargement des engrenages.

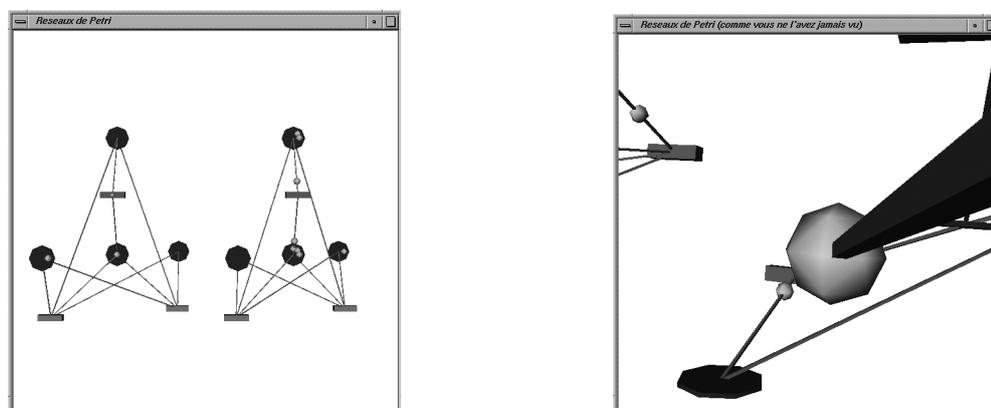


FIGURE A.5: Représentation graphique d'un réseau de Petri

2. Alors que les engrenages fonctionnent, chargement des réseaux de Petri. Les deux univers coexistent en mémoire et possèdent leurs propres fenêtres d'affichage.
3. Déclenchement d'une commande sur une des places du réseau : sur l'entrée d'une marque, on démarre le moteur couplé aux engrenages, sur la sortie d'une marque, on coupe ce même moteur.

Le point 3 est réalisé en entrant dynamiquement (sans arrêter le système) le code suivant (figure A.6) :

Ce code surdéfini les méthodes d'ajout et de suppression de la Place 2. Les méthodes ne sont pas surdéfinies au niveau de la classe (et donc pour toutes les instances) mais uniquement au niveau d'une instance particulière qui donc se singularise et adopte un comportement différent des autres.

Remarque : lors de la surdéfinition d'une méthode d'instance, il est toujours possible d'appeler la méthode de classe (appel par exemple de `Place::add()` dans `Place.2::add()`). Ceci permet très simplement de rajouter une action supplémentaire au comportement par défaut proposé par la classe.

A.5.4 Autres exemples d'applications

Le système ARéVi a été utilisé dans le cadre de contrats industriels :

- Etude de l'ergonomie de locaux techniques : DCN Brest, projet "Charles de Gaulle".
- Visualisation 3D de fonds marins : Ifremer, projet "CARAÏBE 3D".

Surdéfinition de la méthode add de l'instance 2 de la classe Place

```
void Place.2::add(agent mark)
{
  if (!_marks)
  {
    wakeUp(Moteur.1);
  }
  Place::add(mark);
}
```

On démarre le moteur lorsque la place contient au moins une marque

Appel de la méthode add de la classe Place (commune à toutes les instances)

```
int Place.2::del(void)
{
  agent mark = Place::del();
  if (!_marks)
  {
    sleep(Moteur.1);
    return (mark);
  }
}
```

On stoppe le moteur lorsque la place 2 est vide

FIGURE A.6: Association d'un comportement à une place du réseau de Petri

- Représentation 3D interactive du quai Malbert (port de commerce de Brest) : Communauté urbaine de Brest, projet “Images et Aménagement”.

A.6 Conclusion et perspectives

La plateforme ARéVi, architecturée autour du langage oRis, est une plate-forme de développement pour les applications de réalité virtuelle. La présence au cœur du système d’un langage multi-agents dynamique nous permet :

- de concevoir des univers modulaires; la modularité est réalisée par l’absence de contrôleur global et donc par l’utilisation de “briques” élémentaires possédant leurs propres objectifs : les agents,
- d’interagir avec une très grande liberté dans ces univers; les caractéristiques dynamiques du langage oRis nous permettent en tant qu’utilisateur d’agir au plus bas niveau sur les entités présentes (tout ce qu’un agent peut faire, un utilisateur peut le faire), réalisant ainsi une immersion par le langage.

La version précédente de la plate-forme n’incluait pas le langage oRis mais permettait la création d’univers répartis à travers le réseau (Duval et al., 1997). Nous travaillons actuellement sur une version distribuée d’ARéVi / oRis permettant de répartir les agents et donc les univers à travers le réseau. Les communications bas niveaux sont réalisées grâce à Java et supportent les différents modes (Unicast, Broadcast, Multicast). Dans le cadre de notre projet, le choix de ce langage est pénalisant en terme de performances (−20%), mais offre des facilités de développement très intéressantes en termes de simplicité (programmation multi-thread supportée par le langage), de sûreté (ensemble préexistant de classes de communication sécurisée), de développement d’IHM (JavaBeans), d’intégration de traitements multimédia (Java Media Players), d’extension réseau de haut niveau (servlets, RMI, aglets), d’interopérabilité (CORBA) et de connexion aux bases de données (JDBC).



Curriculum Vitae des activités de recherche et d'enseignement

B.1 Formation

- | | |
|------|---|
| 1994 | Thèse de troisième cycle en informatique, spécialité robotique
– Titre : Pilotage réactif d'un robot mobile : étude du lien entre la perception et l'action.
– Laboratoire : LIFIA
– Directeur de Thèse : James L. Crowley |
| 1989 | DEA Informatique (Robotique) sous la direction de James L. Crowley, Laboratoire LIFIA |
| 1989 | Ingénieur ENSIMAG |

B.2 Activité professionnelle

- | | |
|------------------|--|
| à partir de 2000 | Maître de Conférences à l'UFR IMAG, UJF, titulaire d'une PEDR depuis 2005. |
| 1995-2000 | Maître de Conférences à l'Ecole Nationale d'Ingénieurs de Brest (ENIB) |
| 1993-1994 | Attaché Temporaire d'Enseignement et de Recherche (demi poste) à l'ENSIMAG |

B.3 Activité de recherche

B.3.1 Evolutions de ma thématique de recherche

Ma première expérience de recherche (DEA + thèse) a été consacrée à la robotique et plus particulièrement aux systèmes réactifs pour le pilotage de robots mobiles. Je suis Maître de Conférences depuis 1995. J'ai été nommé à l'Ecole Nationale d'Ingénieurs de Brest. J'ai abandonné la robotique pour me consacrer aux systèmes multi-agents pour la réalité virtuelle. J'ai obtenu en Septembre 2000 une mutation à l'Université Joseph Fourier de Grenoble. J'ai rejoint l'équipe Prima du laboratoire GRAVIR (UMR 5527). J'ai réalisé dans ce cadre une troisième conversion thématique. Mon thème actuel de recherche est **la reconnaissance de contexte pour l'informatique ambiante**.

B.3.2 Thématique actuelle

L'espace de travail d'un utilisateur est actuellement constitué d'une station de travail proposant une vaste liste de services (logiciels) indépendants. Cela peut être par exemple l'agenda électronique, le lecteur mail, le navigateur Web ou le traitement de texte. L'utilisateur, en fonction de sa tâche, choisit le logiciel qu'il souhaite utiliser et dialogue avec lui en utilisant le clavier et la souris. Ce schéma classique possède un certain nombre de limitations :

- dans l'évolution technologique actuelle, l'utilisateur ne dispose plus d'une seule station de travail mais peut interagir à l'aide de plusieurs dispositifs hétérogènes, éventuellement partageables et utilisables en coopération (PDA, téléphones portables, etc.).
- La communication homme-machine est essentiellement explicite et basée sur le clavier et la souris. Dans le cadre de la communication homme-homme, une grande partie de l'information est transmise sans communication explicite. Cela peut être les expressions faciales, les relations avec les autres personnes ou les objets présents à proximité etc. L'ensemble des éléments permettant de définir cette communication implicite est appelé le contexte.

L'étude des deux points précédents fait partie du domaine de l'informatique ambiante. L'objectif de l'informatique ambiante est de faire disparaître l'informatique traditionnelle au profit d'un espace informatisé au service des activités humaines. Cet espace informatisé doit prendre en compte la multiplicité des plates-formes et percevoir le contexte afin de mieux comprendre et anticiper les besoins de l'utilisateur et être en mesure de proposer automatiquement les services appropriés. Dans le cadre de l'informatique ambiante, mon thème de recherche est :

- la modélisation et la reconnaissance de contexte,
- les assistants virtuels : proposer à l'utilisateur le service approprié en fonction du contexte.

Ce thème est à la frontière de plusieurs domaines tels que la vision par ordinateur, l'analyse du son, l'apprentissage automatique, l'architecture logicielle et les intergiciels.

B.3.3 Participation à des projets de recherche

ANR Casper

Description : Casper vise à développer un système automatique qui suit l'activité à domicile de personnes âgées et/ou fragiles afin de permettre une assistance humaine locale ou distante en vue de favoriser le maintien à la maison.

Date : à partir de Mars 2007

Contribution : Ma contribution au projet concerne :

- l'architecture logicielle,
- l'intergiciel pour la découverte et la mise en relation de services,

Plateforme LIG

Description : l'objectif de la plate-forme LIG est la mise en place d'une infrastructure d'expérimentation et de démonstration pour l'informatique ambiante, fondée sur la mise en commun de certaines plates-formes expérimentales actuelles présentes au sein des différentes équipes du laboratoire. L'objectif est de construire un système fédérateur permettant à chaque équipe intéressée d'aller plus loin dans ses expérimentations en tirant profit des compétences présentes dans les autres équipes.

Date : à partir de 2006

Contribution : ma contribution à la plate-forme est double :

- Je suis responsable du groupe de travail interaction. Ce groupe de travail regroupe les équipes de recherche Getalp, I3D, IAM, IIHM, Metah, Multicom. Nous avons proposé dans ce cadre un environnement de travail collaboratif multi-sites. Cet environnement facilite la mise en oeuvre de réunions ou de séminaires répartis. Il propose également l'enregistrement automatique du contenu de ces événements, leur mise à disposition dans différentes langues, leur indexation (pris en charge par le groupe de travail données) et la mise à disposition via des interfaces plastiques. En tant que responsable du groupe de travail interaction, je suis également membre du comité de pilotage de la plateforme.
- Mise en place d'un système d'enregistrement et de montage vidéo temps réel automatique basé sur la reconnaissance de contexte. Ce système est en cours d'installation de manière permanente dans la salle de réunion 108 de la maison Jean Kuntzmann.

IMAG Mapping

Description : Mapping est un projet inter-équipes au sein de la fédération IMAG. Les équipes membres sont IIHM du laboratoire CLIPS, Adèle du laboratoire LSR, Prima du laboratoire GRAVIR et l'équipe MAGMA du laboratoire Leibniz. Le projet Mapping s'intéresse à la plasticité des interfaces dans le cadre de l'informatique ambiante. La plasticité des interfaces est un domaine multi-compétences. L'objectif est de réunir quelques-unes de ces compétences au sein d'une même plateforme : les IHMs remodelables (IIHM), l'ingénierie dirigée par les modèles permettant de passer de la spécification de tâches à la génération des interfaces correspondantes (Adèle), la planification pour l'assemblage dynamique des composants d'interaction (Magma) et la perception et la reconnaissance de contexte pour le contrôle du remodelage des IHMs (Prima). Ce projet se poursuit actuellement à travers la plateforme LIG.

Date : Mai 2005 – Mai 2007

Contribution : j'ai participé à la définition du projet, la construction des scénarios et la modélisation du domaine. J'ai été responsable pour l'équipe PRIMA de la perception et de la reconnaissance de contexte dans le cadre de séminaires (reconnaissance des phases comme début de présentation, question de l'audience etc).

IST Fame 200028323

Description : Fame signifie Facilitating Agent for Multicultural Exchange. L'objectif du projet est l'analyse automatique et l'interrogation multilingue de documents audio et vidéo enregistrés lors de présentations scientifiques.

Dates : Octobre 2001 – Janvier 2005

Contribution : j'ai été responsable pour l'équipe Prima du cameraman automatique. Le cameraman automatique est un système qui analyse la situation courante et réalise en temps réel un film à partir des caméras et microphones présents dans une salle de conférence. Ce cameraman a été utilisé pour filmer la totalité des sessions du séminaire "Tecnologies de la llengua : darrers avenços" qui s'est tenue à Barcelone du 12 au 16 Juillet 2004. Il a également été utilisé pour filmer et diffuser sur Internet un séminaire Intech qui s'est déroulé à l'INRIA. Ce cameraman est repris dans le cadre de la plate-forme LIG afin d'être installé de manière permanente dans une des salles de réunion.

IST Caviar 200137540u

Description : Caviar signifie Context Aware Vision using Image-based Active Recognition. Le projet a pour objectif de répondre à la question suivante : est-ce que des informations contextuelles sont en mesure d'améliorer les processus de reconnaissance basés sur une description locale des images.

Date : Octobre 2002 – Octobre 2005

Contribution : le consortium Caviar a défini un ensemble de scénarios types. A chacun de ces scénarios est associée une ou plusieurs vidéos représentatives. Ma contribution dans Caviar concerne dans un premier temps la modélisation “manuelle” de ces différents scénarios et, dans un second temps, l’apprentissage automatique de ces modèles à partir des vidéos d’exemple

IST Chil 506909

Description : Computer in the Human Interaction Loop. L’objectif de Chil est d’inclure l’ordinateur dans l’interaction homme-homme. L’ordinateur doit agir en fonction de sa perception des besoins de l’utilisateur. Il doit être en mesure de proposer des services appropriés tout en étant le moins intrusif possible.

Date : Janvier 2004 – Janvier 2007

Contribution : ma contribution dans le cadre du projet Chil concerne la modélisation et la reconnaissance du contexte dans le cadre des “services aux utilisateurs” prévus par le projet. Nous avons également étudié dans un second temps l’apprentissage du modèle de contexte et de l’association Situation courante => Service proposé (thèse d’Oliver Brdiczka).

RNTL Proact : Contact

Description : l’objectif de Contact est d’explorer de nouvelles approches (essentiellement neuronales) pour la détection et la manipulation d’informations contextuelles dans le cadre d’applications pro-actives. Le coordinateur du projet est Xerox Research Center Europe.

Date : Février 2003 - Février 2005

Contribution : je n’ai participé au projet Contact que pendant sa première année pour la mise en place d’un modèle commun de représentation du contexte. Suite à une restructuration de Xerox, le projet a changé de direction et s’est recentré sur les aspects perception et dispositifs innovants pour l’interaction.

Eureka Mithea EU 110

Description : l’objectif du projet Mithra est la conception d’un robot mobile autonome pour la surveillance de locaux industriels ou de surfaces commerciales.

Date : j’ai collaboré au projet en 1990 et 1991

Contribution : ma contribution au projet Mithra concerne les trois aspects suivants :

- développement d’un contrôleur de véhicule.
- planification de chemin : cette planification géométrique s’appuie à la fois sur une connaissance initiale des bâtiments (plans d’architecte) et sur les données provenant de la perception (capteurs ultrasons)
- la navigation : il s’agit de piloter le robot le long du chemin planifié.

B.3.4 Relations industrielles

2004-2005	IBM, Prague dans le cadre du projet CHIL. L'objectif était de porter le moteur de reconnaissance de contexte dans leur environnement de simulation SitCom.
2003	Xerox Research Center dans le cadre du projet Contact. L'objectif était entre autre le transfert auprès de Xerox de notre modèle de contexte. Cet objectif n'a pas été atteint suite à la restructuration de Xerox Research Center.
1998-1999	Compagnie des Signaux pour la réalisation d'un noyau multiagents sur un <i>cluster</i> de machines (Membre du comité de pilotage).
1996-1997	Développement conjoint avec la société Soft Mountain et le S.E.P.T. (centre d'étude de France Télécom) d'une version réactive du langage oRis.
1996-1997	Mise en place pour le compte de la DCN (Direction des Chantiers Navals) d'un prototype d'outil d'aide à la validation ergonomique des locaux techniques (passerelle du porte avion Charles de Gaule).
1992	Consultant (2 mois) auprès de la société ITMI. Conseil sur le choix de l'architecture logicielle d'un robot de transport en milieu hospitalier.
1991	Transfert auprès de la société ITMI du logiciel de modélisation d'environnement et de relocalisation par capteurs ultrasons. Ce transfert a été effectué dans le cadre du projet de recherche Européen Mithra Eureka : EU 110.
1990	Réalisation d'un contrôleur de mobilité pour le robot Kitborg de la société Aleph Technologie.

B.3.5 Encadrement

Encadrement de doctorants

Je suis impliqué dans l'encadrement de 5 thèses :

<i>A partir de Sept 2008</i>	Co encadrement de la thèse de Rémi Barraquand : – <i>Thème</i> : Acquiring Social Common-Sense
<i>Sept 2005 - Oct 2009</i>	Co encadrement de la thèse de Sofia Zaidenberg : – <i>Thème</i> : apprentissage par renforcement du modèle de contexte pour l'informatique ambiante. – <i>Situation actuelle</i> : ATER à l'Université Pierre Mendès-France, Grenoble
<i>Fév 2003 – Déc 2007</i>	Co encadrement de la thèse de Jérôme Maisonnasse – <i>Titre</i> : Estimation des relations attentionnelles dans un environnement intelligent. – <i>Situation actuelle</i> : participe avec 2 autres membres de l'équipe PRIMA à la création d'une entreprise.
<i>Sept 2004 – Mai 2007</i>	Co encadrement de la thèse d'Oliver Brdiczka : – <i>Titre</i> : Learning Situation models for Providing Context-Aware services – <i>Situation actuelle</i> : chercheur chez Xerox Palo-Alto
<i>Sept 2001 – Mars 2008</i>	Co encadrement de la thèse de Suphot Chunwiphot : – <i>Titre</i> : A formal framework for representation and recognition of human activities – <i>Situation actuelle</i> : enseignant-chercheur à l'Université Technologique de Bangkok
<i>Sept 1998 – Déc 2000</i>	Co encadrement de la thèse de Fabrice Harrouet. – <i>Titre</i> : oRis : s'immerger par le langage pour le prototypage d'univers virtuels à base d'entités autonomes – <i>Situation actuelle</i> : Maître de Conférences

J'ai également encadré entre Septembre 2006 et Juin 2007 Marina Pettinari, doctorante de l'Université de Bologne, en visite pour un an dans l'équipe Prima dans le cadre du programme Européen Visitor (FP6 Marie Curie). L'étude menée au sein de notre équipe concerne le couplage du Wifi et de la vision pour permettre l'identification et la localisation de personnes dans une salle de réunion

Encadrement d'ingénieurs

Lors de mon premier poste à l'ENI de Brest, j'ai encadré les projets de fin d'étude (5 mois à temps plein au sein de l'équipe de recherche lors de leur 5^{ème} année d'étude) de 22 élèves ingénieurs.

Encadrement de magistères

- Oct 2001 – Sept 2002* Encadrement du magistère de Laurent Nau :
- *Titre* : Étude de l'utilisation des réseaux de Pétri pour la reconnaissance de scénarios.

Participation à des jurys (Doctorat et Master Recherche)

Remarque : les thèses que j'ai co-encadrées ne sont pas citées dans cette liste.

- Examineur de la thèse de Morgan Veyret : Un Guide Virtuel Autonome pour la Description d'un Environnement Réel Dynamique : Interaction entre la Perception et la Prise de Décision.
- Examineur de la thèse de Vincent Ganneau : Modèle Utilisateur pour la Plasticité des Interfaces Homme-Machine en Mobilité.
- Examineur de la thèse de Romain Bénard : Raisonnement en contexte pour la simulation et l'étude des situations dynamiques collaboratives. Je n'ai pas pu être présent le jour de la soutenance pour des raisons familiales. Le président du jury a lu mes commentaires et mes questions.
- Examineur de la thèse de Frank Wallner : Position Estimation for a Mobile Robot from Principal Components of Laser Range Data.
- Participation à 10 jurys de Master 2 Recherche en tant qu'examineur ou en tant que rapporteur.

B.3.6 Dissémination

Publications

La liste des publications est fournie dans l'annexe **C**

Dissémination auprès du grand public

<i>Barcelone 2004</i>	Le démonstrateur conçu lors du projet Européen FAME a été utilisé en continu pendant 2 semaines à Barcelone lors du mois de juillet 2004. La première semaine a été consacrée à l'enregistrement automatique par le caméraman intelligent du séminaire « Technologies de la llengua : darrers avenços ». L'ensemble des films produits a été automatiquement indexé par les autres partenaires du projet. Le résultat a pu être consulté durant la seconde semaine par les visiteurs de l'exposition internationale de Barcelone 2004.
<i>Fête de la science</i>	Le démonstrateur FAME a également été exposé pendant 2 jours lors de la fête de la science 2004 à Grenoble. Il a été essayé par une centaine de personnes. La première journée était réservée aux étudiants du secondaire et de l'université. La seconde journée était ouverte au grand public.
<i>Informatica ...</i>	Participation entre 1995 et 2000 (lorsque j'étais Maître de Conférences à l'ENI Brest) à divers salons et manifestations grand public organisés par la ville de Brest.

Dépôt logiciel

<i>Tracker 2D</i>	Participation à hauteur de 10% au logiciel de suivi 2D de cible (Robust Tracker) à partir d'un flux d'images provenant d'une caméra vidéo
<i>Omiscid</i>	Participation à hauteur de 20% au développement de l'intergiciel Omiscid. Cet intergiciel multi plateforme et multi langage propose un modèle de composant logiciel pour le développement d'applications en intelligence ambiante. Le mécanisme de découverte de service repose sur DNS-SD. Cet intergiciel est téléchargeable depuis la forge de l'INRIA

En plus de mes collaborations autour de ces deux logiciels déposés, j'ai développé les systèmes suivants (liste non exhaustive) :

- Monica : Monica est un intergiciel pour l'informatique ambiante construit au-dessus d'intergiciels standards : JMS et OSGi. Monica a été utilisé dans le cadre du projet Européen Fame. Il a servi de base pour la nouvelle version : Omiscid.
- Plugin Eclipse OSGI : Extension de l'environnement de développement Eclipse permettant la mise en place d'un nouveau type de projet : le développement d'un bundle OSGi. Cette extension construit automatiquement un squelette de bundle à partir des informations saisies par l'utilisateur.

- Plugin Eclipse Omiscid : Extension de l'environnement de développement Eclipse permettant la mise en place d'un nouveau type de projet : un composant OMIS-CID.
- Plugin Eclipse de description de contexte : Extension de l'environnement de développement Eclipse permettant à un utilisateur de spécifier graphiquement le modèle de contexte associé à son application : les rôles, les relations, les situations et les contraintes temporelles entre les situations. Cette extension s'appuie sur le méta modèle EMF (Eclipse Modeling Framework).
- Compilateur de description de contexte : cette extension Eclipse permet de transformer un modèle de contexte en un code exécutable capable de le reconnaître. Le langage cible est Jess (système à base de règles de type chaînage avant). L'environnement de transformation de modèles utilisé est JET (Java Emitter Templates) du projet Eclipse.
- Cartes cognitives floues : librairie Java permettant la construction et l'utilisation de cartes cognitives floues telles que décrites par Bart Kosko dans son livre Fuzzy Thinking.
- ARÉVI 4.0 et 5.0 : environnement de réalité virtuelle reposant sur un moteur multi-agents pour la programmation du comportements des entités virtuelles autonomes (la version 5.0 est un portage sous Windows de la version 4.0 et s'appuie sur un autre environnement graphique : OpenGL Optimizer (c)).

B.3.7 Review, comités de programmes, comités d'organisation

- 2009 Membre du comité d'organisation d'IHM 2009 (Octobre 2009, Grenoble).
En charge des actes en liaison avec l'ACM.
- 2008 Review d'un article pour IEEE International Conference on Robotics and Automation 2009
- 2006
- Review d'un article sur la logique floue pour la revue International Journal on Expert System
 - Reviewer pour la conférence internationale on Advanced Concepts for Intelligent Vision Systems, , University of Antwerp, Antwerp, Belgium
 - Review d'un article pour la conférence internationale UBICOMP (Orange County, Californie)
- 2005 Membre du comité de programme de l'International Conference on Multimodal Interfaces (ICMI 2005), Trento, Italie.
- 2003 Review d'un article pour la conférence internationale UBICOMP (Seattle)
- 1996-2000 Membre 4 années consécutives du comité de programme du Symposium on Intelligent Robotic Systems
- 1998 Chairman de la session Learning and Control du Symposium on Intelligent Robotic Systems, Edinburgh.
- 1997 Reviewer pour la revue International Journal on Robotics and Autonomous Systems
- 1996 Reviewer pour la revue International Journal on Intelligent Automation and Soft Computing
- 1995 Reviewer pour le congrès IEEE Robotics and Autonomous Systems
- 1994
- Co organisateur et secrétaire de l'International Workshop on Intelligent Robotic Systems (Grenoble)
 - Co organisateur de la session spéciale Learning for Locomotion and Navigation de IEEE Conference on Intelligent Robots and Systems (IROS) qui s'est déroulée à Munich.
 - Secrétaire des trois Workshops vision organisés à ECCV 94 (Stockholm)
- 1993-1995 Secrétaire des réseaux Européen d'excellence ECVNET et HCM Smart

B.3.8 Autre

<i>Fév 1995 – Avr 1995</i>	Visiteur scientifique au département d'intelligence artificielle de l'université d'Edinburgh (dans le cadre du réseau européen HCM Smart)
<i>Déc 1993</i>	Visiteur scientifique au Laboratoire de Recherches Fondamentales en Technologie de l'Université de Varsovie, Pologne. Cette visite a été organisée dans le cadre du réseau Européen HCM Smart

B.4 Activité d'enseignement

B.4.1 Matières enseignées

Au cours de mes 2 postes de Maître de Conférences (ENI Brest et UFR IMA), j'ai assuré des enseignements en formation initiale et en formation continue, de niveau L1 à M2. Les domaines sont les suivants :

- Algorithmique et langages (C, C++, Prolog)
- Compilation
- Intelligence Artificielle
- Commande qualitative
- Réalité Virtuelle
- Vision par ordinateur
- Programmation internet

Parmi ces thèmes, je ne présenterai dans la section suivante que ceux pour lesquels j'ai assuré la mise en place et la responsabilité pédagogique.

B.4.2 Responsabilités d'enseignement

J'ai conçu le contenu et les supports des cours et des TPs présentés dans cette section. Certains TPs sont assurés par d'autres intervenants. Je n'ai pas listé les cours et TP que j'ai repris pour le compte d'autres enseignants et pour lesquels je n'ai pas participé au contenu.

Master 2 Pro Compétences Complémentaires en Informatique

- A partir de 2004* | – Programmation Internet : approches classiques pour le développement de sites Web dynamiques (13.5h de Cours, 13.5h de TD).
 – Projet d'intégration (3 semaines) (Co-responsabilité avec les enseignants de Bases de Données, Programmation Objet et Génie Logiciel).

Master 2 Pro GI

- A partir de 2006* | Architecture des applications Web (MVC + Ajax) dans le cadre du projet ECOM (3h Cours + 3h TP).
A partir de 2007 | Membre de l'équipe système (Java EE, EJBs, Serveur d'application Jonas) pour le support auprès des étudiants du projet ECOM (27h TD).
2002 - 2007 | Technique de programmation Internet (15h Cours + 15h TP)
2001 | Projet Vision par ordinateur : détection et suivi de personnes dans une séquence vidéo

Licence Pro Developpeur Web pour l'Entreprise[ⓐ]

- 2002-2004* | – Base de la programmation : algorithmique + langage C (18h CTD, 21h de TP).
 – Composant des Applications sur Internet (22.5 CTD)
2004 à aujourd'hui | – Conception des applications internet (18h CTD, 18h TP)
 – Intégration des composants logiciels (21h CTD, 21h TP)
2002 à aujourd'hui | – Projet d'intégration (3 semaines) (Co-responsabilité avec les enseignants de Bases de Données et de Composants de Applications Internet)
 – Projet de développement (2 semaines) (Co-responsabilité avec les enseignants de Systèmes Réseaux et de Composants des Applications Internet)

Licence professionnelle : Métiers de l'informatique Image et Vidéo de l'Université Pierre Mendès France

- 2000-2002* | Java 3D : concepts de base (3h Cours, 3h TP)

[ⓐ]. anciennement Technologie des Applications sur Internet

4ème année ENI Brest

1995-2000	– Commande qualitative : logique floue, réseaux de neurones, algorithmes génétiques (24h Cours, 24h TP). Ces cours ont également été donnés à l'Ecole Nationale Supérieure des Télécommunications de Bretagne, au DESS informatique pour l'Automatisation de l'Information et à l'Institut Supérieur d'Electronique de Bretagne. – Proposition de sujets (1 sujet par binômes) et évaluation pour les mini-projets en commande qualitative
1996-2000	Réalité virtuelle (15h Cours, 17h TP) : – programmation de comportement pour les entités autonomes – rendu temps réel : OpenGL
1998-2000	Rendu 3D temps réel : l'environnement Performer (SGI)
1996-2000	Compilation : Flex++ et Bison++

B.4.3 Responsabilité de filière

Sept 2003 - Sept 2004	Co-responsable de licence pro Technologie des Applications sur Internet. J'ai activement participé à la définition de cette filière et à l'élaboration de son dossier d'habilitation.
A partir de Janv 2007	Responsable de la licence pro Développeur Web pour l'Entreprise (ex Technologie des Applications sur Internet)
Sept 2001 - Sept 2007	Responsable de l'option Communication Homme Machine du Master 2 Pro Génie Informatique.

B.4.4 Conférences pédagogiques

Afin de réfléchir sur mes pratiques en tant qu'enseignant, j'ai participé à deux présentations organisées par le *Service Universitaire de Pédagogie* :

18 Déc 2008	Dieudonné Leclercq, l'enseignement en grands groupes
26 Janv 2009	Marcel Lebrun, un regard sur les théories de l'apprentissage : quel impact pour vos pratiques d'enseignement

B.5 Autres responsabilités collectives

B.5.1 Vie de l'UFR

<i>2007-2008</i>	Membre du bureau de l'UFR
<i>2001-2007</i>	Membre élu du conseil d'UFR
<i>2002-2003</i>	Membre du comité d'avancement de l'UFR IMA
<i>A partir de Sept 2004</i>	Responsable de la commission informatique de l'UFR. Cette commission a pour objectif de servir d'interface entre les usagers (enseignants, étudiants, administration) et les moyens informatiques de l'UFR. Cette commission initialement mensuelle a pris un rythme hebdomadaire au cours de l'année 2006-2007 afin de suivre au plus près les différents projets visant à l'amélioration du service aux usagers.
<i>A partir de 2006</i>	Prise en charge de l'introduction du Bureau Virtuel au sein de l'UFR (rédaction de tutoriaux à destination des enseignants, gestion de la création des sites et des droits d'accès, support aux usagers en collaboration avec un des membres des moyens informatiques).
<i>2001</i>	Participation active dans la définition et l'élaboration du dossier d'habilitation de la licence pro Développeur Web pour l'Entreprise (défense du dossier à deux reprises en 2002 et 2007 auprès du ministère de l'Education Nationale).
<i>1997-2000</i>	Membre élu de la commission de spécialiste de l'ENI de Brest (second vice président de cette commission).
<i>2002-2002</i>	Membre extérieur nommé de la commission de spécialistes de l'ENI de Brest
<i>2010</i>	Membre de la commission de sélection d'un poste Mdc 27 à l'IFSIC, Rennes
<i>2010</i>	Membre de la commission de sélection d'un poste Mdc 27 à l'UJF, Grenoble

B.5.2 Vie du laboratoire

<i>2009</i>	Membre du groupe de travail pour la mise en place du rapport d'activité du LIG en vue de l'évaluation AERES.
<i>A partir de 2008</i>	Membre élu du comité du personnel du LIG
<i>2003-2006</i>	Membre nommé du conseil du laboratoire GRAVIR
<i>A partir de 2001</i>	Membre du comité des utilisateurs des moyens informatiques de l'INRIA Rhône Alpes
<i>A partir de 2001</i>	Correspondant du projet PRIMA auprès des moyens informatiques de l'INRIA Rhône Alpes



Liste des publications

International peer reviewed journal [ACL]

2009

Brdiczka, O., Crowley, J. & Reignier, P. (2009). Learning Situation Models in a Smart Home, *IEEE Transactions on System, Man and Cybernetics, part B* **39**(1).

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2009/BCR09/>

Brdiczka, O., Maisonnasse, J., Reignier, P. & Crowley, J. (2009). Detecting Small Group Activities from Multimodal Observations, *International Journal of Applied Intelligence* **30**(1) : 47–57.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2009/BMRC09/>

Zaidenberg, S., Reignier, P. & Crowley, J. (2009). An Architecture for Ubiquitous Applications, *Ubiquitous Computing and Communication Journal (UBiCC)* **4**(2).

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2009/ZRC09/>

2007

Brdiczka, O., Yuen, P. C., Crowley, J. & Reignier, P. (2007). ASiMo : Automatic Acquisition of Situation Models, *eMinds : International Journal on Human-Computer Interaction* **2** : 3–15.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2007/BYCR07/>

Reignier, P., Brdiczka, O., Vaufreydaz, D., Crowley, J. & Maisonnasse, J. (2007). Context-aware environments : from specification to implementation, *Expert Systems : The Journal of Knowledge Engineering* **24**(5) : 305–320.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2007/RBVC07/>

2000

Chevallier, P., Harrouet, F., Reignier, P. & Tisseau, J. (2000). Virtual Reality and Multi-Agent Systems for Manufacturing System Interactive Prototyping, *Special issue on Virtual Prototyping of the International Journal of Design and Innovation Research* .

1997

Reignier, P., Hansen, V. & Crowley, J. (1997). Incremental Supervised Learning for Mobile Robot Reactive Systems, *Robotics and Autonomous Systems* **19**(3-4) : 347–359.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1997/RHC97/>

1995

Reignier, P. (1995). Supervised Incremental Learning of Fuzzy Rules, *International Journal on Robotics and Autonomous Systems* .

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1995/Rei95/>

1994

Reignier, P. (1994). Fuzzy Logic Techniques for Mobile Robot Obstacle Avoidance, *International Journal on Robotics and Autonomous Systems* **12** : 143–153.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1994/Rei94/>

1992

Crowley, J. & Reignier, P. (1992). Asynchronous Control of Rotation and Translation for a Robot Vehicle, *International Journal on Robotics and Autonomous Systems* **10**.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1992/CR92/>

International peer-reviewed conference proceedings [ACT]

2009

Chunwiphot, S., Reignier, P. & Lux, A. (2009). A Formal Fuzzy Framework for Representation and Recognition of Human Activities, *5th IFIP Conference on Artificial Intelligence Applications and Innovations*, Thessaloniki, Greece.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2009/CRL09/>

Crowley, J., Reignier, P. & Barraquand, R. (2009). Situation Models : A Tool for Observing and Understanding Activity, *in Workshop People Detection and Tracking*,

held in *IEEE International Conference on Robotics and Automation*, Kobe, Japan.
URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2009/CRB09/>

2008

Zaidenberg, S., Reignier, P. & Crowley, J. (2008a). Reinforced learning of context models for ubiquitous computing : a ubiquitous personal assistant, in J. Cardoso (ed.), *Proceedings of the 6th ICEIS Doctoral Consortium - DCEIS 2008*, Vol. 6, INSTICC Press, pp. 36–48.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2008/ZRC08a/>

Zaidenberg, S., Reignier, P. & Crowley, J. (2008b). Reinforcement Learning of Context Models for a Ubiquitous Personal Assistant, *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, Vol. Volume 51/2009 of *Advances in Soft Computing*, Springer Berlin / Heidelberg, Salamanca, Spain, pp. 254–264.

2007

Brdiczka, O., Crowley, J. & Reignier, P. (2007). Learning Situation Models for Providing Context-Aware Services, *Proceedings of HCI International*, LNCS, Springer.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2007/BCR07/>

Brdiczka, O., Reignier, P. & Crowley, J. (2007). Detecting individual activities from video in a smart home, *Proceedings of 11th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES)*, Vol. 4692/2008 of *LNCS*, Springer, pp. 363–370.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2007/BRC07/>

Maisonnasse, J., Gourier, N., Reignier, P. & Crowley, J. (2007). Machine awareness of attention for non-disruptive services, *HCI International*, Beijing, China.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2007/MGRC07/>

Zaidenberg, S., Reignier, P. & Crowley, J. (2007). An Architecture for Ubiquitous Applications, in W. Soraya Kouadri Mostéfaoui & Z. Maamar (eds), *The 4th International Joint Workshop on Wireless Ubiquitous Computing (WUC 2007)*, Vol. 1, INSTICC, INSTICC Press, Avenida D. Manuel I, 27A 2? Esquerdo Set?bal Portugal 2910-595, pp. 86–95.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2007/ZRC07/>

2006

Brdiczka, O., Maisonnasse, J., Reignier, P. & Crowley, J. (2006a). Extracting activities from multimodal observation, *Proceedings of 10th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Vol. 2 of *LNCS*, Springer, pp. 162–170.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/BMRC06a/>

Brdiczka, O., Maisonnasse, J., Reignier, P. & Crowley, J. (2006b). Learning individual roles from video in a smart home, *Proceedings of 2nd IET International Conference on Intelligent Environments*, Vol. 1, IET, pp. 61–69.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/BMRC06/>

Brdiczka, O., Reignier, P., Crowley, J., Vaufreydaz, D. & Maisonnasse, J. (2006). Deterministic and Probabilistic Implementation of Context, *IEEE International Conference on Pervasive Computing and Communications Workshops*.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/BRCVM06/>

Brdiczka, O., Vaufreydaz, D., Maisonnasse, J. & Reignier, P. (2006). Unsupervised segmentation of meeting configurations and activities using speech activity detection, *3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI) 2006*.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/BVMR06/>

Brdiczka, O., Yuen, P. C., Zaidenberg, S., Reignier, P. & Crowley, J. (2006). Automatic acquisition of context models and its application to video surveillance, *Proceedings of 18th International Conference on Pattern Recognition (ICPR)*.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/BYZRC06/>

Crowley, J., Brdiczka, O. & Reignier, P. (2006). Learning situation models for understanding activity, *International Conference on Development and Learning, ICDL '06*, IEEE.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/GBR06/>

Emonet, R., Vaufreydaz, D., Reignier, P. & Letessier, J. (2006). O3MiSCID : an Object Oriented Opensource Middleware for Service Connection, Introspection and Discovery, *1st IEEE International Workshop on Services Integration in Pervasive Environments*, Lyon (France).

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/EVRL06/>

Maisonnasse, J., Brdiczka, O., Gourier, N. & Reignier, P. (2006). Attentional Model for Perceiving Social Context in Intelligent Environments, in I. Maglogiannis (ed.), *3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI)*, 3rd IFIP, IFIP, Springer, pp. 171–178.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/MBGR06/>

Maisonnasse, J., Gourier, N., Brdiczka, O., Reignier, P. & Crowley, J. (2006). Detecting Privacy in Attention Aware System, *2nd International Conference on Intelligent Environments*, IET, IET.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/MGBRC06/>

Vaufreydaz, D., Emonet, R. & Reignier, P. (2006). A Lightweight Speech Detection System for Perceptive Environments, *3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Washington DC, USA.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/VER06/>

Zaidenberg, S., Brdiczka, O., Reignier, P. & Crowley, J. (2006). Learning context models for the recognition of scenarios, *3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI) 2006*, Vol. Volume 204/2006 of *IFIP Interna-*

tional Federation for Information Processing, IFIP, Springer Boston, pp. 86–97.
URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/ZBRC06/>

2005

Brdiczka, O., Maisonnasse, J. & Reignier, P. (2005). Automatic Detection of Interaction Groups, *IEEE International Conference on Multimodal Interfaces (ICMI)*, ACM.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2005/BMR05/>

Brdiczka, O., Reignier, P. & Crowley, J. (2005a). Automatic Development of an Abstract Context Model for an Intelligent Environment, Vol. PerCom 05.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2005/BRC05a/>

Brdiczka, O., Reignier, P. & Crowley, J. (2005b). Supervised Learning of an Abstract Context Model for an Intelligent Environment, *Proceedings of Smart Object and Ambient Intelligence Conference (sOc-EUSAI)*.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2005/BRC05/>

Brdiczka, O., Reignier, P. & Maisonnasse, J. (2005). Unsupervised Segmentation of Small Group Meetings using Speech Activity Detection, *ICMI Workshop Proceedings, International Workshop on Multimodal Multiparty Meeting Processing (MMMP)*.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2005/BRM05/>

Metze, F., Gieselmann, P., Holzapfel, H., Kluge, T., Wolfel, M., Rogina, I., Waibel, A., Crowley, J., Reignier, P., Vaufraydaz, D., Bérard, F., Cohen, B., Coutaz, J., Rouillard, S., Arranz, V., Bertran, M. & Rodriguez, H. (2005). The FAME Interactive Space, *2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Edinburgh - UK, p. 4.

2004

Caporossi, A., Hall, D., Reignier, P. & Crowley, J. (2004). Robust Visual Tracking from Dynamic Control of Processing, *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Vol. PETS '04.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2004/CHRC04/>

2003

Crowley, J. & Reignier, P. (2003a). An architecture for context aware observation of human activity, *Workshop on Computer Vision System Control Architectures (VSCA 2003)*, Graz, Austria.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2003/CR03c/>

Crowley, J. & Reignier, P. (2003b). Dynamic Composition of Process Federations for Context Aware Perception of Human Activity, *International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS)*.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2003/CR03b/>

2002

Crowley, J., Coutaz, J., Rey, G. & Reignier, P. (2002a). Perceptual Components for Context Aware Computing, *UBICOMP 2002, International Conference on Ubiquitous Computing*, Goteborg, Sweden.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2002/CCR02/>

Crowley, J., Coutaz, J., Rey, G. & Reignier, P. (2002b). Perceptual Components for Context-Aware Computing, *UbiComp 02, Goteburg, Suède, à paraître*.

2001

Parenthoen, M., Reignier, P. & Tisseau, J. (2001). Put Fuzzy Cognitive Maps to work in virtual worlds, *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, p. P038.

Parenthoen, M., Tisseau, J., Reignier, P. & Dory, F. (2001). Agent's perception and Charactors in virtual worlds : put Fuzzy Cognitive Maps to work, in S. Richir, P. Richard & B. Tarave (eds), *International Conference on Virtual Reality*, Laval, France, pp. 11–18.

Querrec, R., Reignier, P. & Chevaillier, P. (2001). Humans and autonomous agents interactions in a virtual environment for fire-fighting training, in S. Richir, P. Richard & B. Tarave (eds), *International Conference on Virtual Reality*, Laval, France, pp. 57–64.

2000

Nedelec, A., Reignier, P. & Rodin, V. (2000). Collaborative prototyping in distributed virtual reality using anagent communication language, *IEEE International Conference on Systems, Man, and Cybernetic*, Vol. 2, Nashville, USA, pp. 1007–1012.

1999

Chevaillier, P., Harrouet, F., Reignier, P. & Tisseau, J. (1999). Using virtual reality for manufacturing system prototyping, In *Proc. of the International Scientific Workshop "VIRTUAL REALITY AND PROTOTYPING", 7th Workshop of the French Virtual Reality Group GT-RV*, Laval, France.

Harrouet, F., Reignier, P. & Tisseau, J. (1999). Multi-Agent Systems and Virtual Reality for Interactive Prototyping, *International Conference on Information Systems, Analysis and Synthesis*, Orlando, USA.

Reignier, P., Harrouet, F. & Tisseau, J. (1999). Interactive Behavioural Prototyping Using Virtual Reality Toolkit, *Workshop on Mobile Robot*, Zakopane (Poland).

1998

Reignier, P., Harrouet, F., Morvan, S., Tisseau, J. & Duval, T. (1998). ARéVi : A Virtual Reality Multiagent Platform, *First International Conference on Virtual Worlds*, Springer Computer Science, Paris.

1997

Reignier, P., Harrouet, F., Morvan, S., Tisseau, J. & Duval, T. (1997). AReVi 4.0 : Une Plate-forme Multi-Agents de Réalité Virtuelle, *Journée de l'Association Française de l'Informatique Graphique (AFIG)*, Rennes.

1995

Reignier, P., Hansen, V. & Crowley, J. (1995). Incremental Supervised Learning for Mobile Robot Reactive Control, *Intelligent and Autonomous Systems*, Karlsruhe, Germany.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1995/RHC95/>

1994

Dubrawski, A. & Reignier, P. (1994). Learning to Categorize Perceptual Space of a Mobile Robot Using Fuzzy-ART Neural Network, *IROS*, Munich, Germany.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1994/DR94/>

Reignier, P. (1994). Molusc : an Incremental Fuzzy Learning Approach, in J. Crowley & A. Borkowski (eds), *Proc. of the International Workshop on Intelligent Robotic Systems*, Grenoble, France, pp. 178–187.

1993

Reignier, P. (1993). Fuzzy Logic Techniques for Mobile Robot Obstacles Avoidance, *Proc. of the International Workshop on Intelligent Robotic Systems*, Zakopane, Poland.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1993/Rei93/>

1991

Crowley, J., Causse, O. & Reignier, P. (1991a). Layers of Control in Autonomous Navigation, *International Conference on Mechatronics and Robotics*, Aachen, Germany, pp. 67–76.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1991/CCR91/>

Crowley, J., Causse, O. & Reignier, P. (1991b). Layers of Control in Autonomous Navigation, *Proc of the Second International Symposium on Experimental Robotic*,

Toulouse, France.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1991/CCR91a/>

Crowley, J., Reignier, P., Causse, O. & Wallner, F. (1991). Action level behaviours for locomotion and perception, *Experimental Robotics II The 2nd International Symposium*, Springer Berlin / Heidelberg, Toulouse, France.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1991/CRCW91/>

1989

Crowley, J. & Reignier, P. (1989). Dynamic modeling of free-space for a mobile robot, *IEEE/RSJ International Workshop on Intelligent Robots and Systems '89*, pp. 623–633.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2009/CR09/>

Short communications [COM] and posters [AFF] in conferences and workshops

2006

Reignier, P., Zaidenberg, S., Emonet, R., Vaufreydaz, D. & Letessier, J. (2006). jOMiS-CID, un intergiciel sous OSGi pour l'informatique ubiquitaire, *Atelier OSGi, 3e Journées Francophones Mobilité et Ubiquité*, Paris, France.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/RZEVL06/>

Scientific books and chapter [OS]

2009

Zaidenberg, S., Reignier, P. & Crowley, J. (2009). *Pervasive Computing : A Multidimensional Approach*, IT, Icfai Books, chapter An Architecture for Ubiquitous Applications. ISBN : 978-81-314-2425-4.

URL: <http://www.books.iupindia.org/newarticle.asp?isbn=978-81-314-2425-4>

2006

Crowley, J., Reignier, P. & Coutaz, J. (2006). *True Vision in Ambient Intelligence*, Springer Verlag, chapter Designing Context Aware Services for Ambient Informatics, pp. 231—244.

National peer reviewed journal [ACLN]

2006

Brdiczka, O., Reignier, P. & Crowley, J. (2006). Modéliser et faire évoluer le contexte dans des environnements intelligents, *Ingénierie des Systèmes d'Information (ISI), Lavoisier* **11**(5) : 9–34.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/BRC06/>

2005

Tisseau, J., Parenthoen, M., Buche, C. & Reignier, P. (2005). Comportements perceptifs d'acteurs virtuels autonomes : Une application des cartes cognitives floues, *Technique et Science Informatique*.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2005/TPBR05/>

2002

Harrouet, F., Tisseau, J., Reignier, P. & Chevaillier, P. (2002). oRis : un environnement de simulation interactive multi-agents, *Technique et Science Informatiques* **21**(4) : 499–524.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2002/HTRC02/>

1997

Duval, T., Morvan, S., Reignier, P., Harrouet, F. & Tisseau, J. (1997). ARéVi : une boîte à outil 3D pour des applications coopératives, *Calculateurs Parallèles* **9**(2) : 239–250.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1997/DMRHT97/>

National peer-reviewed conference proceedings [ACTN]

2006

Maisonnasse, J., Gourier, N., Brdiczka, O., Reignier, P. & Crowley, J. (2006). Gestion d'applications confidentielles sur la base d'un modèle attentionnel, *Troisièmes Journées Francophones : Mobilité et Ubiquité (UbiMob)*, ACM,IEEE France, ACM.

URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/2006/MGBRC06a/>

1999

Chevallier, P., Harrouet, F., Tisseau, J. & Reignier, P. (1999). oRis : un environnement pour la simulation multi-agents des systèmes manufacturiers de production, *In actes de la 2ème conférence francophone de modélisation et de simulation (MOSIM'99)*, Annecy, France.

1998

Tisseau, J., Reignier, P. & Harrouet, F. (1998). Exploitation de modèles et Réalité Virtuelle, *Actes des 6ème journées du groupe de travail Animation et Simulation*, IRIT, Toulouse (France), pp. 13–22.

1997

Harrouet, F., Cozien, R., Reignier, P. & Tisseau, J. (1997). oRis, un langage pour simulations multi-agents, *Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents*, La Colle-sur-Loup.

Doctoral Dissertations and Habilitations Theses [TH]

1995

Reignier, P. (1995). *Pilotage Réactif d'un Robot Mobile : Etude du Lien entre la Perception et l'Action*, PhD thesis, Institut National Polytechnique, Grenoble.
URL: <http://www-prima.inrialpes.fr/prima/pub/Publications/1995/Rei95a/>

Other Publications [AP]

2005

Barralon, N., Bérard, F., Cohen, B., Borkowski, S., Coutaz, J., Crowley, J., Gieselmann, P., Holzapfel, H., Letessier, J., Metze, F., Reignier, P., Rey, G., Rouillard, S., Tato, R., Tolos, M. & Vaufreydaz, D. (2005). Evaluation of the FAME Interaction Techniques and Lessons Learned, *Technical report*.

1997

Morvan, S., Reignier, P., Tisseau, J. & Le Gal, L. (1997). Visualisation of cartographic data using ARéVi toolkit, *Ilog Visualization Suite International Users Meetings*, Paris.

Reignier, P., Harrouet, F. & Tisseau, J. (1997). oRis, un noyau multi-agents pour ARéVi, *ARéVi 97*, Brest.

1996

Morvan, S., Duval, T., Nedelec, A., Reignier, P., Rodin, V. & Tisseau, J. (1996). Enseignement de la Réalité Virtuelle : une expérience, *4ème journées nationales du Groupe de Travail Réalité Virtuelle du PRC Communication Homme/Machine*, Rennes.

Summary

	1989	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	Total
International peer reviewed journal [ACL]	0	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	2	0	3	10
International peer-reviewed conference proceedings [ACT]	1	3	0	1	2	1	0	1	1	3	1	3	2	2	1	5	11	4	2	2	46
Short communications [COM] and posters [AFF] in conferences and workshops	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
Scientific books and chapter [OS]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2
International peer reviewed journal [ACLN]	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	4
International peer-reviewed conference proceedings [ACTN]	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	4
Doctoral Dissertations and Habilitations Theses [TH]	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Other Publications [AP]	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	1	0	0	0	0	4
Total	1	3	1	1	3	3	1	6	2	4	2	3	3	2	1	7	15	6	2	6	72

