

Université du Québec à Chicoutimi

Mémoire présenté à l'Université du Québec à Chicoutimi
Comme exigence partielle
de la maîtrise en informatique

offerte à

l'Université du Québec à Chicoutimi

par

Amir Ajroud

Un modèle de reconnaissance d'activités temporelles:
extension du modèle de Weida

Novembre 2010

RÉSUMÉ

Le vieillissement de la population représente une préoccupation grandissante des gouvernements en raison de l'ampleur qu'il prendra au cours des prochaines décennies et de la rapidité de son évolution. Une des évolutions préconisées pour répondre à cette transformation de la société concerne l'assistance technologique aux personnes âgées en perte d'autonomie à l'intérieur d'un habitat dit intelligent. L'une des problématiques clé inhérente à l'habitat intelligent émane du besoin intrinsèque, pour fournir un service utile, de reconnaître les activités de la vie quotidienne réalisées par le patient à l'intérieur de son habitat. Cette difficulté correspond en fait à une forme particulière d'une problématique bien connue du domaine de l'intelligence artificielle qui est appelée la reconnaissance de plans. Celle-ci réfère au fait qu'on suppose l'existence d'une structure d'activité (ensemble d'actions organisées dans le temps) planifiée au départ par l'entité observée, et qui constitue le résultat que l'agent observateur cherche à reconnaître à partir de sa base de connaissances (librairie de plans). En effet, l'ensemble des informations collectées reliées au temps comme l'ordonnancement des actions, durée d'une action, laps de temps entre deux actions, etc., se révèle d'une très grande importance afin de minimiser les hypothèses plausibles et identifier les erreurs de réalisation : une activité peut être exécutée de manière erronée si la durée acceptable entre deux étapes du plan est dépassée ou bien insuffisante, de même, le patient peut effectuer deux actions en parallèle d'où l'importance du facteur temps.

Ce mémoire de maîtrise apporte des pistes de solutions à la problématique énoncée, qui porte sur la reconnaissance temporelle des activités de la vie quotidienne d'une personne en perte d'autonomie. Il propose une extension à un modèle de reconnaissance d'activités existant, développé par Weida, en incorporant la notion de concepts d'action en logique de description et en introduisant une nouvelle structuration temporelle augmentée permettant de tenir compte de l'aspect quantitatif lié aux durées et aux délais des actions, qui était absent du modèle original. Cette addition d'une couche temporelle numérique offre un moyen efficace d'exploiter les intervalles de temps dans le processus de reconnaissance et permet d'augmenter la performance et la précision du modèle de Weida. Elle permet également la détection de nouvelles anomalies liées au comportement d'un individu en perte d'autonomie. Enfin, le modèle présenté a été implémenté et validé lors de simulations effectuées à l'aide d'un ensemble de scénarios de cas réels tirés d'une étude clinique. Les résultats ont été analysé et comparé à ceux obtenus avec le modèle original de façon à bien montrer l'apport de notre proposition.

REMERCIEMENTS

En premier lieu, j'aimerais remercier mes directeurs de recherche, M. Bruno Bouchard et M. Abdenour Bouzouane, pour leur soutien moral et financier, l'encouragement, leur grande patience, leur disponibilité, et pour les commentaires pertinents qui m'ont énormément aidé à achever ce projet.

Je tiens à remercier aussi mes parents pour leur aide inconditionnelle et la confiance qui m'ont alloué, sans oublier le support moral de mes frères et de ma sœur, et leur grande patience et les encouragements de ma femme tout le long de ce projet.

Il faut souligner aussi l'appui quotidien des membres du département d'informatique et de mathématique et, plus particulièrement, de tous les professeurs qui m'ont enseigné durant mes études de maîtrise. Un grand merci à mes collègues du groupe de recherche en informatique pour les discussions intéressantes.

Finalement, je remercie tous ceux qui ont contribué à leur façon à la réalisation de ce mémoire.

TABLE DES MATIÈRES

Résumé	II
Remerciements	III
Table des matières	IV
Liste des figures	VII
Liste des tableaux	IX
Liste des algorithmes	X
1 Introduction	1
1.1 Contexte	1
1.2 Problématique : la reconnaissance d'activités et le temps	2
1.3 Illustration de la problématique liée aux erreurs d'origine temporelle	4
1.4 Travaux existants sur la reconnaissance temporelle	5
1.5 Méthodologie de recherche	8
1.6 Organisation du mémoire	10
2 Raisonnement temporel en intelligence artificielle	12
2.1 Introduction	12
2.2 Approche symbolique	14
2.2.1 Logique temporelle d'Allen	14
2.2.2 Représentation du temps basée sur la notion d'instant	23
2.2.3 Comparaison des approches d'Allen et de McDermott	25
2.3 Représentation numérique du temps	26
2.3.1 L'approche temporelle de Dechter	27
2.3.2 Conclusion sur les approches numériques	33
2.4 Approche mixte	34

2.4.1	Approche de Kautz et Ladkin	34
2.4.2	Approche de Meiri	34
2.4.3	Comparaison entre les approches de Kautz et Meiri	35
2.5	Conclusion du chapitre	36
3	La reconnaissance d'activités temporelles	37
3.1	Introduction	37
3.1.1	Définition d'une activité de vie quotidienne (AVQ)	38
3.1.2	Les origines de la reconnaissance d'activités	39
3.1.3	Classification des différents types de reconnaissance d'activités	39
3.1.4	Reconnaissance d'activités dans un environnement intelligent	41
3.1.5	Reconnaissance d'activités intégrant l'aspect temporel	42
3.2	Les principaux travaux sur la reconnaissance temporelle	43
3.2.1	Approche de Kautz	44
3.2.2	Approche de Jakkula et Cook	49
3.2.3	Approche de Weida	54
3.3	Conclusion	59
4	Modèle temporel de reconnaissance de plans: extension du modèle Weida	61
4.1	Introduction	61
4.2	Extension théorique du modèle de Weida	64
4.2.1	Sémantique de l'action en logique de description	65
4.2.2	La subsomption entre les actions	66
4.2.3	Extension numérique des contraintes temporelles de Weida	67
4.2.4	Bilan de notre apport théorique	73
4.3	Implémentation et validation de notre nouveau modèle étendu	74
4.3.1	Architecture logicielle de notre implémentation	75
4.3.2	Le simulateur d'habitat intelligent SIMACT	76
4.3.3	Modèle d'activité de la vie quotidienne utilisée pour les tests	77
4.3.4	Approche expérimentale utilisée pour collecter les données de SIMACT	79
4.3.5	L'application de reconnaissance temporelle	81
4.3.6	Étude comparative des performances du nouveau modèle	86
4.3.7	Présentation des résultats	86
4.4	Positionnement de ce travail au sein du LIARA	90
4.5	Bilan de la contribution de ce mémoire	92
4.6	Conclusion du chapitre	93

5 Conclusion générale	94
5.1 Réalisation des objectifs fixés en introduction	95
5.2 Bilan sur le modèle développé	96
5.3 Limitations et perspectives de développement futur	97
5.4 Bilan personnel sur ce travail de recherche	98
Annexe A : Introduction aux logiques de description	100
Bibliographie	110

LISTE DES FIGURES

2.1	Les différentes approches de modélisation de temps	13
2.2	Les points limites d'un intervalle	14
2.3	Les relations temporelles dans la théorie d'Allen	15
2.4	Graphe temporel selon la théorie d'Allen	18
2.5	Graphe temporel après le calcul de la nouvelle relation	19
2.6	Graphe temporel construit au départ, avant l'application de l'algorithme.....	20
2.7	Graphe temporel après le calcul de la contrainte	21
2.8	Arbre d'états selon de modèle de McDermott	23
2.9	Exemple de graphe temporel selon McDermott	24
2.10	Représentation temporelle à l'aide des TCSP	30
2.11	Un graphe de distance représentant une partie d'un TCSP	31
2.12	Relations de base entre un point P et un intervalle I.....	35
3.1	Exemple de librairie d'activités de cuisine selon Kautz [1991].....	45
3.2	Effet de l'hypothèse de Song et Cohen sur la propagation [1991].....	48
3.3	Un exemple de réseau de contraintes temporelles simples selon Weida	55
3.4	Exemple de subsomption entre un plan de la librairie et un plan observé.....	56
3.5	Le système T-REX.....	58
4.1	Graphe temporel, scénario du poulet mariné.....	61
4.2	Exemple de scénario impliquant les nouvelles contraintes temporelles	67
4.3	Exemple de l'observation de la réalisation de l'activité <i>PréparerSpaghettiAlfredo</i> ..	68
4.4	Relation de subsomption entre un modèle d'activité et un plan observé.....	70
4.5	Classification d'une activité observée dans la taxonomie de modèle d'activités	71
4.6	Identification d'erreurs de types dépassement et d'insuffisance du temps.....	72

4.7	Architecture logicielle de l'implémentation proposée.....	74
4.8	L'outil de simulation d'habitat intelligent SIMACT	76
4.9	Modèle des activités choisis : <i>faire du café et préparer des toasts</i>	78
4.10	L'application de création des plans.....	81
4.11	Un graphe généré à partir d'un modèle de plan.....	83
4.12	Fenêtre de gestion des plans	84
4.13	Proportion d'hypothèses rejetées versus celles identifiées comme plausibles pour le modèle de Weida sans prise en compte du temps	86
4.14	Proportion d'hypothèses rejetées en raison des contraintes temporelles versus les autres rejetées et celles identifiées plausibles pour le modèle original de Weida	87
4.15	Proportion d'hypothèses rejetées en raison des contraintes temporelles numériques versus les autres rejetées et celles identifiées plausibles pour le modèle du mémoire.	88
4.16	Système de reconnaissance d'activités développées par notre équipe. L'image montre l'application utilisée dans les infrastructures du CRHI.....	89

LISTE DES TABLEAUX

2.1	Extrait de la table de transitivité d'Allen	18
2.2	Correspondance entre les relations d'intervalles d'Allen et les contraintes d'instants de McDermott	26
2.3	Longueur des chemins les plus courts de la figure 2.10	32
2.4	Le réseau minimal correspondant à la figure 2.10	32
3.1	Paramétrage de l'ensemble de données utilisé pour trouver des contraintes	51
3.2	Paramétrage et génération de règles dans Weka pour l'ensemble de données réelles ...	52
3.3	Paramétrage et génération de règles dans Weka pour les données synthétiques	52
3.4	Comparaison des prédictions à base d'ActiveLezi avec et sans règles temporelles.....	54
4.1	Moyenne du nombre d'activités possibles au terme du processus de reconnaissance pour chacune des versions du modèle de reconnaissance	86

LISTE DES ALGORITHMES

2.1	Inférence d'une nouvelle relation temporelle dans un graphe	19
2.2	Propagation de contraintes	22
2.3	Algorithme Floyd-Warshall	33
3.1	Analyseur d'intervalle temporel	52
3.2	Prédiction améliorée des règles temporelles	53
4.1	Vérification de subsomption d'un plan de la librairie avec une observation	83

CHAPITRE 1

INTRODUCTION

1.1 Contexte

Le vieillissement de la population représente une préoccupation grandissante pour les gouvernements en raison de l'ampleur qu'il prendra au cours des prochaines décennies et de la rapidité de son évolution [United Nations, 2002]. Une des solutions préconisée pour répondre à cette transformation de la société concerne l'assistance technologique aux personnes âgées en perte d'autonomie à l'intérieur d'un habitat dit *intelligent*. Ce domaine émergeant s'est développé très rapidement en un champ attractif de plusieurs disciplines scientifiques. Il est issu du paradigme de l'intelligence ambiante [Ramos et al., 2008]. Celui-ci réfère à la miniaturisation d'un ensemble de dispositifs électroniques, tels que des capteurs (ex. tapis tactiles, contacts électromagnétiques, détecteurs de mouvements, étiquettes d'identification par radiofréquence, etc.) et des effecteurs (haut-parleurs, écrans, etc.). On intègre ensuite ces capteurs dans n'importe quels objets du quotidien (porte, cuisinière, réfrigérateur, gilet, système d'éclairage, etc.), de manière transparente pour l'occupant. Ces dispositifs sont appelés à communiquer avec un ensemble d'agents artificiels au sens des systèmes multi-agents [Weiss, 2000], dotés de capacités d'ubiquité, de communication et d'intelligence (on parle alors d'agents ambiants) en vue de porter une assistance ponctuelle. Par exemple, si la porte du frigo est ouverte, le frigo doit être capable d'avoir une idée sur le comportement de l'occupant, tel que cette ouverture s'inscrit dans un contexte de préparation d'un repas, en communiquant avec les autres objets de l'habitat, par exemple la cuisinière. L'assistance pour fermer la porte en raison d'un oubli dû à une perte de

mémoire peut se faire soit à travers un envoi de message de rappel (vocal, vidéo) ou soit par une interaction (animation, stimulation, émotion) entre l'agent ambiant associé à la porte du frigo et l'occupant humain, en expliquant la notion de fermeture et la manière d'effectuer cette action en fonction du déficit potentiel de l'individu. Les avancées significatives obtenues dans le domaine de l'intelligence artificielle (IA), nous conduisent aujourd'hui à appliquer ces développements résultants d'une cinquantaine d'années de recherche à cette nouvelle réalité stimulante.

L'une des principales problématiques inhérentes au domaine de l'habitat intelligent émane du besoin intrinsèque, pour fournir un service utile, de reconnaître d'abord les activités de la vie quotidienne réalisées par le patient à l'intérieur de son habitat [Bouchard et al., 2007]. Cette difficulté correspond en fait à une forme particulière d'une problématique bien connue en intelligence artificielle qui est appelée la reconnaissance de plans [Carberry, 2001]. Un plan correspond à une suite d'actions élémentaires représentant une activité donnée. Dans notre contexte applicatif, cette reconnaissance de plans a pour but d'interpréter le comportement d'une personne en vue d'offrir, au moment opportun, les services appropriés sans être rejetés par l'individu. Plus spécifiquement, elle vise à interpréter l'information en provenance des capteurs de bas niveau de manière à identifier l'activité en cours, l'état de compléTION de celle-ci, de déceler si le patient a commis des erreurs liés à sa perte d'autonomie, et de cibler le moment opportun pour effectuer une assistance efficace [Patterson et al., 2003]. L'existence de fonctions de reconnaissance, complétées par d'autres fonctions numériques de captures à travers des senseurs à la fine pointe de la technologie, permettront aux agents ambients de s'organiser dynamiquement pour offrir des services adaptés [Mihailidis et al., 2007]. Ces services utiliseront la notion de contexte, c'est-à-dire des espaces d'informations structurés et dynamiques qui leur permettront de comprendre leur environnement, d'agir de façon pertinente et d'interagir de manière adaptée entre eux et avec les êtres vivants.

1.2 Problématique de ce mémoire: la reconnaissance d'activités et le temps

La reconnaissance d'activités dans un habitat intelligent se décrit comme étant un processus d'inférence, effectué par un agent ambiant en mode d'observation, d'un ensemble de plans possibles (des hypothèses comportementales) concernant un patient observé, tel qu'illustre à la Figure 1.1. Il est basé : (i) sur les observations perçues à travers des capteurs à la suite des interactions (actions) de la personne avec son environnement, (ii) sur un principe de sélection des hypothèses (activités possibles), et (iii) sur une méthode d'appariement des observations avec les plans provenant de la librairie décrivant les activités potentiellement observables [Bouchard et

al., 2007]. Cette librairie est une ontologie décrivant, dans un formalisme logique, les concepts d'actions et d'activités. Le terme *reconnaissance d'activités* réfère ainsi au fait qu'on suppose l'existence d'une structure d'activité (i.e. un ensemble d'actions organisées dans le temps) planifiée au départ par l'entité observée (le patient), et qui constitue le résultat que l'agent assistant cherche à reconnaître à partir de sa base de connaissances (librairie de plans). L'objectif de cet agent consiste, au terme du processus de reconnaissance, à être en mesure de prévoir le plus précisément possible les actions futures du patient en minimisant le nombre d'hypothèses possibles [Roy et al., 2009].

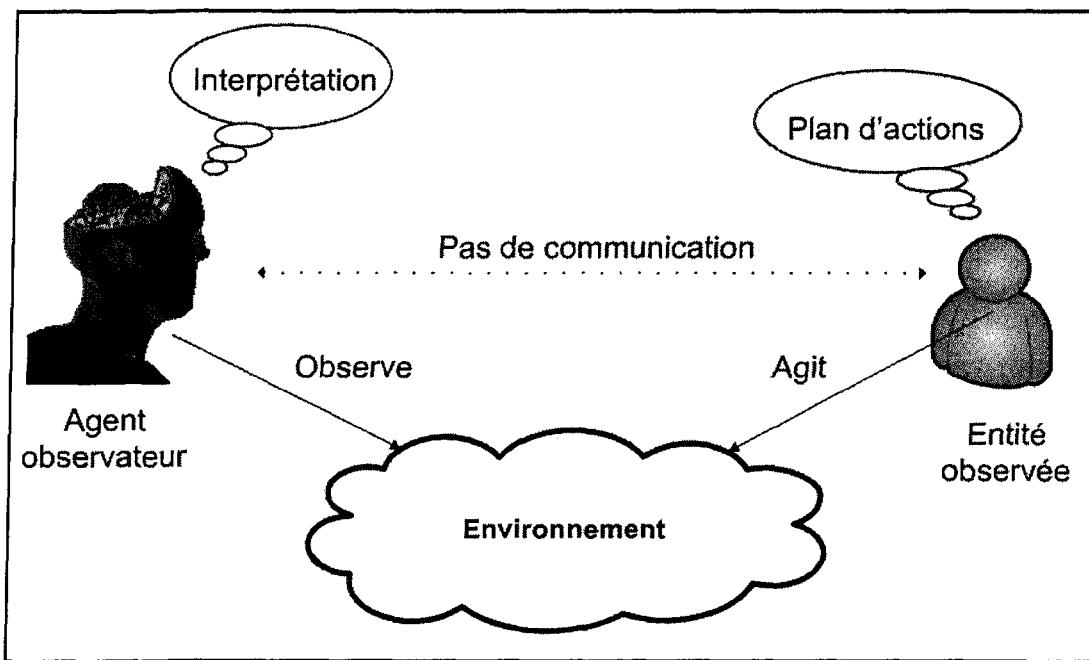


Figure 1.1 : Schéma simple illustrant le processus de reconnaissance [Bouchard, 2006]

Pour atteindre les objectifs de reconnaissance que nous venons de citer, l'ensemble des informations temporelles collectées (durée d'une action, laps de temps entre deux actions, etc.) se révèle d'une importance capitale et souvent sous-estimée [Pollack, 2005], en particulier pour minimiser les hypothèses plausibles et identifier les erreurs de réalisation. À titre d'exemple, on peut supposer deux étapes d'une même activité qui, même si elles sont réalisées dans un ordre adéquat, résulte quand même à une exécution erronée de l'activité si la durée acceptable entre ces deux étapes est dépassée ou bien insuffisante. De plus, le patient peut effectuer deux actions en parallèle, légèrement décalées, ou encore en chevauchement, d'où l'importance du facteur temps.

À partir de ce point, si on veut établir un modèle efficace de reconnaissance d'activités, il faut qu'il soit en mesure de reconnaître l'ensemble des erreurs de réalisation et en particulier celles de nature temporelle [Jakkula & Cook, 2007], qui constituent une nécessité conforme avec la réalité de l'habitat intelligent, surtout avec un patient affecté par un déficit cognitif qui peut perdre la notion du temps causant, par exemple, un incendie. En ce sens, nous proposons dans ce mémoire de se pencher sur cette problématique spécifique combinant reconnaissance d'activités et erreurs d'origines temporelles en fonctions du déficit cognitif d'un patient, dans un objectif d'assistance.

1.3 Illustration de la problématique liée aux erreurs d'origine temporelle

Afin de bien illustrer l'importance d'exploiter l'ensemble des éléments d'information temporels dans notre contexte applicatif de reconnaissance d'activités à l'intérieur d'un habitat intelligent, nous allons utiliser un petit exemple de scénario mettant en évidence les limites d'un système d'assistance cognitive et de reconnaissance du comportement écartant cet aspect. Ce scénario est une extension de celui décrit dans [Bouchard et al., 2007] mettant en relief la problématique du comportement erroné inhérent à un patient souffrant d'Alzheimer.

Prenons le cas fictif d'une personne âgée en perte d'autonomie nommé Pierre, souffrant de la maladie d'Alzheimer à un stade léger (préliminaire) et résidant dans un habitat intelligent doté d'un système d'assistance lui fournissant les services adéquats de support. À l'heure du dîner, Pierre est présent dans la cuisine et un agent artificiel observe toutes ses actions, étant à l'affût d'une opportunité d'assistance. Pierre commence par saisir une boîte de pâtes. D'après la base de connaissances de l'agent (sa librairie de plans), cette action *SaisirPâtes*, qui vient d'être identifiée grâce à une étiquette et une antenne de localisation par radio fréquence (RFID), ne peut correspondre qu'à une seule activité ou intention de la part du patient, soit celle de *CuireDesPâtes*. Cette dernière activité est composée des actions suivantes : *SaisirPâtes*, *FaireBouillirEau* et *MettrePâtesDansEau*. Ces trois actions doivent respectées cet ordre. En considérant que la librairie de l'agent est complète, que la seule activité possible incluse dans celle-ci est *CuirePâtes*, et que Pierre a respecté le séquencement préétabli de son activité, alors la seule conclusion que le système peut tirer est que cette activité s'est déroulée de manière correcte et ne nécessite aucune assistance. Cependant, si on considère le déficit cognitif de Pierre, cette conclusion peut être fausse si l'aspect temporel lié à la durée n'est pas pris en compte. Par exemple, l'agent assistant pourrait observer Pierre exécuter l'action *FaireBouillirEau* pendant 2 heures et 17 minutes. Cette durée est dangereuse et de toute évidence problématique et nécessite une assistance. Dans l'exemple précédent, si le système se contente uniquement de tenir compte

de l'ordonnancement temporelle des actions en faisant abstraction de la durée, il ne pourra inférer que Pierre a réalisé son plan de manière erratique en commettant une erreur que nous qualifierons de *débordement de la durée normal de temps requis pour une action*. Pierre a été en mesure d'organiser l'ensemble d'actions correctement dans le temps, mais son déficit cognitif la rendu incapable de bien estimer la durée du temps que devait prendre une action. Le problème auquel nous faisons face ici est que l'estimation erronée de la durée d'une action observée peut engendrer des conséquences graves.

Prenons maintenant un second exemple dans lequel on suppose que Pierre a entamé l'action *FaireBouillirEau*. Une minute après le début de cette action, l'agent observe Pierre effectuer l'action *MettrePâtesDansEau*. Encore une fois, même si l'ordonnancement des deux actions dans le temps peut être considérée comme respectée, il reste que le délai entre la première action et la seconde s'avère insuffisant pour faire cuire correctement les pâtes car l'eau n'aura pas eu le temps d'être porté à ébullition en une seule minute. À l'instar du premier exemple, le déficit cognitif de Pierre a fait en sorte qu'il a mal estimé le délai requis pour faire bouillir l'eau, commentant cette fois-ci une erreur que nous qualifierons de *délai « incohérent » entre deux actions*. Ainsi, il apparaît évident que pour être apte à détecter ces situations problématiques, le système de reconnaissance doit tenir compte des informations temporelles sur les durées et les délais à l'intérieur de la librairie préétablie de plans, sinon il ne pourra inférer que cette réalisation est incorrect, à la vue de sa base de connaissances.

1.4 Travaux existants sur la reconnaissance temporelle

L'importance de la problématique de la reconnaissance d'activités dans le contexte de l'assistance cognitive à l'intérieur d'un habitat intelligent a entraîné une recrudescence des travaux portant sur cet axe dans les dernières années [Roy et al., 2010]. Malgré cela, on note que la plupart des travaux ciblés dans le domaine des habitats intelligents proposent des modèles de reconnaissance rigides qui ne tiennent pas compte de l'aspect temporel ou encore qui l'incorpore d'une façon très limitée [Singla et al., 2008], réduisant ainsi leur capacité à reconnaître toutes les erreurs d'une personne souffrant de déficit cognitif. Certains modèles de reconnaissance issus du domaine de l'intelligence artificielle, mais arborant un autre contexte applicatif que celui qui nous concerne, ont tenté d'exploiter les contraintes temporelles afin d'améliorer leur performance. En parcourant la vaste littérature disponible, nous avons identifié principalement trois approches de reconnaissance incorporant l'aspect temporel et pouvant nous servir de point

de départ pour l'élaboration d'un modèle adapté à nos besoins et capable de tenir compte de la problématique citée à la section précédente.

La première approche, proposée par Kautz [Kautz, 1991], repose sur un modèle basé sur la logique du premier ordre. Kautz propose de modéliser chaque activité à travers une structure d'arbre organisé en une hiérarchie/décomposition et encodée en logique de premier ordre par une série d'axiomes. La racine de cet arbre correspond à l'objectif (le plan) le plus abstrait, les branches dérivées de cette racine correspondent à des activités spécifiques qui sont définies à l'aide de feuilles correspondant à des actions (événements) observables dans l'environnement. La décomposition hiérarchique repose sur deux relations : la spécialisation et la décomposition. La spécialisation se traduit en une relation de généralité entre deux activités. Par exemple, l'action *FaireDesPâtes* pourrait être définie comme étant une spécialisation de l'action *PréparerUnRepas*. La décomposition permet quant à elle de lier un plan à l'ensemble des actions ou événements nécessaires à son accomplissement. À partir de là, le processus de reconnaissance, qui est déductif et purement logique, est quant à lui défini par un ensemble de règles d'inférence s'appuyant sur la théorie de la circonscription de McCarthy [Kautz, 1986]. Dans ce modèle, la prise en compte de l'aspect temporel est fondée sur l'introduction, dans la définition formel d'un plan, de contraintes temporelles d'ordonnancement (*before*, *after*, *meet*, etc.) entre chaque paire d'actions formant le plan. Ces contraintes sont formalisées à l'aide de la logique temporelle d'Allen [Allen, 1983]. Un algorithme de propagation de contraintes est incorporé au système d'inférence de façon à ce qu'à chaque observation d'un nouvel événement, on puisse vérifier si les contraintes d'ordonnancement temporelles sont satisfaites afin d'éliminer certains plans candidats. La principale faiblesse de cette approche est qu'elle tient uniquement compte de l'aspect qualitatif des contraintes temporelles liées à l'ordonnancement des actions dans l'activité, en négligeant l'aspect quantitatif lié aux durées des actions et aux délais entre ces actions, qui permet de cerner plus efficacement les erreurs cognitives.

Le deuxième modèle recensé, proposé par Jakkula et Cook [Jakkula et Cook, 2007], provient directement du domaine applicatif lié à l'habitat intelligent. Leur approche s'appuie sur une architecture multi-agents dans laquelle les agents perçoivent l'état de l'environnement à partir des données senseurs et tentent, d'une manière rationnelle, de maximiser le confort et la productivité de l'occupant en se basant sur un système de reconnaissance du comportement. Ce modèle de reconnaissance d'activités exploite l'algorithme de reconnaissance et de prédiction *Active Lezi* [Gopalratnam, 2003], utilisé dans le domaine du forage de données ou *datamining* [Witten, 2005], permettant d'identifier des *patterns* séquentiels. De façon sommaire, il utilise les principes de la théorie de l'information afin d'apprendre, à partir de l'historique des séquences

enregistrées, les habitudes de vie de la personne pour ainsi pouvoir reconnaître les schémas de comportement qui sont habituels et inhabituels. Dans ces travaux, Jackula et Cook gèrent l'aspect temporel en créant une extension à l'algorithme *Active Lezi* qui intègre l'apprentissage des contraintes temporelles entre les actions de l'activité. La modélisation de ces contraintes se fonde, une fois encore, sur la logique temporelle d'Allen [Allen, 1983]. Par conséquent, ces travaux ne tiennent pas compte des durées de réalisation réelles des actions et des délais entre celles-ci. De plus, ce modèle nécessite une longue période d'apprentissage avant d'être vraiment performant et peut mener, dans le cas d'une personne souffrant de déficit cognitif, à l'apprentissage erroné de comportements qui sont répétés (donc habituel) mais incorrect.

Le troisième modèle recensé se base sur la Logique de Description (LD) [Baader et al., 2007] afin d'établir un modèle logique de reconnaissance d'activités [Weida, 1993]. Dans ces travaux, Weida utilise le formalisme de représentation terminologique de la LD pour encoder les actions de bas niveau qui sont représentées par des concept et définis par des rôles. Les activités sont par la suite modélisées à l'aide d'une structure de graphe temporel dans laquelle les actions (les étapes de l'activité) correspondent à des nœuds et les contraintes temporelles entre ces actions sont représentées par des arrêtes. Le système de reconnaissance est implémenté en fusionnant les systèmes de représentation de la connaissance K-Rep [Mays et al., 1991] et le système de raisonnement temporel MATS [Kautz et Ladkin, 1991]. Dès qu'un plan est défini, le système d'inférence vérifie sa cohérence syntaxique, normalise sa définition en dérivant l'information implicite et la classe dans la librairie de plans en vérifiant la relation de subsumption avec les plans définies antérieurement. Le système de raisonnement temporel adopté est un modèle proposé par Kautz et Ladkin qui se partage en graphe d'Allen [Allen, 1983], et qui admet des relations temporelle symboliques, et un graphe métrique. Cette partie du système d'inférence possède une limite caractérisée par la complexité des conversions des contraintes temporelles, à chaque instanciation, d'un fait dans les deux graphes cités précédemment, à savoir, les conversions métriques-symboliques ou symboliques-métriques.

La contribution de ce mémoire d'initiation à la recherche suit les traces des approches de reconnaissance temporelles citées, en essayant de faire un pas en avant en proposant des réponses à la problématique évoquée qui est liée au temps. Cette contribution revêt un volet théorique et un volet pratique / expérimental.

D'un point vue théorique, ce mémoire propose une extension du modèle de reconnaissance d'activités de Weida [Weida, 1993] dans laquelle nous avons incorporé, d'une part, la notion de concepts d'action en logique de description, qui permet de faciliter l'appariement des événements observés et des actions de librairie et, d'autre part, dans laquelle nous avons introduit une

nouvelle structuration temporelle augmentée permettant de tenir compte de l'aspect quantitatif lié aux durées et aux délais des actions, qui était absent du modèle original. La nouvelle structure de contraintes temporelles quantitatives proposée pour améliorer le modèle s'appuie sur les travaux de Dechter [Dechter, 1991], qui offrent un moyen efficace d'exploiter des intervalles de temps fermés en utilisant la notion d'instants. L'extension proposée dans le mémoire permet ainsi d'augmenter la performance et la précision du modèle original de Weida tout en permettant l'identification de nouvelles anomalies liées au temps.

D'un point de pratique, ce mémoire a conduit à l'implémentation du modèle proposé afin de démontrer sa faisabilité technique. Le modèle original de Weida a également fait l'objet d'une implémentation. Les deux modèles ont ainsi pu être comparés lors d'une expérimentation réalisée à l'aide d'un simulateur d'habitat intelligent [Bouchard et al., 2010] ainsi qu'à l'aide de données synthétiques tirées d'une étude clinique réalisée avec des sujets humains. Cette contribution pratique et expérimentale permet de bien montrer comment les améliorations théoriques proposées résultent en gains concrets au niveau de la performance, de la précision, et de la capacité de reconnaissance des erreurs liées au déficit cognitif.

1.5 Méthodologie de recherche

Le projet de recherche présenté dans ce mémoire s'est effectué en suivant une méthodologie de recherche se divisant en trois principales étapes.

La première étape du projet avait pour objectif d'acquérir une connaissance ciblée du domaine de recherche en effectuant une revue de la littérature disponible sur la problématique de la reconnaissance d'activités en général [Bouchard et al, 2007] [Pollack, 2005] [Patterson et al, 2005] [Carberry, 2001] et plus particulièrement les approches exploitant l'aspect temporel [Jakkula et Cook, 2006] [Weida, 1993] [Kautz, 1991]. Une étude des principaux outils formels dans le cadre du raisonnement temporel, notamment les théories d'Allen [Allen, 1983] et de Dechter [Dechter, 1991], a aussi été réalisée afin d'être en mesure de comprendre les travaux explorés. Le premier volet a permis d'avoir une vue d'ensemble du domaine de la reconnaissance d'activités, particulièrement dans un contexte applicatif lié à l'assistance des personnes en perte d'autonomie. Elle a permis de mieux cerner les enjeux et les besoins spécifiques d'un modèle de reconnaissance destiné à cet usage à l'intérieur d'un habitat intelligent. Le second volet de cette étape avait pour objectif de réaliser un état de l'art ciblé sur les modèles de reconnaissance existants qui exploitent les informations temporelles dans leur processus de reconnaissance. Ce

volet a permis d'aboutir à des pistes de solutions qui ont conduit à la contribution proposée dans le mémoire.

La deuxième étape avait comme objectif de définir les bases théoriques d'un nouveau modèle de reconnaissance d'activités capable de répondre à la problématique temporelle citée à la section 1.3, en se fondant sur les pistes dégagées lors de la première étape. Pour ce faire, nous avons adopté une approche qui consiste à proposer une extension d'un modèle existant que nous avons exploré, soit celui de Weida [Weida, 1993]. Ce modèle répondait bien à nos besoins applicatifs car il intègre à la base une gestion limitée des contraintes temporelles pouvant être étendue et il est compatible avec nos outils applicatifs actuels car il se fonde sur la logique de description [Baader et al., 2007] qui, comme il a été démontré dans [Bouchard, 2006] et [Roy et al., 2009], permet d'exploiter la relation de subsumption pour transformer le processus de reconnaissance en une opération de classification dans une ontologie (librairie de plans). Cette étape a permis la définition d'un nouveau modèle de reconnaissance étendu permettant d'identifier les anomalies liées aux délais et aux durées d'exécution d'une activité de la vie quotidienne d'un patient. Il répond ainsi en partie à la problématique soulevée.

L'étape finale du projet avait comme objectif la validation du modèle de reconnaissance temporel développé au cours de l'étape précédente, afin de vérifier l'aspect opérationnel de celui-ci. Pour cette étape, une implémentation en Java du modèle original de Weida a été produite puis, une version améliorée a été implémentée en intégrant les propositions théoriques formulées lors du volet précédent. En marge de cette étape, un simulateur d'habitats intelligents virtuels, nommé SIMACT, a été développé par notre équipe [Bouchard et al., 2010]. Il permet la simulation de scénarios de cas réels avec une visualisation en 3-dimensions du résultat. Ce simulateur a servi de plateforme expérimentale pour tester, avec des données synthétiques tirées d'une étude clinique, les performances du nouveau modèle proposé, notamment au niveau de la précision et de la capacité à reconnaître les erreurs liées à l'aspect temporel. Ces résultats ont ensuite pu être comparés à ceux obtenus avec la version originale du modèle de Weida et ont permis de démontrer l'apport des travaux du mémoire.

Il est à noter que cette dernière partie du travail a fait l'objet d'une publication scientifique de 10 pages dans la réputée conférence internationale Ubiquitous Computing and Multimedia Applications (UCMA '2010), dont l'éditeur est Springer, avec un taux d'acceptation de seulement 10% (30 articles sélectionnés sur 300 soumissions). Cette reconnaissance encourageante de la part des pairs vient appuyer les conclusions du mémoire ainsi que l'importance des travaux réalisés par notre équipe et des résultats obtenus.

1.6 Organisation du mémoire

Ce mémoire est organisé en quatre chapitres qui s'inscrivent en chronologie directe avec la méthodologie de recherche employée.

Le premier chapitre qui s'achève se voulait une introduction au mémoire, à notre contexte d'étude et à la problématique soulevée dans cette recherche. Il présente un résumé de la problématique, ainsi que quelques exemples aidant à sa compréhension. Il décrit également une synthèse des travaux antérieurs, de notre contribution originale, et présente les différentes étapes de la méthodologie de recherche employée dans le cadre de ce projet.

Le second chapitre propose une introduction aux outils formels du raisonnement temporel en intelligence artificielle. Il décrit en particulier la théorie proposée Allen [Allen, 1983], qui est utilisées abondamment dans un contexte de reconnaissance d'activités et qui permet de modéliser et de raisonner sur des intervalles de temps. L'exploration détaillée de cette théorie est importante car elle est à la base (précurseur) de la majorité des travaux effectués sur la modélisation du temps. Cette introduction aux principaux outils formels est nécessaire à la compréhension des modèles présentés dans le chapitre suivant.

Le troisième chapitre présente, quant à lui, un état de l'art ciblé concernant les principales approches de reconnaissance temporelle identifiées lors de la première phase de nos recherches. Il décrit les principaux travaux apparentés portant sur la reconnaissance d'activités qui exploitent l'aspect temporel [Kautz, 1991] [Weida, 1993] [Jakkula et Cook, 2006]. Enfin, la conclusion du chapitre dresse un bilan sur ces approches explorées qui permet de mieux situer notre contribution par rapport à ces travaux antérieurs.

Le quatrième chapitre porte sur la contribution théorique et pratique découlant de ce travail de recherche. La première section décrit les éléments formels permettant d'étendre le modèle Weida [Weida, 1993] pour y incorporer la notion d'intervalles fermés servant à représenter le temps, en s'appuyant sur la théorie des instants temporels de Dechter [Dechter, 1991]. La seconde partie du chapitre présente l'implémentation de ce nouveau modèle de reconnaissance ainsi que le développement d'un outil de simulation intitulé SIMACT, pouvant servir de plateforme expérimentale pour tester notre approche. Enfin, la dernière partie du chapitre porte sur la démarche expérimentale employée pour effectuer la validation du modèle proposé et présente une analyse comparative des résultats obtenus avec diverses versions du modèle (modèle sans prise en compte du temps, modèle original de Weida et nouvel extension proposée).

Enfin, le cinquième et dernier chapitre conclu le mémoire en présentant un bilan détaillé du projet de recherche qui met en relief l'apport de ce travail par rapport aux travaux antérieurs. Ce

chapitre traitera également des limitations de l'approche développée et des travaux futurs pouvant découlés de cette recherche. Le chapitre se termine par un bilan à caractère plus personnel de cette expérience d'initiation au monde de la recherche scientifique.

CHAPITRE 2

Raisonnement temporel en intelligence artificielle

2.1 Introduction

L'importance du temps et de sa représentation ont attiré l'attention des chercheurs en IA depuis plus de 30 ans déjà. Beaucoup de travaux ont été menés, dans le secteur de la modélisation de la connaissance, afin d'aboutir à une représentation générale du temps ainsi qu'à des outils formels qui permettent le raisonnement à partir de celle-ci [Russel et al., 2003]. Ainsi, plusieurs approches ont été proposées. Parmi elles on peut dégager deux grandes écoles de pensées distinctes concernant la représentation du temps [Tolba, 1991]. La première consiste à considérer le temps d'un point de vue purement numérique, ce qui signifie qu'elle vise principalement à répondre numériquement à des questions à variables réelle ou complexe. La seconde se fonde sur une représentation logique et conceptuelle basée sur un modèle axiomatique. D'une façon plus précise, le premier axe de recherche représente le temps sous forme d'une droite sur laquelle on cherche à situer des événements dans un référentiel temporel de façon numérique. Ces événements peuvent être des fenêtres temporelles ou des moments situés de début et de fin. Ces représentations utilisent des méthodes de recherche opérationnelle afin de gérer les relations temporelles (programmation dynamique, satisfaction de contraintes, etc.). Le deuxième axe de recherche se base sur une représentation symbolique du temps, fondée sur un formalisme logique. Cette branche se subdivise en trois types d'approches : celles basées sur une logique classique (prédicats), celle fondée sur une logique modale et celle exploitant une logique dite réifiée, c'est-à-dire qui appréhende les concepts en tant qu'objets concrets. Dans le cas des logiques temporelles réifiées, on peut aussi distinguer deux sous branches : le temps subjectif et le temps objectif. Le premier s'intéresse à l'aspect causal et irréversible des

événements, on parle alors d'analyseurs de systèmes temporels (*Temporal System Analysers*, TSA) [Hatou, 1991]. Le deuxième a une tendance plus descriptive, dans ce cadre on peut citer le cas des gestionnaires de graphes temporels (*Time Maps Managers*, TMM), dont le but est d'assurer la gestion et la mise à jour des données temporelles au sein de ce type de graphes, ils permettent aussi d'aborder les notions de succession et de simultanéité des événements [Hatou, 1991]. Il est à noter que la plupart des travaux liés au domaine de l'habitat intelligent et qui intègre un raisonnement temporel utilisent une représentation fondée sur une logique réifiée exploitant le temps subjectifs ou objectifs. La figure 2.1 résume en une vue synthétique les différentes visions du temps que nous venons de citées et qu'on retrouve dans la littérature de l'intelligence artificielle.

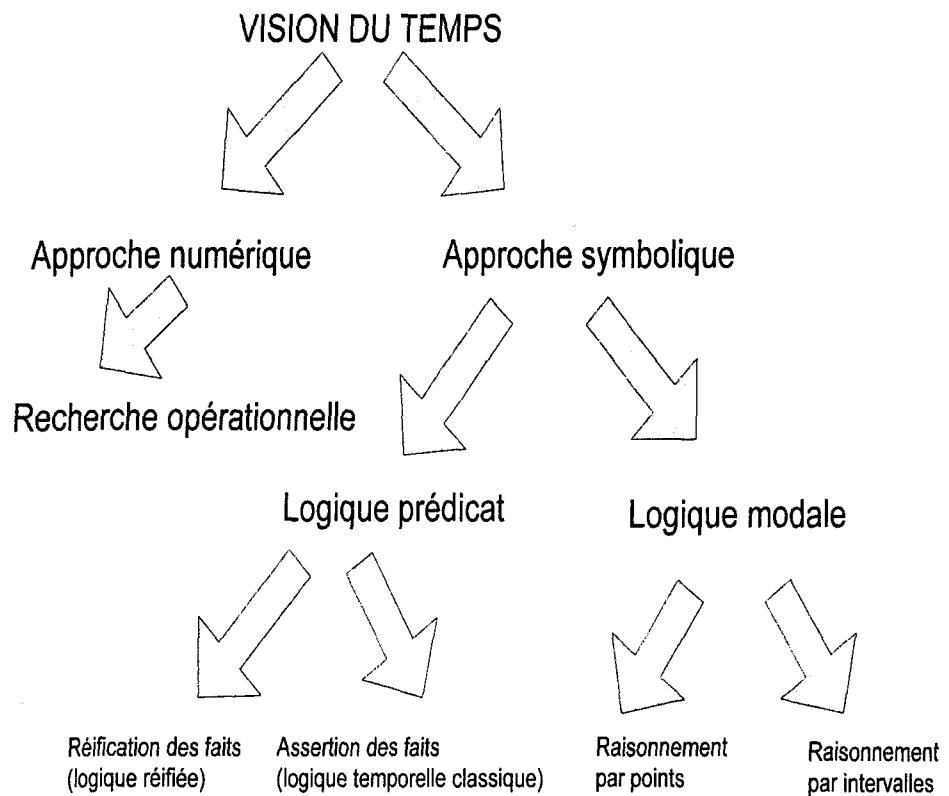


Figure 2.1. Les différentes approches de modélisation de temps [Bassiére, 1997]

Ce chapitre propose une étude des principaux outils formels employés en IA pour modéliser le temps issus des écoles de pensées symbolique et numérique. Il explore notamment la théorie d'Allen [Allen, 1983], qui se veut l'instigatrice d'une majorité des travaux sur cet axe, et celle de Dechter [Dechter, 1991], que nous exploitons dans le modèle que nous proposons au

chapitre 4. Cette mise en contexte permettra au lecteur de se familiariser avec les principales notions formelles nécessaires à la compréhension des modèles de reconnaissance d'activités qui feront l'objet de la suite du mémoire.

2.2 Approche symbolique

L'approche symbolique se divise principalement en deux grandes théories temporelles fondatrices : la célèbre théorie d'*Allen* [Allen, 1981], où l'élément de base est l'*intervalle* donnant naissance à une algèbre complète, et la théorie de *McDermott* [McDermott, 1982], basée sur la notion d'*instant*. Ces deux théories fournissent un moyen de représenter des relations temporelles entre des événements et sont parmi les plus utilisées en intelligence artificielle. À titre d'exemple, un grand nombre de travaux [Jakkula et Cook, 2007] [Weida, 1993] [Kautz et Ladkin, 1991] exploitent ces représentations dans un contexte de reconnaissance d'activités, afin de mieux situer les événements dans un cadre temporel expressif. Dans les deux prochaines sections, nous allons présenter ces modèles symboliques et nous ferons un bilan comparatif de celles-ci pour en bien saisir les forces et les faiblesses de chacune d'elles.

2.2.1 Logique temporelle d'Allen

La théorie d'Allen est connue sous le nom d'*algèbre des intervalles* [Allen, 1983]. Pour Allen, le temps est constitué d'une droite correspondant à une suite de point infinie et continue. Un intervalle représente quant à lui une fraction du temps (ou un point, dans le cas où la borne inférieure et supérieure de l'intervalle sont équivalents) situé sur cette droite, comme le montre la figure 2.2. Cet intervalle peut servir à représenter la durée d'un événement ou d'une action, comme « cuire des œufs ». Il peut aussi représenter la durée d'une activité ou un ensemble d'actions, comme par exemple « faire une salade grecque » qui va nécessiter respectivement les actions « couper piment », « couper tomate », « couper oignon », « couper cornichon » et « mélanger le tout ». Dans cette optique, on peut dire qu'un intervalle peut être subdivisé, si on le souhaite, en sous intervalles.

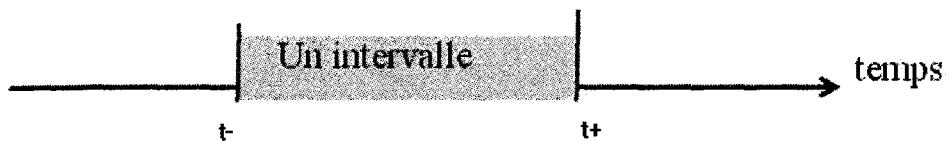


Figure 2.2. Les points limites d'un intervalle

D'une façon plus formelle, les systèmes traitant les intervalles sont des systèmes symboliques dans lesquels l'intervalle est un ensemble ordonné de points $T = \{t\}$, tel que :

$$(\exists t_-) (\forall t \in T) (t_- < t) \text{ et}$$

$$(\exists t_+) (\forall t \in T) (t_+ > t)$$

La paire (t_-, t_+) représente les points limites « end-points » qui signifient un temps de début et un temps de fin d'un événement, comme le montre la figure 2.2. Dans notre contexte applicatif, on peut considérer que cette notion d'événement réfère à une action détectée par un ou plusieurs capteurs distribués dans l'habitat.

2.2.1.1 Caractérisation des relations temporelles entre deux intervalles

En s'appuyant sur cette notion d'intervalle, Allen définit treize relations binaires permettant de caractériser toutes les relations temporelles pouvant exister entre deux événements (intervalles) distincts, telles qu'une action qui se produit juste après une autre action, pendant une autre, etc. La figure 2.3 qui suit décrit l'ensemble de ces relations.

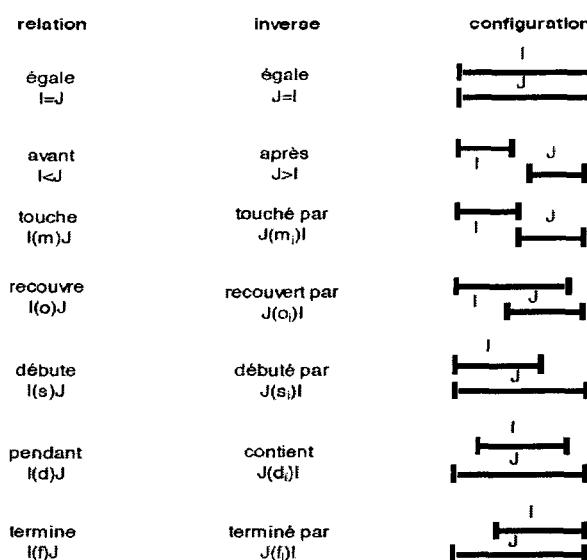


Figure 2.3. Les relations temporelles dans la théorie de Allen

Sur cette image, les symboles I et J représentent respectivement deux événements différents. Chacun de ces événements I et J correspond, quant à lui, à un intervalle de temps distinct, c'est-à-dire une portion finie de la ligne du temps balisée par une borne inférieure (t_-) et une borne supérieure (t_+), chacun. Sur la figure, la ligne située sous les symboles J dans la

colonne « configuration » permet de caractériser de façon relative la durée de cet événement ainsi que sa superposition temporelle par rapport à l'événement I. Ces relations sont complètes et mutuellement exclusives de telle sorte qu'une de ces relations est vraie entre deux intervalles datés.

2.2.1.2 Exemple applicatif utilisant les intervalles

Nous pouvons, à titre d'exemple, illustrer la grande capacité de ce modèle simple et flexible à représenter l'aspect temporelle d'une situation réelle à l'aide d'intervalles et de ces relations. Supposons, par exemple, qu'un patient soit en train de cuire des pâtes (événement représenté par l'intervalle I) et que celui-ci débute la préparation d'un café durant la réalisation du repas (événement représenté par l'intervalle J), préparation qui sera terminée seulement dans 20 minutes, soit après que le repas soit complété. La relation entre le premier événement (I) et le second (J) serait que *I recouvre J*, se qui s'exprime formellement par $I(o)J$, puisque l'événement « cuire des pâtes » à débuter avant « préparer un café », mais « préparer un café » c'est terminé après la fin de l'événement « cuire des pâtes ». En ce sens, on remarque toute la puissance de cette représentation qui peut être utilisée pour caractériser les relations temporelles entre les événements dans une multitude de situations.

2.2.1.3 Algèbre des intervalles

À partir des éléments que nous venons décrire, soit la notion d'intervalle et les treize relations binaires qui s'y rattachent, Allen propose une algèbre qui permet de travailler et de manipuler ces éléments. Formellement, on peut représenter celle-ci par une structure $\langle \text{Int}, B_{\text{int}} \rangle$ où $\text{Int} = \{I_1, I_2, I_3 \dots I_n\}$, est un ensemble finie d'intervalles et $B_{\text{int}} = \{<, m, s, f, d, o, =, d_i, f_i, s_i, o_i, m_i, >\} = \{B_1, B_2 \dots B_{13}\}$, est un ensemble de relations binaires sur Int appelé aussi relation élémentaire [Garlatti, 2002]. Ces relations sont appelées respectivement inférieur, *meet*, *start*, *finish*, *during*, *overlap*, *equal* et leurs relations inverses notée par un *i* en indice. B_{int} permet de décrire l'ensemble $2^{B_{\text{int}}}$ des relations composées par disjonction de ces relations élémentaires, qui sont aussi appelées relations disjonctives. Ces relations disjonctives servent à représenter les positions relatives entre deux intervalles lorsqu'elles ne sont pas totalement connues et ont les propriétés suivantes :

- Elles sont mutuellement exclusives :

$$\forall I_1, I_2 \in \text{Int} \text{ et } \forall R_1, R_2 \in 2^{B_{\text{int}}}, \neg (R_1(I_1, I_2) \wedge R_2(I_1, I_2)).$$

- B_{int} est une base complète :

$$\forall I_1, I_2 \in \text{Int} \text{ et } \forall R \in 2^{B_{int}}, R(I_1, I_2) \Leftrightarrow (B_{i1}(I_1, I_2) \vee B_{i2}(I_1, I_2) \vee \dots \vee B_{ik}(I_1, I_2)).$$

On note $R = \sum_i B_i$ avec $I = \{I_1, I_2, \dots, I_k\}$ et $\sum_i B_i$ une disjonction de relations entre deux intervalles de B_{int} . De manière générale, toute disjonction de relations de B_{int} est noté $Q = \sum_i B_i$ avec $I = \{I_1, I_2, \dots, I_k\}$. Les lois suivantes ont été définies sur l'ensemble B_{int} :

- La relation d'inclusion :

$$\text{Soit } I \text{ et } J \text{ deux intervalles, } R = \sum_i B_i, Q = \sum_j B_j, R \subseteq Q \text{ si et seulement si } I \subseteq J.$$

- L'union : $R \vee Q = \sum_{I \cup J} B_k$
- L'intersection : $R \wedge Q = \sum_{I \cap J} B_k$
- La composition : $R \circ Q = \sum_{I \times J} B_k$

Allen a construit une table qui définit la loi de composition qui est présentée à la table 2.1, pour tous les couples de la base. C'est une loi interne, associative, dont la relation d'égalité $=$ est l'élément neutre de \wedge ; ε est l'élément neutre de \vee . L'inverse de $B_i \circ B_j$ est $B'_j \circ B'_i$, avec B'_k la relation inverse de B_k . Les trois lois sont deux à deux distributives. On peut établir que $(2^{B_{int}}, \vee, \circ)$ a une structure d'algèbre d'intervalles définissant les propriétés suivantes :

- La règle d'inférence qui suit: Soit $I_1, I_2, I_3 \in \text{Int}$, si $R(I_1, I_2)$ et $Q(I_2, I_3)$ alors $[R \circ Q](I_1, I_3)$. Par exemple : Si I pendant J et J avant K alors I avant K .

- Deux relations de $2^{B_{int}}$ $R = \sum_i B_i, Q = \sum_j B_j$ sont en contradiction si et seulement si

$$R \wedge Q = \varepsilon \Leftrightarrow I \cap J = \emptyset.$$

- Si $P(I_1, I_3)$, $R(I_1, I_2)$ et $Q(I_2, I_3)$, il y a cohérence de la relation P par rapport à Q et R si et seulement si $[P \wedge (P \circ Q)] \neq \varepsilon$.

De façon générale, en fonction de la table de composition 2.1 présentée ci-dessous, qui est aussi appelée table de transitivité, Allen a défini une règle de transitivité qui permet de déduire la

position relative entre deux événements temporels, tout en connaissant la relation de chacun de ces événements avec un troisième [Tolba, 1991].

o	<	>	d	di
<	<	Les 13 primitives	< o d m s	<
>	Les 13 primitives	>	> oi d mi f	>
d	<	>	d	Les 13 primitives
di	< o m di fi	> oi di mi si	o oi	= s f si di fi

Table 2.1. Extrait de la table de transitivité [Allen, 1983]

2.2.1.4 Modélisation d'une situation avec les graphes temporels

À partir de l'ensemble d'outils formels que nous procure cette algèbre, on peut représenter la dynamique temporelle de plusieurs événements interagissant à l'intérieur d'une même situation à l'aide d'une structure de graphe, appelée graphe temporel ou réseau de contraintes temporelles. Dans une telle structure, un nœud du graphe représente un événement (un intervalle) et chaque arc reliant une paire d'événements, donc deux nœuds, représente un ensemble de relations temporelles possibles. En d'autres mots, ces arcs correspondent à une disjonction des relations élémentaires présentées à la figure 2.2. On peut effectuer des opérations sur le graphe afin d'inférer de nouvelles relations, en mettant à jour celui-ci, de façon à obtenir un graphe minimal en termes de relations temporelles à l'aide d'opérations de conjonctions, compositions, etc., telles que proposées par la théorie d'Allen. À titre d'exemple, prenons la situation suivante « *Fred lisait le journal pendant son petit déjeuner. Il pose le journal et termine son café. Après le petit déjeuner, il part se promener* » formulée en langage naturel. La représentation sous forme de graphe temporel de cette situation est illustré à la figure 2.4 [Van Beek, 1992].

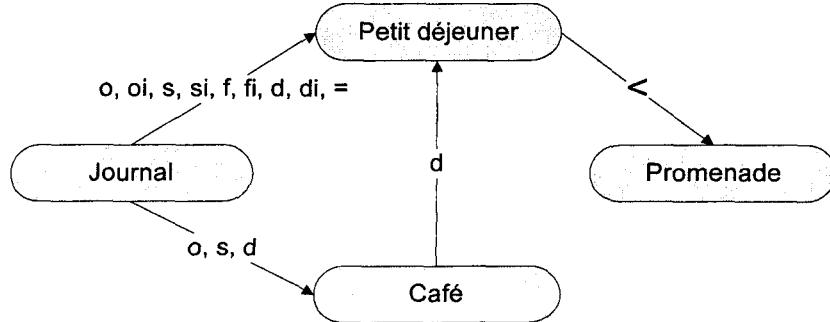


Figure 2.4. Exemple d'un graphe temporel selon la théorie d'Allen [Van Beek, 1992]

L'interprétation de cet exemple peut donner diverses variations, spécialement, celle du terme « pendant » qui peut signifier un chevauchement, commencement, fin ou inclusion entre les faits « *lire le journal* » et « *prendre le petit déjeuner* ». L'incertitude quant à la relation temporelle entre deux intervalles est envisageable quand les extrémités d'un intervalle ne sont pas connues, comme lors de la composition de plusieurs relations. Dans ce cas, les arcs ne portent plus une seule primitive, mais une disjonction de primitives qui va refléter cette incertitude. L'absence d'une information concernant les relations possibles entre les intervalles, engendre une attribution d'une disjonction de treize relations. L'étiquetage d'un arc peut être modifié que si un nouveau fait ou événement est introduit dans le graphe temporel en ajoutant une disjonction de primitives temporelles entre deux nœuds du graphe, comme le montre la figure 2.5. Cette nouvelle disjonction de primitives représente une contrainte temporelle qui doit alors être propagée dans le graphe de telle sorte à maintenir ou rétablir la cohérence. Cette propagation utilise le tableau de transitivité pour déduire toutes les relations qui rendraient le réseau inconsistante. Le calcul de la nouvelle relation est effectué selon l'algorithme suivant :

```

Contraintes (R1, R2)
C ← {} ;
Pour tout r1 dans R1
Pour tout r2 dans R2
    C ← C ∪ (r1 o r2) ;
Retourner C ;

```

Algorithme 2.1. Inférence d'une nouvelle relation temporelle dans un graphe

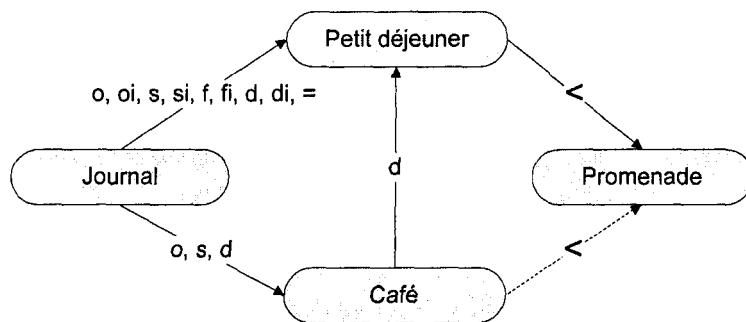


Figure 2.5. Graphe temporel après le calcul de la nouvelle relation

Cette nouvelle étiquette « C » est calculée par conjonction de l'ancienne étiquette avec le résultat de la composition de r₁ et r₂, notant que la conjonction de deux étiquettes est définie comme l'union des étiquettes.

2.2.1.5 Raisonnement sur un graphe temporel

Afin de mieux illustrer le déroulement de l'algorithme, nous allons détaillé pas à pas un exemple de construction d'un graphe temporel. Soit S, l'événement correspondant à appuyer sur l'interrupteur, L l'intervalle de temps durant laquelle la lampe est allumée, et R l'intervalle durant laquelle John est dans la pièce. On peut écrire les relations suivantes :

- La lampe s'est allumée pendant ou juste après qu'on ait appuyé sur l'interrupteur, d'où l'utilisation des primitives recouvre(o) et touche(m).
- John n'était pas dans la pièce lorsqu'on a appuyé sur l'interrupteur, on utilise : avant(<), touche(m), touché-par(mi) et après(>).

À partir de cette description, on peut construire le graphe temporel de la figure 2.6 :

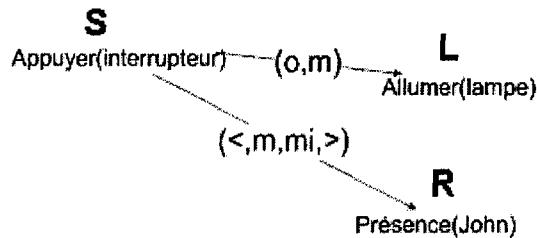


Figure 2.6. Graphe temporel construit au départ, avant l'application de l'algorithme

En utilisant maintenant l'algorithme 2.1, on peut effectuer l'inférence d'une nouvelle contrainte qui correspondra à un arc entre L et R. Le déroulement de l'algorithme s'effectue de la manière suivante :

John n'était pas là au moment où on a appuyé sur l'interrupteur, donc :

- Soit il est venu avant qu'on a appuie
 - Il est parti avant que la lampe s'allume : avant(<)
- Soit il est venu après qu'on ait appuyé sur l'interrupteur, à partir de là, plusieurs configurations sont possibles, soient :
 1. En même temps que la lampe allume : (débute s, termine-par e_i, débute-par s_i, égale =)
 2. Après que la lampe s'allume (recouvre o, contient d_i)
 3. Après que la lampe soit éteinte (après >, touche m)

Le graphe résultant correspond à la figure 2.7 suivante :

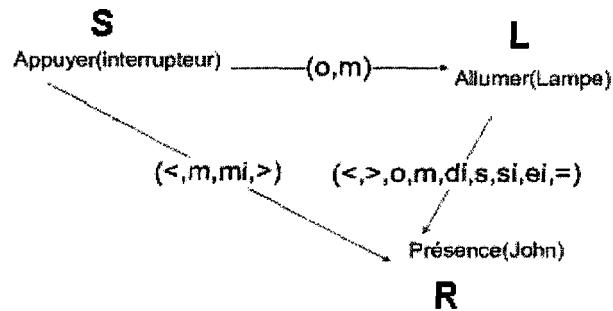
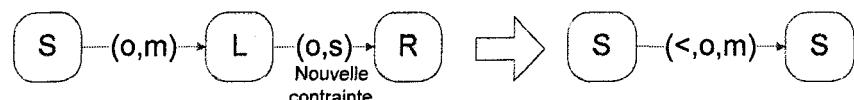


Figure 2.7. Graphe temporel après le calcul de la contrainte

Si on suppose que John est dans la pièce lorsque la lumière s'est éteinte, alors qu'il y était au moment qu'elle était allumée, alors le nouvel arc entre L et R se met à jour par la conjonction de l'ancienne valeur avec la nouvelle :

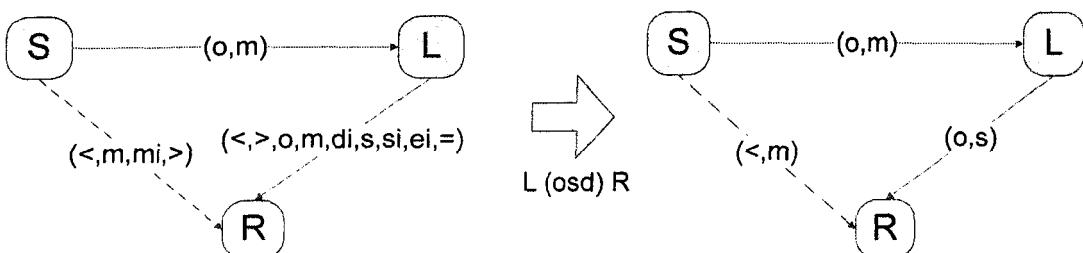
$$\text{Conjonction } ((<,>,o,o_i,m,d_i,s,s_i,e,e_i,=),(o,s,d)) = o,s$$

La propagation de cette nouvelle contrainte sur le graphe devient comme suit :



En appliquant la conjonction avec l'ancienne valeur, la relation entre S et R devient comme suit :

$$\text{Conjonction } ((<,o,m),(>,m_i,m,<)) = <,m$$



Enfin, l'inférence effectuée nous permet de déduire que John est arrivé dans la pièce après qu'on ait touché l'interrupteur. Dans ces travaux, Allen utilise un algorithme de propagation de contrainte, présenté ci-dessous (2.2), qui peut être vu comme un algorithme de consistance de chemins. Cet algorithme opère sur la totalité du graphe et il n'autorise pas les relations

impossibles entre deux intervalles [Tolba, 1991]. Pour des raisons d'efficacité, les techniques de propagation locale de contraintes sont en général les plus utilisées et permettent d'aboutir à une complexité polynomiale grâce à une approximation qui consiste à vérifier les contraintes de manière locale à deux, trois ou n nœuds du graphe. Les éléments de la disjonction de primitives qui seront supprimées sont ceux qui ne vérifient pas les contraintes localement. Allen utilise une telle technique en considérant une consistance seulement à trois nœuds. La propagation se fait par fermeture transitive sur le graphe en considérant, pour chaque couple d'intervalles (I, J), les modifications apportées sur les relations correspondantes de I et J avec leurs voisins.

```

Propager (Rab)
    Empiler(Rab) ;
    Tant que la pile n'est pas vide faire
        Dépiler(Rij) ;
        Pour tout k dans [1,n], k≠i et k≠j faire
            Nouv-Rik ← conjonction(Rik, contrainte(Rij, Rjk)) ;
            Nouv-Rkj ← conjonction(Rkj, contrainte(Rki, Rij)) ;
            Si Nouv-Rik = Ø ou Nouv-Rkj = Ø Alors :
                Contradiction temporelle (Arrêt) ;
            Si Nouv-Rik ≠ Rik Alors
                Rik ← Nouv-Rik ;
                Empiler Rik ;
            Si Nouv-Rkj ≠ Rkj Alors
                Rkj ← Nouv-Rkj ;
                Empiler (Rkj) ;

```

Algorithme 2.2. Propagation de contraintes [Allen, 1983]

La propagation de contraintes permet, lors de l'ajout d'un nouveau nœud (faits), de supprimer les primitives temporelles existantes entre deux intervalles qui ne peuvent qu'augmenter la précision de ces contraintes. Une étiquette de disjonction vide génère alors une incohérence entre les relations.

2.2.2 Représentation du temps basée sur la notion d'instant

L'algèbre des intervalles d'Allen, bien que largement utilisée, ne constituent pas la seule et unique façon symbolique d'appréhender le temps. Dans ces travaux, McDermott [McDermott, 1982] propose un modèle similaire à celui d'Allen, s'appuyant sur une logique temporelle réifiée

du premier ordre. La principale différence réside dans le fait que, contrairement à la présomption d'une ligne du temps définie et continue telle que vue par Allen, McDermott pose qu'une partie de cette ligne (le futur) est nécessairement indéterminée (donc plusieurs futurs possibles) et que le passé est unique, même s'il n'est pas nécessairement connu. Ce postulat de départ l'amène à introduire l'utilisation de primitives temporelles ponctuelles, c'est-à-dire un raisonnement sur des états, représentant ce que McDermott qualifie d'*instants*, au lieu de travailler avec des intervalles. Dans ce formalisme, le temps est donc linéaire dans le passé et branché dans le futur. Il est aussi qualifier de dense et infini [Eude, 1999], en rapport à la multitude de scénarios futurs possibles dans ce cas.

Dans le modèle de McDermott, un fait F est défini par l'ensemble des instants t pendant lesquels il est vrai, ce qui se note $vrai(F, t)$ [Tolba, 1991]. L'ensemble des états admettent une relation de précédence, avant ou même moment, noté \leq , qui est une relation d'ordre : réflexive, antisymétrique et transitive [Haton, 1991]. La figure 2.8 suivante présente un arbre d'états où l'ensemble est partiellement ordonné.

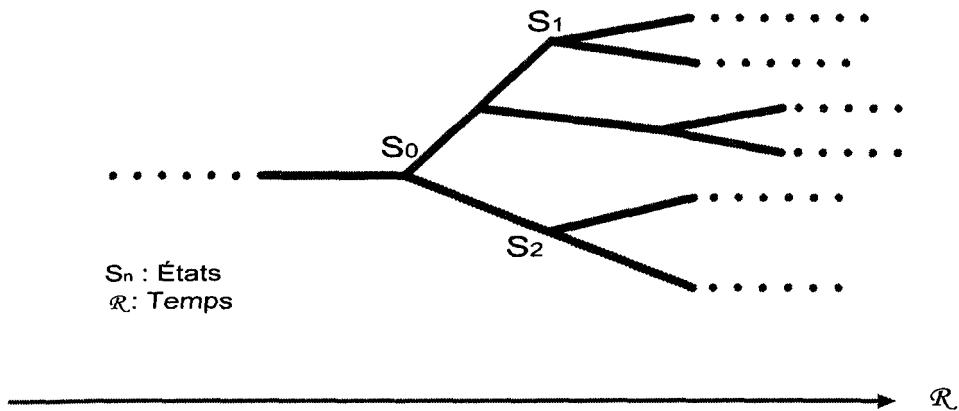


Figure 2.8. Arbre d'états selon le modèle de McDermott [Haton, 1991]

La relation d'ordre est linéaire à gauche (dans le passé) : pour tout état S_0, S_1, S_2 , si les relations $(\leq S_0 S_1)$ et $(\leq S_0 S_2)$ sont vérifiées alors la relation $(\leq S_1 S_2)$ ou $(\leq S_2 S_1)$ est vérifiée. Comme on peut le constater sur cette figure, le temps est arborescent et branché à droite (futur). Par conséquent, le graphe est un arbre induit de cette relation d'ordre. Pour tout état S_1, S_2 , si la relation $(\leq S_1 S_2)$ est vérifiée, alors il existe un état S_3 tel que les relations $(\leq S_1 S_3)$ et $(\leq S_3 S_2)$ sont vérifiées. Afin de retracer l'« histoire du monde », McDermott introduit la notion de *chronique*, qui peut être vue comme un ensemble d'états convexes, non bornés et totalement ordonnés. Elle est représentée par une branche dans l'arbre des états [McDermott, 1982].

Une fonction de datation a aussi été introduite afin de correspondre à chaque état, chaque date prenant sa valeur dans \mathcal{R} . Ces chroniques peuvent être considérées comme une copie de la ligne des temps \mathcal{R} qui se croisent en des points de ramifications de possibilités disjonctives à partir desquels leurs passées sont confondues. On représente le graphe temporel avec des nœuds correspondant aux états de début ou de fin d'un événement particulier, quand aux arcs ils sont orientés et représentent une contrainte de précédence.

La figure 2.9 illustre un exemple de graphe temporel suivant ce modèle. Chaque arc du graphe est étiqueté par une valeur métrique qui indique la durée entre deux états et qui représente un couple dénotant la valeur minimale et la valeur maximale pour tenir compte de l'incertitude qui peut avoir lieu entre les états. Le principe de représentation du graphe temporel peut considérer la présence de plusieurs contraintes entre deux nœuds, cela signifie qu'on peut avoir plusieurs chemins entre chaque paire de nœuds du graphe.

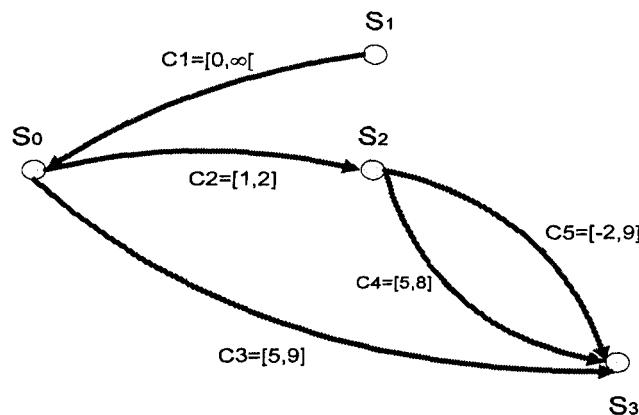


Figure 2.9. Graphe temporel suivant le modèle de McDermott [Hatou, 1991]

Dans l'exemple illustré ci-dessus, les chemins liant S_0 à S_3 sont $S_0 \xrightarrow{c_3} S_3$, $S_0 \xrightarrow{c_2} S_2 \xrightarrow{c_4} S_3$ et $S_0 \xrightarrow{c_5} S_2 \xrightarrow{c_5} S_3$. En faisant la somme des valeurs minimales et maximales sur chaque chemin, plusieurs contraintes sont obtenues entre les états S_i et S_j . Les valeurs pour cet exemple sont $[5,9]$, $[6,10]$ et $[-1,11]$. La durée minimale imposée entre i et j est représentée par le maximum des valeurs minimales $d_{\min_{ij}}$ entre ces deux nœuds. Le minimum des valeurs maximales $d_{\max_{ij}}$ représente la durée maximale. Au cas où le maximum des valeurs minimales est supérieur au minimum des valeurs maximales, on peut dire alors qu'il y a une incohérence du graphe. Pour l'exemple du graphe temporelle de la figure 2.9, l'ensemble de contraintes est bien cohérent puisque $d_{\min_{13}} = 6$ et $d_{\max_{13}} = \infty$. L'introduction d'un nouveau nœud dans le graphe est le fait d'introduire un moment où $d_{\min_{nouv}}$ représente l'instant au plus tôt de S_i et $d_{\max_{nouv}}$ l'instant au

plus tard de S_i . L'ajout d'un nouvel arc c_{ij} entraînera la modification des $d_{\min_{\text{nouv}}}$ et $d_{\max_{\text{nouv}}}$ des seuls états successeurs de l'état S_j . La figure 2.10 illustre l'exemple déjà traité à la figure 2.4 en se servant cette fois-ci du formalisme de McDermott. Les flèches simples représentent la relation avant et les flèches doubles la relation même-moment.

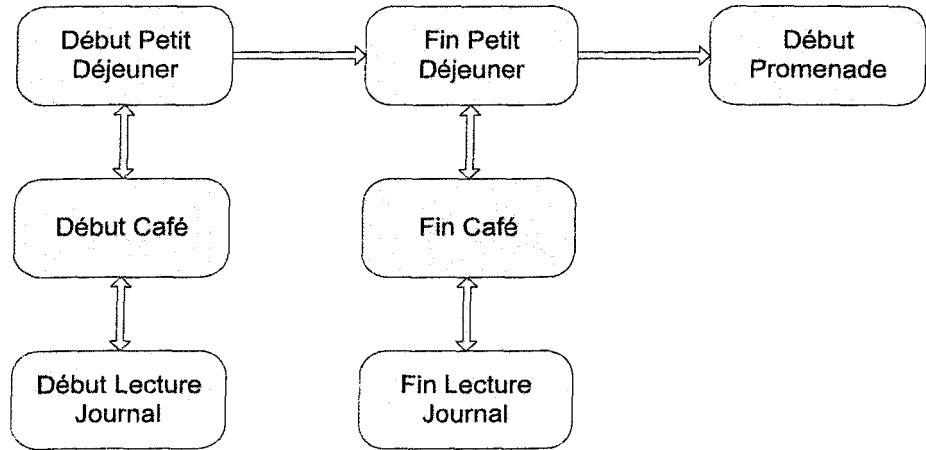


Figure 2.10. Exemple de la figure 2.4 modélisé avec l'approche de McDermott

2.2.3 Comparaison des approches d'Allen et de McDermott

En raison de leur proximité algébriques, l'algèbre des intervalles d'Allen et celle des instants de McDermott peuvent se rejoindre et être ramener en une algèbre de points en partant du principe des temps de début et de fin de chaque intervalle [Ghallab, 1989], comme le montre la table 2.2. Dans l'algèbre des instants, chaque intervalle doit être caractérisé par deux instants ordonnés associés à son début et à sa fin. Soient I et J , deux intervalles d'Allen, I_- et J_- sont respectivement le début de I et J , I_+ et J_+ sont respectivement la fin de I et J , avec $I_- < I_+$, $J_- < J_+$. À l'aide du formalisme précédent, on peut donc représenter les relations entre instants en considérant l'ensemble fini d'instants $\text{Inst} = \{t_1, t_2, \dots, t_n\}$ et $B_{\text{inst}} = \{<, =, >\} = \{B_1, B_2, B_3\}$, qui correspond à un ensemble de relation mutuellement exclusives sur Inst , appelé relations élémentaires. Ainsi, l'ensemble 2^{Binst} correspond aux relations composées par disjonction des relations élémentaires, ce qui se traduit de la façon suivante:

$$\forall t_1 \text{ et } t_2 \in \text{Inst} \text{ et } \forall R(t_1, t_2) \in 2^{\text{Binst}} \text{ on a } R(t_1, t_2) = [B_{i_1}(t_1, t_2) \vee \dots \vee B_{i_k}(t_1, t_2)], \text{ on note } R = \sum_i B_i \text{ avec } I = \{i_1, \dots, i_k\}$$

Pour deux intervalles I et J liés temporellement par une relation $B_i(I, J)$, on peut calculer les relations équivalentes entre $I-$, $I+$, $J-$ et $J+$. Par exemple : $<(I, J) \leftrightarrow <(I-, I+) \wedge <(I+, J-)$

Relation entre intervalles	Relation entre les instants
$I < J$	$I+ < J-$
$I = J$	$(I- = J-) \wedge (I+ = J+)$
I overlaps J	$(I- < J-) \wedge (I+ > J-) \wedge (I+ < J+)$
I meets J	$I+ = J-$
I during J	$(I- > J-) \wedge (I+ < J+)$
I starts J	$(I- = J-) \wedge (I+ < J+)$
I finishes J	$(I- > J-) \wedge (I+ = J+)$

Table 2.2. Correspondance entre les relations d'intervalles d'Allen et les contraintes d'instants de McDermott [Garlatti, 2002]

La représentation de McDermott permet de manipuler l'aspect numérique du temps, d'où une possibilité de gérer des contraintes métriques sur les durées. Pour des raisons de commodité de manipulation, il est souvent préférable de choisir l'algèbre d'instants pour une complexité $O(n^3)$, le graphe induit par la relation de précédence est globalement plus cohérent que celui d'Allen [Haton, 1991]. Parmi les principales divergences entre les deux théories, on trouve que celle de McDermott considère plusieurs futurs possibles. Par contre, il n'a pas introduit la notion de plusieurs passés hypothétiques [Haton, 1991]. La représentation du temps selon le modèle d'Allen reste plus riche que celle de McDermott du point de vue l'expressivité, mais les algorithmes qui en découlent sont plus complexes et donc moins efficaces.

2.3 Représentation numérique du temps

Les modèles que nous venons de parcourir proposaient une représentation symbolique et relative du temps. Ils s'intéressaient, d'une manière qualitative, aux liens temporels relatifs (ex. ordonnancement temporelle) existant entre les différentes actions composant une activité. La connaissance reliant l'aspect temporel peut cependant être représentée aussi sous une forme numérique. Ce type de représentation s'avère très utile pour ramener à une quantification du temps des événements. Plusieurs travaux ont été élaborés dans cette optique, parmi eux, on retrouve ceux de Fontaine [Fontaine, 1993], Dousson [Dousson, 1994] et de Dechter [Dechter et al., 1991], qui considèrent majoritairement que l'élément de base dans les représentations

numériques est le point, c'est-à-dire la plus petite unité de temps observable. Dans un tel contexte numérique, on fait aussi appel à un graphe temporel pour représenter une situation donnée. Les nœuds de ce graphe représentent des événements ponctuels pour lesquels le début ou la fin de l'événement admet une durée. Dans un cadre plus général, les relations temporelles sont des disjonctions d'intervalles numériques quelconques, appelés aussi fenêtres temporelles, ayant la forme $[a, b]$, tel que a et b sont des points fixés dans le temps, délimitant de façon concrète (et non relative) le moment de début le plus tôt et de fin le plus tard, d'une action donnée.

2.3.1 L'approche temporelle de Dechter

Dechter [Dechter, 1991] présente une approche de raisonnement temporel basée sur un formalisme de réseau de contraintes, afin d'avoir une représentation optimisée appelé réseau minimal, qui encodera toutes les relations temporelles entre une paire de points liés à un événement, en incluant les limites absolues dans leur différence de temps. Dechter considère aussi les événements comme étant des propositions auxquelles on associe des intervalles temporels. Par exemple, les propositions « j'ai conduit une voiture » ou « le livre est posé sur la table ». Chaque intervalle représente une période de temps durant laquelle la proposition s'accorde. Dechter distingue deux types d'informations temporelles, une information relative, comme par exemple « P_1 se produit avant P_2 », ou bien une information métrique, comme par exemple « P_1 avait commencé au moins 3 heures avant que P_2 termine ». Les expressions disjonctives peuvent être nécessaires afin d'exprimer moins d'information spécifiques, par exemple « tu peux venir avant ou après l'heure de dîner ». Dans sa représentation temporelle, Dechter se sert aussi des références pour les temps absolus comme « 4:00 PM », et pour les durées des propositions, comme par exemple « P a une durée d'au moins 2 heures et d'au plus 4 heures ». Des informations temporelles de ce type font appel à des questions comme : est-il possible que la proposition P se déroule au temps t ? Quelles sont les relations temporelles possibles entre deux propositions P_1 et P_2 ? L'approche proposée par Dechter est relativement simple et permet de répondre à ce type de question.

De façon synthèse, Dechter affirme que si un ensemble de propositions représentent des événements, et que chaque proposition P_i est associée à un intervalle $I_i = [a_i, b_i]$, alors l'information qui concerne le choix du temps des événements peut être exprimé par des contraintes sur les intervalles. Comme on a vu dans précédemment (section 2.1), Allen a défini treize relations qui peuvent exprimer une position d'un intervalle par rapport à un autre dans le temps. Vilain et Kautz [Vilain et al., 1986] ont suggérés que l'information puissent être exprimée

par des contraintes sur le point de début et de fin de chaque intervalle. De cette approche découle un algorithme s'exécutant en temps polynomial, mais qui ne peut manipuler que certaines classes limitées de problèmes. Ladkin et Maddux [Ladkin et Maddux, 1989] ont proposés une approche algébrique pour des problèmes similaires à ceux qui sont posés par Allen, Vilain et Kautz. À la différence de ces approches, le système de Dechter traite les connaissances métriques, il considère les points du temps comme étant des variables sur lesquelles on peut établir des contraintes, un point dans le temps peut être un point de début ou un point de fin d'un événement quelconque, comme il peut être un point neutre du temps comme 4 :00 pm. Malik et Binford [Malik et Binford, 1983] et Valdés [Valdés, 1986] ont proposés d'établir des contraintes sur les distances temporelles entre des points du temps, à savoir, si x_i et x_j sont deux points du temps, il existerait une contrainte sur leur distance temporelle de la forme $x_j - x_i \leq c$ qui donne un ensemble d'inégalités linéaires sur les x_i . Cependant, de telles contraintes sont insuffisantes et il est nécessaire de prendre en considération les expressions disjonctives.

Considérons l'exemple suivant : « John va à son travail soit en voiture, auquel cas cela lui prendra de 30 à 40 minutes, soit par autobus, auquel cas cela lui prendra 60 minutes ou plus. Fred va à son tour au travail, soit avec sa voiture et cela lui prendra de 20 à 30 minutes, soit en covoiturant, ce qui lui prendra de 40 à 50 minutes ». Soit P_1 , la proposition « John allait au travail » et P_2 la proposition « Fred allait au travail ». On associe respectivement à P_1 et P_2 les intervalles $[x_1, x_2]$ et $[x_3, x_4]$, pour lesquels x_1 représente le temps où John a quitté la maison, alors que x_4 représente le temps à lequel Fred est arrivé au travail. En tenant compte des contraintes citées précédemment, on peut décrire les inégalités s'appliquant pour John: $30 \leq x_2 - x_1 \leq 40$ ou $x_2 - x_1 \geq 60$, ce qui correspond à dire que la durée du trajet de John sera de 30 à 40 minutes inclusivement dans le cas où il prenne le bus et de 60 minutes dans le cas où il prenne sa voiture. Dans le cas de Fred, ces contraintes correspondent à $20 \leq x_4 - x_3 \leq 30$ ou $40 \leq x_4 - x_3 \leq 50$. Supposons maintenant qu'aujourd'hui, John ait quitté la maison entre 7:10 et 7:20 et que Fred soit arrivé au travail entre 8:00 et 8:10. ». Si nous savons que John est arrivé au travail entre 10 à 20 minutes après que Fred ait quitté sa maison, pouvons-nous répondre aux questions suivantes : « Est-ce que ce scénario est cohérent ? », « Est-ce qu'il est possible que John prenne sa voiture et que Fred covoiture ? ». La réponse aux deux premières questions est affirmative, ce scénario est bel et bien cohérent et John a pu prendre le bus pendant que Fred covoiturait. On peut prouver ces faits en posant le scénario suivant : John part à $x_1=7h10$, il arrive au travail à $x_2=7h40$, il a donc pris le bus et répond à la contrainte imposée car $20 \leq 7h40 - 7h10 \leq 30$. Fred part de la maison à $x_3=7h30$, arrive à $x_4=8h10$, et il a fait du covoiturage en répondant à la

contrainte $40 \leq 8h10 - 7h30 \leq 50$. De plus, on note que $10 \leq x_2 - x_3 \leq 20$, car $7h40 - 7h30 = 10$ minutes, ce qui signifie que, dans ce cas, il est donc tout à fait possible que John soit arrivée à destination entre 10 à 20 minutes après le départ de Fred de la maison. Le scénario est par conséquent cohérent.

Afin de représenter et traiter de tels problèmes, Dechter [Dechter, 1991] introduit une structure de graphe basée sur un formalisme de réseau de contraintes, à savoir, les problèmes de satisfaction de contraintes temporelles (TCSP) et les problèmes temporels simples (STP). L'approche de résolution de problème par satisfaction de contraintes temporelles peut être définie comme un réseau de problèmes temporels simples (STPs) qui, à son tour, est considéré comme un TCSP limité et simplifié. Les contraintes temporelles d'un TCSP sont des intervalles fermés respectivement disjonctifs et non disjonctifs (ou simples) alors que dans un STP les contraintes sont des intervalles simples et non disjonctifs et sa résolution peut être effectuée dans un temps polynomiale. Les TCSP ont un pouvoir d'expressivité très riche, comme le montre la figure 2.10 de l'exemple du scénario précédent, mais sa résolution est très complexe. Pour cela, Dechter propose un ensemble de techniques afin de pallier à ce problème à savoir, la décomposition du TCSP en un nombre exponentiel de STP, un schéma d'approximation et des approches basés sur les réseaux. Dans l'approche proposée dans ce mémoire, on utilisera le modèle de Dechter sans tenir compte des contraintes disjonctives entre les événements et par conséquent, on se limitera à présenter les STP qui nous seront utiles dans notre contribution.

2.3.1.1 Représentation temporelle avec les STP

La définition d'un problème temporel simple (STP) est très proche de celle d'un CSP ou plus exactement d'un TCSP. Un STP implique un ensemble de variables de points du temps $X=\{x_1, \dots, x_n\}$ représentant des événements, et un ensemble de contraintes binaires sur les points du temps, $C=\{c_1, \dots, c_m\}$, limitant la différence du temps entre deux événements. Chaque arc $i \rightarrow j$ est étiqueté par un intervalle $[a_{ij}, b_{ij}]$ représentant une contrainte de la forme $a_{ij} \leq x_j - x_i \leq b_{ij}$. Alternativement, la contrainte pourrait être exprimée comme une paire d'inégalités : $x_j - x_i \leq b_{ij}$ et $x_i - x_j \leq a_{ij}$. Ainsi, pour résoudre un STP, cela revient à résoudre un ensemble d'inégalités linéaires sur l'ensemble des variables x_i inclus dans X .

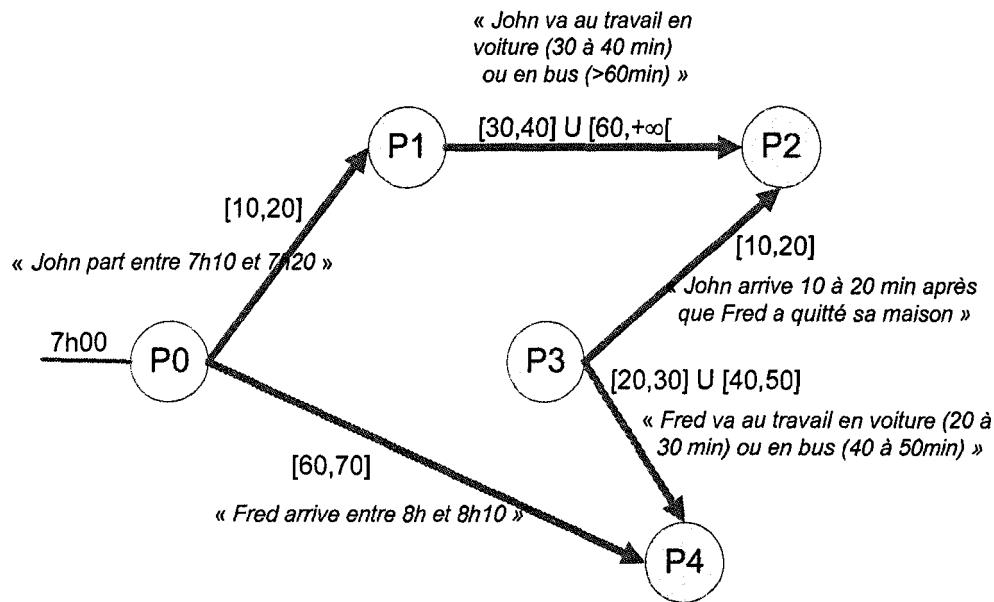


Figure 2.10. Représentation temporelle à l'aide des TCSP [Eude, 1999]

Le problème de résolution d'un système d'inégalités linéaires est bien connu dans la littérature, il peut se résoudre avec la méthode du simplexe, qui est exponentielle, ou bien avec l'algorithme de Khachian [Khachian, 1979], qui est cependant complexe à mettre en oeuvre en pratique. La classe des inégalités linéaires caractérisant un STP admet une solution plus simple, par le fait de donner une représentation de graphe convenable aux inégalités en y appliquant l'algorithme du plus court chemin [Aspvall et Shiloach, 1980] [Leiserson et saxe, 1983]. Une structure de données appelé graphe de temps (*time maps*), introduite par Dean et McDermott [Dean et McDermott, 1987], peut faciliter la planification, mais son principal inconvénient est qu'elle n'admet pas une formalisation mathématique.

Dechter associe à un STP un graphe avec des arcs pondérés et orienté de la forme $G_d = [V, E_d]$, appelé graphe de distance qui admet le même ensemble de nœuds tel que G , et chaque arc $i \rightarrow j$ est étiqueté par un poids a_{ij} représentant l'inégalité linéaires $X_j - X_i \leq a_{ij}$. Reprenant le scénario cité concernant John et Fred, si on considère que John à pris la voiture et Fred le co-voiturage, on a : $C_{12} = \{[30,40]\}$ et $C_{34} = \{[40,50]\}$ et un graphe de distance comme le montre la figure 2.11. Chaque chemin de i à j dans G_d , $i_0 = i$, $i_1, \dots, i_k = j$, répond à la contrainte suivante : $X_j - X_i \leq \sum_{j=1}^k a_{i_{j-1}, i_j}$.

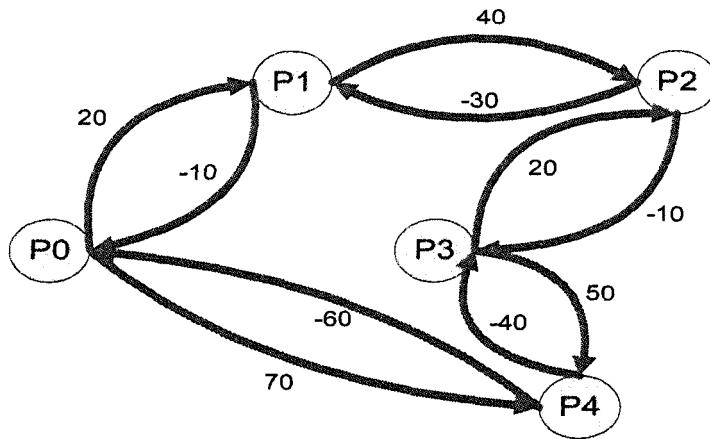


Figure 2.11. Un graphe de distance représentant une partie d'un TCSP [Dechter et al, 1991]

S'il existe plus qu'un chemin de i à j , alors il est facile de vérifier l'intersection de toutes les contraintes de chemin, $X_j - X_i \leq d_{ij}$ où d_{ij} est la longueur du plus court chemin de i à j . Dans ce qui suit, on trouve les propriétés et les conditions établies par Dechter, pour lesquels un STP peut être considéré comme un problème résolu.

- **Consistance**

Un STP est dit consistant si et seulement si son graphe de distance G_d n'admet pas de cycle négatif. S'il existe un cycle négatif C sur les nœuds $i_1, \dots, i_k = i_1$, alors la somme des inégalités le long de C donne $X_{i_1} - X_{i_1} \leq 0$, qui ne peut pas être satisfaite. Inversement, s'il n'existe pas un cycle négatif dans G_d , alors le chemin le plus court entre chaque paire de nœuds est bien défini. Pour toute paire de nœuds i et j , le chemin le plus court satisfait $d_{0j} \leq d_{0i} + a_{ij}$, ainsi $d_{0j} - d_{0i} \leq a_{ij}$. De là, le tuple (d_{01}, \dots, d_{0n}) est une solution de cet STP. Plus généralement, un STP peut être spécifié par un graphe dirigé, dit *d-graph*, où chaque arc $i \rightarrow j$ est étiqueté par le chemin le plus court d_{ij} dans G_d qui correspond à une représentation plus explicite du STP.

- **Décomposition**

Tout STP consistant est décomposable par rapport aux contraintes de son *d-graph*. L'importance de ce théorème est qu'il permet d'établir un algorithme efficace pour avoir une solution pour un STP, en attribuant à chaque variable, toute valeur qui satisfait les contraintes de *d-graph* par rapport aux attributions précédentes. La Décomposition assure qu'une telle valeur peut toujours être trouvée, sans tenir compte de l'ordre de l'attribution.

2.3.1.2 Réseau minimal

On peut remarquer que le d-graph peut représenter un réseau plus restreint que le STP original. À partir du théorème de décomposition, on peut conclure que ce nouveau réseau est le réseau minimal. Soit T un STP consistant, et M défini par : pour tout i, j , $M_{ij} = \{[-d_{ij}, d_{ij}]\}$ est la représentation du réseau minimal de T . Soit le graphe de distance de la figure 2.11. Tant qu'il n'existe pas de cycle négatif, le STP correspondant est consistant. Les distances des plus courts chemins d_{ij} sont montrées dans la table 2.3. Les domaines minimaux sont $10 \leq x_1 \leq 20$, $40 \leq x_2 \leq 50$, $20 \leq x_3 \leq 30$ et $60 \leq x_4 \leq 70$. En particulier, une solution spéciale est le tuple (d_{01}, \dots, d_{04}) , à savoir, l'attribution $\{x_1 = 20, x_2 = 50, x_3 = 30, x_4 = 70\}$, qui choisit pour chaque variable son dernier temps possibles. Selon cette solution et en reprenant l'exemple de John et Fred, on peut dire que John a quitté la maison vers 7h10 et il est arrivé au travail à 7h50, alors que Fred a quitté la maison vers 7h30 et il est arrivé au travail à 8h10. D'après le tableau 2.4, on s'aperçoit que le réseau minimal est symétrique dans le sens que $T_{ij} = \{[a, b]\}$ donc $T_{ji} = \{[-b, -a]\}$. Un scénario alternatif, dans lequel John utilise le bus et Fred utilise le covoiturage, où $T_{12} = \{[60, \infty]\}$ et $T_{34} = \{[40, 50]\}$, aura un cycle négatif et une inconsistance, démontrant l'impossibilité de celui-ci.

	0	1	2	3	4
0	0	20	50	30	70
1	-10	0	40	20	60
2	-40	-30	0	-10	30
3	-20	-10	20	0	50
4	-60	-50	-20	-40	0

Table 2.3. Longueur des chemins les plus courts de la figure 2.11 [Dechter et al, 1991]

	0	1	2	3	4
0	[0]	[10,20]	[40,50]	[20,30]	[60,70]
1	[-20,-10]	[0]	[30,40]	[10,20]	[50,60]
2	[-50,-40]	[-40,-30]	[0]	[-20,-10]	[20,30]
3	[-30,-20]	[-20,-10]	[10,20]	[0]	[40,50]
4	[-70,-60]	[-60,-50]	[-30,-20]	[-50,-40]	[0]

Table 2.4. Le réseau minimal correspondant à la figure 2.11 [Dechter et al, 1991]

2.3.1.3 Algorithme Floyd-Warshall

Le d-graph d'un STP peut être construit en appliquant l'algorithme *Floyd-Warshall* [Papadimitriou et Steiglitz, 1982], représenté dans l'algorithme 2.3, sur un graphe de distance.

```

FloydWarshell (G graphe de distance)
  D = G
  Pour k = 1 à n
    Pour i = 1 à n
      Pour j = 1 à n
        Dij = min (Dij, Dik + Dkj)

```

Algorithme 2.3. Algorithme Floyd-Warshall [Dechter et al, 1991]

Le temps de calcul de cet algorithme est de $O(n^3)$, il détecte les cycles négatifs en examinant le signe des éléments de la diagonale (D_{ii}). Il constitue donc un algorithme à temps polynomial pour déterminer la consistance d'un STP et pour calculer le domaine minimale et le réseau minimal. Une fois que le d-graph est disponible, la génération d'une solution nécessite seulement un temps de $O(n^2)$ vu que chaque attribution successive a besoin d'être vérifié avec les attributions précédentes.

2.3.2 Conclusion sur les approches numériques

Comme nous avons pu le constater, les approches numériques, telle que celle proposée par Dechter, offrent un moyen simple et efficace de représenter et de manipuler le temps d'une façon quantitative. On peut utiliser ce type d'approches avec diverse structure, dont les STP et les TCSP. Bien que les TCSP aient un pouvoir représentatif plus riche, ils sont aussi beaucoup plus complexes à manipuler, contrairement aux STP où les contraintes temporelles sont des intervalles fermés respectivement non disjonctives (simple). Dans le cadre applicatif qui nous intéresse dans ce mémoire, la représentation par STP est suffisante pour le modèle temporel que nous souhaitons développer. Pour ce faire, nous exploiterons une représentation numérique des événements afin de tenir compte les indications quantitatives sur les durées et les dates, ce qui n'est pas le cas avec l'aspect symbolique. Nous reviendrons sur cette modélisation au chapitre 4 du mémoire.

2.4 Approche mixte

Bien que la très grande majorité des travaux sur la représentation temporelle adopte une approche symbolique ou numérique, certains travaux ont tenté de combiner les deux types d'approches afin de marier l'expressivité de l'approche symbolique à l'efficacité calculatoire de l'approche numérique. Cette perspective a été envisagée sous deux formes différentes par Kautz et Ladkin [Kautz et Ladkin, 1991] et Meiri [Meiri, 1991]. Dans cette partie du chapitre, nous allons faire un bref survol de ces approches.

2.4.1 Approche de Kautz et Ladkin

Kautz et Ladkin [Kautz et Ladkin, 1991] ont proposé d'utiliser deux graphes distincts dans leur système nommé MATS. Le premier est similaire à celui d'Allen où l'on peut exprimer toutes les relations symboliques entre les intervalles. Le deuxième est un graphe de type STP qui tient compte des intervalles fermés, ouverts, semi-ouverts ou semi fermés. Il permet de compléter les relations exprimées par le réseau d'Allen avec des précisions sur les dates absolues de début ou de fin d'un intervalle, sur les durées des intervalles ou sur les délais entre eux. Pour tout intervalle I du graphe d'Allen, on introduit dans le STP les deux points du temps I_L et I_R de début et de fin de l'intervalle avec la relation obligatoire $I_L < I_R$. Les deux graphes s'impliquent mutuellement : si I avant J alors on peut déduire que $I_R - J_L < 0$ et réciproquement. Il existe d'autres relations d'Allen faisant intervenir plusieurs contraintes numériques plus faciles à traduire, par exemple, si $3 < (I_R - I_L) < \infty$ et $-\infty < (J_R - J_L) < 2$ alors on peut inférer que I ne peut pas être pendant J puisqu'il dure plus longtemps. Kautz et Ladkin proposent deux algorithmes permettant de passer d'un graphe à un autre, ce passage coïncide avec une perte minimale d'information. Au cas où on aura à modifier le graphe, on effectue sa traduction dans son graphe complémentaire. On reviendra à l'utilité du formalisme de Kautz et Ladkin dans ce mémoire plus tard, lors de la description du travail de Weida [Weida, 1993] sur la reconnaissance d'activités basée sur la logique de description et sur le modèle temporelle d'Allen.

2.4.2 Approche de Meiri

À la différence de l'approche que présentent Kautz et Ladkin [Kautz et Ladkin, 1991], Meiri [Meiri, 1991] regroupe la totalité de l'information dans un seul graphe dont les nœuds peuvent être des instants ou des intervalles. Dans ce type de graphe, on retrouve les relations symboliques de l'algèbre d'intervalles ou de l'algèbre des instants. De plus, on trouve les relations numériques entre points du type TCSP avec des intervalles fermés, ouverts, semi

ouverts ou semi fermés. On peut trouver aussi des relations qualitatives entre un point et un intervalle ou l'inverse comme le montre la figure 2.12.

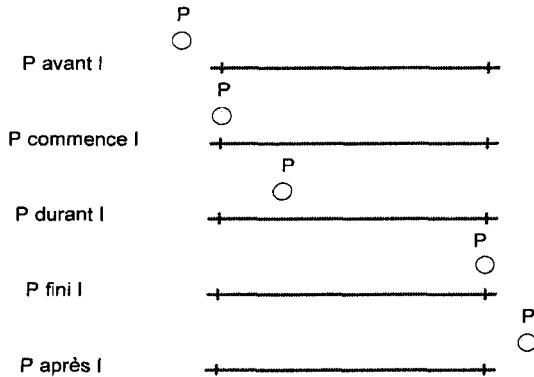


Figure 2.12. Relation de base entre un point P et un intervalle I [Meiri, 1991]

On forme avec toutes les relations symboliques une algèbre qualitative comportant les 2^{13} relations de l'algèbre d'intervalles, les 2^3 relations de l'algèbre de points, les 2^5 relations intervalles points et les 2^5 relations points intervalles. Pour tout intervalle I, on introduit les contraintes (I_L débute I) et (I_R termine I). On peut avoir, soit une relation symbolique, soit une relation numérique entre deux points. Pour maintenir ces deux types de relations temporelles, on établit une équivalence entre l'algèbre d'instants et les relations numérique entre les instants.

2.4.3 Comparaison entre les approches de Kautz et Meiri

Kautz et Ladkin ont introduit un modèle alternatif pour le raisonnement temporel mixte (symbolique et numérique). Ils se constituent de deux composants : un graphe métrique et un graphe d'Allen. Cependant, ces deux graphes ne sont pas lié par des contraintes internes; plutôt, ils sont traités séparément et les relations entre les objets sont gérés par un contrôle externe. Pour pallier à cette faiblesse, Kautz et Ladkin proposent un algorithme qui résout chaque objet de manière indépendante et ainsi, faire circuler l'information entre les deux graphes en utilisant une traduction qualitative-quantitative, jusqu'à ce qu'un point fixe est atteint. L'approche de Meiri possède deux avantages par rapport à celui de Kautz et Ladkin. D'abord, toutes les informations sont regroupées dans un seul graphe. Ensuite, on établit des limites plus restreintes pour des tâches de raisonnement variées. Par exemple, afin de convertir un graphe particulier à une forme de consistance de chemin, l'algorithme de Kautz et Ladkin peut prendre $O(n^2)$ transmissions d'information qui donne une complexité de $O(n^5R^3)$, comparé à $O(n^3R^3)$ avec le modèle de Meiri. Il y a donc un avantage au niveau de l'efficacité algorithmique.

2.5 Conclusion du chapitre

Dans ce chapitre, nous avons présenté une introduction aux principaux outils formels du raisonnement temporel, utilisés en intelligence artificielle.

La première section de ce chapitre a permis d'explorer les modèles de représentation symbolique, qui permettent de modéliser et de raisonner sur des intervalles de temps relatifs et qui offrent une grande expressivité. Dans cette partie, une attention particulière a été porté sur la théorie proposée par Allen [Allen, 1983], qui se veut l'une des théories fondatrices en ce qui concerne la représentation de connaissances temporelles. Celle-ci est de plus très utilisée dans le contexte de la reconnaissance d'activités. Au terme de cette première section, nous avons pu constater la puissance expressive de ce type d'approches, mais aussi pu constater les défis liés à la complexité algorithmique souvent associée à leur mise en œuvre concrète.

Le second volet de chapitre traitait des modèles de représentation de la connaissance temporelle qui s'appuient sur une approche dite numérique. Ces approches permettent d'introduire une quantification du temps, des durées, etc., par rapport aux événements dont la connaissance est décrite, au lieu de seulement traiter d'une façon relative et sans référentiel réel les relations temporelles pouvant exister entre les événements liés à une activité. Dans cette partie du chapitre, une attention particulière a été portée à la théorie de Dechter [Dechter, 1991], que nous avons exploité dans la contribution présentée au chapitre 4. Au terme de cette section, nous avons pu voir comment ce type d'approches, qui se veut simple et efficace du point de vue algorithmique, permet de tenir compte des contraintes temporelles liées aux durées et délais qui ont été illustrés en introduction du mémoire.

Enfin, la troisième et dernière partie de ce chapitre proposaient un bref survol des approches de représentation hybride incorporant à la fois une modélisation symbolique et numérique du temps. Ces approches, bien qu'intéressante, ne sont pas très utilisées dans la littérature et comportent une complexité accrue par rapport aux approches purement numérique. Au terme de cette investigation des modèles temporels existants, il apparaît clair que, dans notre cadre applicatif, l'utilisation d'une approche numérique était à préconiser, car celle-ci s'avérait suffisante pour répondre à nos besoins et moins complexe qu'une approche hybride.

Cette introduction aux principaux outils formels permettant de traiter l'aspect temporel était nécessaire afin que le lecteur puisse apprécier pleinement la suite de ce mémoire. Le prochain chapitre s'inscrit en continuité avec celui-ci et présente un état de l'art ciblé sur les principales approches de reconnaissance d'activités tenant compte de l'aspect temporel et qui présentent un intérêt en rapport à la contribution proposée dans ce mémoire.

CHAPITRE 3

La reconnaissance d'activités temporelles

3.1 Introduction

La reconnaissance d'activités a été un sujet actif de l'intelligence artificielle pendant les dernières décennies [Carberry, 2001]. Cependant, c'est seulement plus récemment, avec l'arrivée du concept d'informatique diffuse (omniprésente) et avec le développement d'environnement intelligent, que cette problématique est vraiment devenue une pièce maîtresse dans le grand puzzle de l'intelligence ambiante [Roy et al., 2010]. Dans ce type de contexte applicatif, l'information temporelle disponible (durée d'une action, laps de temps entre deux actions, etc.) se révèle d'une importance capitale, en particulier pour minimiser les hypothèses plausibles et identifier différents type d'erreurs de réalisation. En ce sens, ce chapitre propose un état de l'art ciblé portant sur la reconnaissance d'activités en générale et sur les principales approches de reconnaissance qui exploitent l'aspect temporel [Kautz, 1991] [Weida, 1993] [Jakkula et Cook, 2006]. La première partie du chapitre permettra au lecteur de comprendre d'une façon plus claire les multiples facettes caractérisant le problème de reconnaissance d'activités, notamment dans le contexte de l'habitat intelligent. Ainsi, cette section expliquera les notions de base liées au sujet et montrera comment ce problème a évolué depuis les 30 dernières années pour en arriver à sa forme actuelle dans un cadre d'intelligence ambiante. La seconde partie du chapitre sera consacrée à la présentation des principaux modèles temporels de reconnaissance proposés dans la littérature et dressera un bilan de ces approches qui permettra de mieux situer notre contribution par rapport à celles-ci.

3.1.1 Définition d'une activité de vie quotidienne (AVQ)

Lorsqu'on parle de reconnaissance d'activités, la première question à ce poser, et non la moindre, est : à quoi le concept d'activité réfère t-il exactement? Par ailleurs, dans notre contexte applicatif lié à l'assistance cognitive, on parle plus précisément de reconnaître les « activités de la vie quotidienne » d'une personne en perte d'autonomie. La notion d'activités de la vie quotidienne (AVQ) a été définie par le Dr. Sydney en 1963 [Katz et al., 1963] comme étant un ensemble d'activités qu'un individu exécute à titre de routine pour prendre soin de lui-même, par exemple, le fait de s'habiller, effectuer sa toilette, la préparation de repas, etc. Les professionnels de santé réfèrent souvent à la capacité ou à l'incapacité d'une personne à exécuter une AVQ comme une mesure du statut fonctionnel de celui (son niveau d'autonomie). Cette mesure est utile pour évaluer le degré de dégénérescence cognitive d'un patient ainsi que pour évaluer le type de support dont il aura besoin [Giovannetti et al., 2002]. En somme, l'AVQ représente un ensemble d'activités communes qu'une personne normale est supposée être apte à exécuter par elle-même afin qu'on puisse la qualifier d'autonome. On peut distinguer trois types d'AVQ : basique, instrumentale et augmentée [Rogers et al., 1998] :

- Les *activités de vie quotidienne basique* (AVQB) consistent en des tâches fondamentales et nécessaires pour combler les besoins primaires de la personne. Par exemple se déplacer du lit vers une chaise, aller à la salle de bain, marcher à pied, etc. Ces activités sont composées seulement de quelques étapes et n'exigent pas de planification réelle.
- Les *activités de vie quotidienne instrumentales* (AVQI) nécessitent des manipulations d'objets et une certaine forme de planification. ne sont pas reliées au fonctionnement fondamental, mais permettent à une personne d'être relativement autonomes chez soi ainsi dans une communauté. À titre d'exemple, ces activités peuvent être le fait de préparer un repas, prendre des médicaments, faire des achats d'une épicerie ou d'un magasin, utiliser un téléphone, gérer son argent, etc. Les AVQ instrumentales sont plus complexe et exigent un niveau plus haut de planification et meilleur jugement qu'un AVQ basique.
- Les *activités de vie quotidienne augmentées* (AVQA) correspondent aux tâches exigeant une adaptation de la part de l'individu à cause de la nature de l'environnement.

Par exemple, rappeler une personne à partir d'un téléphone qui garde les dix derniers appelants, est une tâche qui exige une adaptation (au nouveau téléphone).

Dans la littérature concernant les technologies d'assistance dans les habitats intelligents [Giroux et al., 2009; Heigh et al., 2006; Patterson et al., 2007], les chercheurs réfèrent souvent sans distinctions à ces trois types sous le vocable générique d'AVQ. Cependant, la plupart des systèmes de reconnaissance développés pour assister les gens chez eux se focalisent seulement sur les AVQ basiques et instrumentales.

3.1.2 Les origines de la reconnaissance d'activités

À la base, le problème lié à la reconnaissance d'activité a été identifié par la communauté œuvrant en intelligence artificielle et a été longtemps référé sous l'appellation « reconnaissance de plan ». Le problème de la reconnaissance de plan a été défini pour la première fois par Schmidt [Schmidt, 1978], qui a synthétisé celui-ci par la tâche de « ... prendre en entrée une séquence d'actions effectuée par un acteur et d'inférer son but en organisant cette séquence d'actions en une structure de plan ». Ainsi, on comprend que l'expression « reconnaissance de plan » réfère au fait que l'on suppose, au départ, l'existence d'une structure de plan (c'est à dire un ensemble d'actions organisées dans le temps), préalablement planifié par l'entité observé (dans notre cas un patient), et constituant le résultat que l'agent observateur (dans notre cas l'habitat intelligent) cherche à reconnaître.

En se référant à la littérature sur ce sujet, on se rend compte que cette vision préalable du problème est un héritage historique des premiers systèmes experts, qui ont été à l'origine utilisés pour résoudre des problèmes de planification [Waern et al., 1996]. La problématique liée à la planification constitue aussi un défi bien connu dans la communauté de l'intelligence artificielle, que l'on peut considérer comme l'inverse de la reconnaissance d'activités [Russel et Norvig, 2003]. Cette difficulté consiste, pour un agent, à identifier une suite d'actions (un plan) qui lui permettra, au terme de son exécution, d'atteindre un objectif désiré [Georgeff, 1987]. Par conséquent, on peut voir le problème de la reconnaissance d'activités comme l'opération inverse, dans laquelle un autre agent (l'observateur), ignorant l'objectif initial planifié par l'agent observé, a pour mandat d'identifier l'objectif et le plan d'action mis en branle par celui-ci en inférant, à partir des actions observées, l'ensemble des activités possiblement en cours.

3.1.3 Classification des différents types de reconnaissance d'activités

Au fil du temps, la problématique liée à la reconnaissance d'activités s'est précisée pour se subdiviser en plusieurs catégories. La classification la plus utilisée dans la littérature est celle proposée par Cohen [Cohen et al., 1982], puis récemment complétée par Geib [Geib et Goldman, 2001]. Cette classification permet de scinder les types de reconnaissance en rapport à la caractérisation de la relation qui existe entre l'agent observé et son observateur. Cette relation permet de diviser la reconnaissance d'activités en trois sous-catégories : la reconnaissance dite *communicative* (« intended »), la reconnaissance dite *contradictoire* (« adversial ») et la reconnaissance à l'insu (« keyhole »). Chacun de ces types de reconnaissances pose un postulat de base différent concernant la relation existante entre l'agent observateur et l'entité observée qui guidera le processus de reconnaissance et permettra de présumer certaines choses. Cette classification est systématiquement utilisée afin d'expliquer le point de vue adopté en fonction du contexte de reconnaissance et constitue un élément important lorsqu'on souhaite construire un modèle de reconnaissance approprié pour un contexte spécifique comme c'est le cas ici.

- **Reconnaissance d'activités dite communicative :** Dans un contexte de reconnaissance communicative, on suppose que l'agent observé est explicitement au courant de ce fait et, par conséquent, qu'il tentera d'adapter délibérément son comportement de manière à faciliter le processus de reconnaissance [Kautz, 1991]. Par conséquent, cette forme de reconnaissance suppose un effort coopératif direct de la part de l'entité observée. Cette catégorie de reconnaissance a été introduite par Cohen [Cohen et al., 1982] dans le but de situer le problème de la reconnaissance d'activités dans un cadre particulier de traitement du langage naturel, dans lequel il travaillait. Quelques chercheurs ont supposé, en se basant sur la présomption de coopérativité, que l'agent observateur pourrait directement demander à l'acteur de l'aider afin d'obtenir des détails, en cas d'incertitude, sur la tâche en cours [Leshet al., 1998]. Dans le contexte applicatif qui nous intéresse, il n'est pas réaliste de situer le problème de reconnaissance dans ce cadre communicatif. À titre d'exemple, dans le cas d'un environnement intelligent qui permet d'assister un malade Alzheimer dans ses tâches quotidiennes, il est clairement peu réaliste de supposer la capacité du patient à soutenir un effort coopératif en adaptant son comportement pour influencer positivement le processus de reconnaissance. De plus, on ne peut pas laisser le système demander des détails au patient en cas d'incertitude car cette action va conduire à

une charge cognitive accrue, qui est peu convenable dans un contexte d'assistance cognitive.

- **Reconnaissance d'activité dite contradictoire** : Cette catégorie a été introduite récemment par Geib [Geib et Goldman, 2001] comme étant une contrepartie de la première catégorie, c'est-à-dire se situant à l'inverse de la reconnaissance communicative. Dans un contexte de reconnaissance contradictoire, au lieu de présumer que l'agent observé va contribuer positivement au processus de reconnaissance, on suppose l'inverse, c'est-à-dire que l'agent observé est hostile et essaiera de compromettre le processus de reconnaissance, dans un geste anti-coopératif [Mao et Gratch, 2004]. En d'autres termes, l'agent observateur est considéré comme un ennemi et il va donc, volontairement, essayer d'exécuter des actions qui sont incohérentes avec ses buts, d'une manière à amener l'observateur à inférer de fausses conclusions concernant son comportement. Ce type de reconnaissance est bien adapté à un contexte applicatif militaire [Albrecht et al., 1998]. Cependant, il ne correspond pas bien à un contexte d'assistance cognitive. Même si on doit considérer le fait qu'un patient ayant un déficit cognitif peut exécuter des actions erronées et se tromper dans la réalisation des ces activités, comme c'est le cas généralement dans un contexte contradictoire, il reste que ces erreurs ne sont pas exécutées délibérément, et donc il n'y pas de volonté intrinsèque de nuisance. Les erreurs commises sont plutôt la conséquence des symptômes du patient et non pas le résultat d'une stratégie bien planifiée pour mettre en péril le processus de reconnaissance.
- **Reconnaissance d'activité à l'insu** : Cette dernière catégorie caractérise l'hypothèse que l'agent observé n'est pas au courant de ce fait ou qu'il n'en tient pas compte et ainsi, qu'il n'essayera pas d'influencer le processus de reconnaissance ni positivement, ni négativement [Cohen et al., 1982]. En introduisant cette catégorie, Cohen a voulu définir un cadre générique pour la reconnaissance d'activités, dans lequel on ne fait aucune hypothèse concernant l'effort coopératif déployé par l'agent observé. De là, Cohen fait une analogie, d'où le nom anglophone « keyhole » de la catégorie émane, avec une personne observée, à l'intérieur d'une pièce, par le trou de la serrure de la porte. Cette personne agit dans la pièce de manière naturelle, sans se soucier de l'observation et sans essayer d'aider ou de nuire à une personne potentielle qui regarderait par le trou de cette serrure. Dans le contexte applicatif lié à l'assistance cognitive au sein d'un habitat intelligent, le problème de la reconnaissance d'activités s'inscrit dans cette catégorie

[Bouchard, 2006], car on ne peut certainement pas présumer la collaboration ni la nuisance délibérée du patient.

3.1.4 Reconnaissance d'activités dans un environnement intelligent

Depuis l'arrivée des catégories de Cohen et Geib, la définition de la reconnaissance d'activité à l'insu a été reformulée plusieurs fois, notamment par [Goldman et al., 1999; Boger et al., 2005; Bouchard et al., 2007; Patterson et al., 2003], afin de préciser ses particularités dans un contexte de reconnaissance ambiante au sein d'un environnement intelligent. Dans ces nouvelles définitions raffinées, on note une tendance claire à vouloir rendre explicite la notion d'environnement ambiant, qui se réfère au contexte de reconnaissance, et de lier d'une manière formelle cette notion avec les défis soulevés par la problématique. Par exemple, Goldman [Goldman et al., 1999] présente la reconnaissance d'activités comme étant « ... processus d'inférence des plans d'un agent à partir des observations faites de ses actions ». Cette définition peut paraître très similaire à celle de Schmidt, cependant une différence subtile mais importante rejaillit dans le fait qu'on explicite clairement la distinction entre la notion d'action exécutée par l'agent, élément qui n'est pas directement observable, et les observations (perceptions) qui peuvent être directement détectées par le système de reconnaissance. Plus récemment, Patterson [Patterson et al., 2003] précisa une fois de plus la problématique en rapport à l'habitat intelligent en la décrivant comme « ... la tâche d'inférer l'activité exécutée par l'entité observée à partir de données fournies par des capteurs de bas niveau ». Cette nouvelle vision très "terre à terre" du problème est largement partagée par une grande partie des chercheurs travaillant dans la communauté des habitats intelligents et adhérant au paradigme récent de l'informatique diffuse [Weiser, 1991], qui prône la mise en place d'environnements améliorés avec des processeurs miniaturisés, des agents logiciels ambiants communiquant entre eux, et des capteurs multimodaux qui sont intégrés dans les objets quotidiens afin de les rendre le moins visible possible pour l'occupant. À partir de ce point de vue, on peut situer le contexte applicatif faisant l'objet de ce mémoire en précisant que l'agent observé évolue à l'intérieur d'un environnement avec les capteurs de bas niveau, qui sont déclenchés par les résultats de l'interaction entre l'agent observé et son environnement, et où l'agent observateur peut percevoir ces déclencheurs.

3.1.5 Reconnaissance d'activités intégrant l'aspect temporel

Une activité (un plan d'action) peut être vue comme une succession d'événements liés à des paramètres qui décrivent un système [Pollack, 2005]. Parmi ces paramètres, le temps représente

une pièce maîtresse permettant, comme nous l'avons vu en introduction, de déduire un certain nombre d'informations utiles au processus de reconnaissance [Jakkula et Cook, 2006]. Les propriétés discriminantes du temps prennent alors un statut particulier, on parle alors de *plan temporel*. Ainsi on peut avoir des prédictions datées par rapport au processus observé. Le processus de reconnaissance s'appuie sur une librairie de plans qui regroupe la description (la connaissance) à propos de toutes les activités susceptibles d'être observées. La représentation de la connaissance reliée à une activité dépend du modèle et du mécanisme de reconnaissance choisis. La séquence d'actions observées est comparée aux activités décrites dans la librairie, en fonction du modèle d'inférence sélectionné, afin d'identifier les hypothèses plausibles et de faire un choix parmi celles-ci. Pour effectuer adéquatement cette opération on a besoin de satisfaire les conditions suivantes [Roy et al., 2010]:

- On doit posséder un moyen efficace de comparer les instances d'action observée, formant une séquence, aux activités incluses dans la librairie, ce qui signifie du même coup que leur représentation doivent être compatibles;
- Les relations existantes entre les actions observées doivent pouvoir être efficacement comparées aux contraintes imposées à la description d'une activité dans la librairie afin de déterminer si cette activité les satisfait, en quel cas celle-ci pourra être retenue comme une hypothèse plausible. Dans le contexte qui nous intéresse, ces relations sont principalement temporelles.

Dans le cadre de ce mémoire, nous proposons une approche qui utilise une représentation de la connaissance (donc de la librairie d'activités) basée sur la logique de description [Baader et al., 2007]. La relation de subsumption de cette logique nous permet, comme nous le verrons au chapitre suivant, de satisfaire à la première condition en exploitant celle-ci pour faire efficacement le *matching* entre les actions observées et les activités de la librairie. Par la suite, l'extension que nous proposons au modèle de représentation du temps de Weida [Weida, 1993] nous permettra de répondre à la seconde condition et de comparer efficacement l'ordonnancement, les délais et les durées temporelles observées aux contraintes imposées aux descriptions d'activités incluses dans la librairie. Cependant, avant de passer à la présentation de notre modèle, il est important de faire un tour d'horizon des travaux connexes sur la reconnaissance d'activités ayant tenté, avant nous, de répondre à ces conditions.

3.2 Les principaux travaux sur la reconnaissance temporelle

Le domaine de la reconnaissance d'activités est bien établi et foisonne d'un nombre imposant de travaux [Carberry, 2001] qui sont reliés de façon plus ou moins pertinente à l'approche que nous proposons. Ainsi, il est impensable et surtout, non nécessaire, de parcourir l'ensemble de ces travaux à l'intérieur de ce mémoire. C'est pourquoi, cette section se focalisera uniquement sur quelques travaux ciblés, qui constituent en quelques sortes les approches clés inhérentes à la reconnaissance d'activités dans une perspective temporelle. Ceux-ci permettent de synthétiser en grande partie la tendance de la plupart des modèles observés dans le domaine. Parmi ces travaux, on retrouve l'approche fondatrice de Kautz [Kautz, 1986] [Kautz, 1991] qui s'appuient sur la logique du premier ordre, les travaux de Jakkula et Cook [Jakkula et Cook, 2006] qui se basent sur les relations temporelles d'Allen pour pouvoir détecter les anomalies d'exécution d'activités dans un habitat intelligent, et les travaux précurseurs de Weida [Weida, 1993], qui présentent un modèle de reconnaissance d'activités s'appuyant sur un système temporel existant [Kautz et Ladkin, 1991] afin de créer une librairie de plan temporel. Dans les prochaines sections, nous parcourons ces différentes approches et positionnerons les choix effectuées dans le cadre du modèle proposé par rapport à ces travaux.

3.2.1 Approche de Kautz [Kautz, 1986; Kautz, 1991]

La théorie proposée par Kautz [Kautz, 1986] consiste à exploiter la logique du premier ordre afin de formaliser le processus d'inférence de la reconnaissance d'activités. Elle suppose que l'agent observateur démarre avec une librairie constituée de schémas de plans (description des activités), tels que présentés à la Figure 2.13, encodée à l'aide d'axiomes. Dans cette représentation logique, une action possèdera un certain nombre de sous-types (une manière particulière de spécialiser cette action) et pourra avoir une décomposition (une procédure à suivre pour exécuter l'action), l'ensemble formant un graphe hiérarchique d'abstraction/décomposition. Dans ce modèle, les relations d'abstractions et de décomposition permettant de former la hiérarchie sont encodées à l'aide d'axiomes de type (ABS) et (DEC), qui sont chacun spécifié sous la forme suivante:

$$(ABS) \quad \forall x(\alpha(x) \rightarrow \alpha'(x))$$

$$(DEC) \quad \forall x(\alpha(x) \rightarrow (\alpha_1(step_1(x)) \wedge \dots \wedge (\alpha_n(step_n(x)) \wedge \kappa))$$

où α correspond à un événement au sens large (une action ou une activité), α' correspond à une abstraction directe d'un autre événement α , spécifiée grâce à une relation de type (ABS), et α_i

correspond à la $i^{\text{ème}}$ étape (action) dans la décomposition d'une certaine activité (événement) α , encodée en utilisant une relation de type (DEC). Par exemple, l'axiome (ABS) $\forall x \text{ Make_Pasta_Dish}(x) \rightarrow \text{PrepareMeal}(x)$ décrit formellement qu'une relation d'abstraction existe entre « Make pasta dish » et « Prepare Meal » dans l'exemple de la Figure 3.1. L'événement *End* caractérise quant à lui la racine de la hiérarchie et est utilisé pour contrôler le processus inférentiel de reconnaissance. Enfin, le symbole κ dénote une conjonction de contraintes (pré-conditions, relations temporelles entre actions, etc.). C'est grâce à cette conjonction de contrainte que Kautz introduit la prise en compte du temps, mais nous y reviendrons dans les sections ultérieures. La librairie d'activités est présumée correcte, c'est-à-dire qu'il n'existe aucune erreur dans la définition des relations d'abstraction et de décomposition, et complète, c'est-à-dire qu'elle est supposée contenir l'ensemble des actions que le système peut être amené à observer et toutes les activités qui sont possibles.

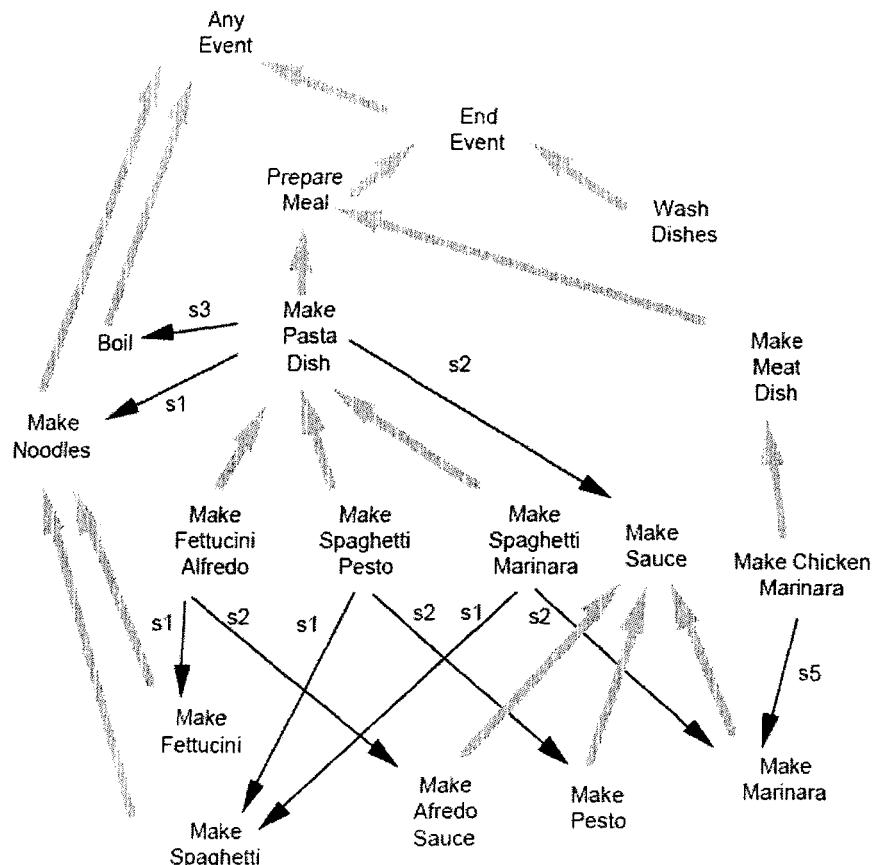


Figure 3.1. Exemple de librairie d'activités de cuisine selon Kautz [1991]

À partir d'une librairie d'activités encodées de cette façon, Kautz définit un processus de reconnaissance basé sur quatre règles d'inférence qui, à partir d'une observation introduite sous la forme d'une assertion logique, permettent d'extraire un modèle minimal d'interprétation de cette librairie (un sous-ensemble couvrant). Le résultat du processus d'inférence consiste alors en une disjonction d'hypothèses (un ensemble disjoint d'activités plausibles) correspondants à aux activités incluses dans l'arbre minimal couvrant. En fait, ce processus d'inférence s'inspire de la théorie de la circonscription de McCarthy [McCarthy, 1980]. Les règles d'inférence logique utilisées par ce processus de reconnaissance sont les suivantes :

- (EXA) $\forall x(\alpha(x) \rightarrow \alpha_1(x) \vee \dots \vee \alpha_n(x))$
- (DJA) $\forall x(\neg\alpha_1(x) \vee \neg\alpha_2(x))$
- (CUA) $\forall x(\alpha(x) \rightarrow END(x) \vee (\dots \vee \exists y(\alpha'(x) \wedge (x = step_i(y))) \vee \dots))$
- (MCA_n) $\forall x_1 \dots \forall x_n (END(x_1) \dots END(x_n) \rightarrow \vee_{i,j}(x_i = x_j))$

L'hypothèse d'exhaustivité (EXA) réfère à une règle d'inférence spécifiant qu'un événement observé (une action) x , appartenant à un certain type d'événement α , est nécessairement aussi une instance de l'une des spécialisations possibles de celui-ci $\{\alpha_1 \dots \alpha_v\}$. Par exemple, si l'on observe un certain événement x de type « Make pasta dish », on peut alors inférer grâce à cette règle que cet événement appartient à l'une des spécialisations suivantes : « Make fettuccini alfredo », « Make spaghetti pesto » ou « Make spaghetti marinara ». En complément, la règle de disjonction (DJA) spécifie que cet événement est nécessaire l'instance d'un et d'un seul de ces sous-types. Par conséquent, x ne peut pas être du type « Make fettuccini alfredo » et être aussi du type « Make spaghetti pesto ». Ainsi, une assertion permettant d'inférer l'une de ces hypothèses nous permettra aussi d'infirmer l'autre. Ces deux règles sont cependant limitées par le fait qu'elles perdent leur validité si la librairie est incomplète (l'inverse est présumé par Kautz) ou si l'héritage multiple est admis (ce que Kautz a banni). La règle de composition (CUA) implique quant à elle qu'un événement observé doit être une instance d'une « intention », c'est-à-dire à un événement (activité) de type *End*, ou alors être un composant (faire partie de la décomposition) d'un tel événement. Par exemple, une événement observé x peut être du type « Make meat dish », lequel est une intention car un sous-type de l'événement *End*. Si x n'appartient pas à un sous-type de *End*, alors il doit correspondre à un composant (par exemple « Make marinara ») d'une des ses spécialisations. Comme on peut le constater, cette règle présume également que la librairie est complète. Finalement, la quatrième règle introduite par Kautz concerne la cardinalité

minimum (MCA) et se fonde sur la supposition que si deux observations peuvent être reliées à un même événement de type *End*, alors ces observations émane nécessairement de la même intention de la part du sujet observé. En d'autres mots, en cas de doute, Kautz conserve l'explication la plus simple, c'est-à-dire celle qui implique le moins d'événements possibles.

À partir de ces règles d'inférence préétablies, d'une hiérarchie encodée telle que nous venons de le décrire, et d'une ou plusieurs observations fournies sous forme d'assertion prédicative du type « *Make_mainara(x)* », le principe de la reconnaissance se base sur une séries d'inférences logiques déductives se déroulant approximativement de la manière suivante (Kautz, 19991): (i) à chaque assertion d'une observation, on utilise la règle de composition (CUA) afin d'identifier les plans possibles issus de la hiérarchie contenant l'événement de base observé, ce qui nous donnera plusieurs activités possibles dispersées à plusieurs endroits dans la hiérarchie (ii) ensuite, on applique l'axiome d'abstraction (ABS) de façon récursive, afin de remonter, à partir de chacun de ces plans possibles, jusqu'à la racine de type *End*, (iii) par la suite, nous redescendrons la hiérarchie en tentant d'éliminer des hypothèses en utilisant la règle disjonction (DJA), (iv) on validera, lors de cette descente, les contraintes de décomposition (DEC) pour chaque hypothèse possible ainsi que les contraintes k définies dans la description logiques des activités, (v) finalement, en cas d'observations multiples, on appliquera la règle de minimisation des modèles couvrants (MCA) afin de réduire au minimum le nombre de plans pouvant expliquer les événements observés. Ce processus permettra d'extraire un sous-arbre à partir de la hiérarchie de départ correspondant au modèle minimal couvrant permettant d'expliquer la situation observée.

3.2.1.1 Reconnaissance temporelle selon Kautz

Comme nous l'avons mentionné brièvement, Kautz [Kautz, 1991] introduit partiellement le concept de temps dans son approche à travers les contraintes κ imposées dans la décomposition des étapes d'une activité avec les axiomes de type (DEC). Pour ce faire, Kautz s'appuie directement sur la théorie d'Allen [Allen, 1983] et donc, l'ensemble des paires d'étapes (actions) formant une activité peuvent donc se voir imposer des contraintes quant à leur ordonnancement temporel uniquement, suivant les treize relations possibles. Afin de pouvoir vérifier si la description d'une activité de librairie concorde, au niveau des contraintes d'ordonnancement temporelles, avec une suite d'actions observées, Kautz associe à chaque assertion d'une observation un intervalle de temps qualifié d'imprécis. Cet intervalle est de la forme $\langle a, b, c, d \rangle$ et indique que l'événement débute entre un certain temps a et un certain temps b et se termine entre un certain temps c et un certain temps d . Cet élément diverge de la théorie d'Allen au

niveau formel et nécessite un certain nombre d'axiomes pour effectuer la propagation de contraintes, dont voici un exemple :

Si $T_1 = \langle a, b, c, d \rangle$ et $T_2 = \langle e, f, g, h \rangle$ et T_1 débuté par T_2

Alors $T_1 = \langle \max(a, e), \min(b, f), \max(c, g), d \rangle$

Dans ce premier apport à la reconnaissance temporelle, Kautz s'est basé sur un algorithme de type CSP pour la propagation des contraintes. Il consiste à utiliser les règles de décompositions d'Allen, jusqu'à ce qu'il n'y a plus de modification, permettant ainsi, d'éliminer certains plans candidats ne vérifiant pas les contraintes temporelles.

3.2.1.2 Apport de Song et Cohen à la théorie de Kautz

Song et Cohen [Song et Cohen, 1991] ont proposé d'enrichir l'approche de Kautz au niveau de l'aspect temporel en introduisant une règle d'inférence supplémentaire considérant que toute action est totalement déterminée et bornée dans le temps par ses sous-actions, c'est-à-dire qu'elle doit nécessairement débuter au moment où commence sa première sous-action et se terminer à la fin de la toute dernière. Voici un exemple qui montre l'impact de cette nouvelle hypothèse : soit l'action A qui se décompose en deux sous-actions a_1 et a_2 , avec a_1 (avant) a_2 , et a_1 (débute(s), finit(f), pendant(d) ou égal(=)) A, en supposant qu'on a la même relation entre a_2 et A. Si l'on suppose également qu'une des deux sous-actions, au moins, débute au même temps que A et que l'une des deux sous-actions, au moins, se termine en même temps que A, on obtient alors des contraintes plus précise. Par conséquent, a_1 est la seule sous-action qui puisse débuter en même temps que A, étant la première de la séquence, et a_2 la seule qui puisse se terminer en même temps que A. Comme le montre la figure 3.2, l'algorithme d'Allen n'est pas assez restrictif et pour inférer ce fait et demeure insuffisant. Dans le cas où on aura plus que deux sous-actions, on fait un retour au sous-cas précédent en introduisant des sous-actions fictives [Eude, 1999].

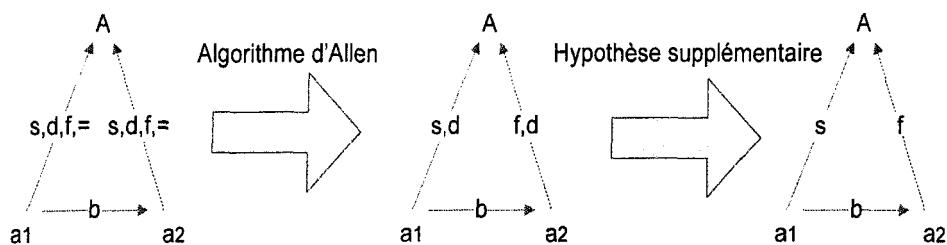


Figure 3.2. Effet de l'hypothèse de Song et Cohen sur la propagation [1991]

Suite à cet apport de Song et Cohen avec leur hypothèse supplémentaire, l'algorithme de reconnaissance de Kautz devient plus efficace dans le traitement et la vérification de la validité des relations d'ordonnancement temporelles. L'approche de Kautz, avec ou sans l'apport de Song et Cohen, reste cependant limité en ce qui a trait à l'aspect temporelle. Comme nous l'avons mentionné en introduction, sa principale faiblesse réside dans le fait qu'elle tient uniquement compte de l'aspect qualitatif des contraintes temporelles liées à l'ordonnancement des paires d'actions dans l'activité, en négligeant l'aspect quantitatif lié aux durées des actions et aux délais entre ces actions.

3.2.2 Approche de Jakkula et Cook [2006]

Jakkula et Cook [Jakkula et Cook, 2007] œuvrent directement dans le domaine applicatif lié à l'habitat intelligent. L'approche de reconnaissance qu'ils proposent s'appuie sur une architecture multi-agents [Wooldridge, 2002] dans laquelle les agents perçoivent l'état de l'environnement à partir des données senseurs et tentent, d'une manière rationnelle, de maximiser le confort et la productivité de l'occupant en se basant sur un système de reconnaissance du comportement. Dans cette approche, la modélisation des contraintes temporelles se fonde, une fois encore, sur la logique temporelle d'Allen [Allen, 1983]. Ce modèle de reconnaissance d'activités exploite l'algorithme de reconnaissance et de prédiction *Active Lezi* [Gopalratnam, 2003], utilisé dans le domaine du forage de données ou *datamining* [Witten, 2005]. Cet algorithme permet d'apprendre les *patterns* séquentiels, en se basant sur les principes de la théorie de l'information afin d'apprendre, à partir de l'historique des séquences enregistrées, les liens temporels existants entre différents événements qui se déclenchent fréquemment. Par exemple, si l'historique des événements montrent qu'habituellement, après qu'une tasse ait été sortie de l'armoire, la bouilloire s'active peut de temps après, le système inférera une relation temporelle, tirée des treize d'Allen, entre ces deux événements de type *avant*, stipulant que l'événement *prendre tasse* s'effectue avant *activer bouilloire*. Les relations temporelles ainsi inférées sont par la suite utilisées pour la reconnaissance des anomalies comportementales du résident. En reprenant l'exemple précédent, cela signifie qu'après la période d'apprentissage, si l'événement *activer bouilloire* s'effectue sans la présence de l'événement *prendre tasse*, ou bien juste avant, ce comportement sera considéré anormal puisqu'il diverge des routines habituelles qui ont été apprises. Suivant cette approche, la détection d'anomalies est donc plus précise quand celle-ci se base sur les actions qui sont plus fréquentes et par conséquent plus prévisibles. Sommairement, l'algorithme d'apprentissage prend en entrée un fichier d'événements correspondant à

l'historique acquis par les capteurs. Chacun de ces événements est associé avec un temps de déclenchement et un horaire. L'algorithme filtre d'abord le fichier pour en ressortir une liste de séquences d'événements qui se produit fréquemment dans une même fenêtre de temps. Ensuite, chaque séquence fréquente est analysée afin d'en extraire les relations temporelles existantes entre chacune des paires d'actions impliquées dans la séquence. Enfin, toujours en se basant sur l'historique, l'étape finale de l'algorithme consiste à calculer la probabilité qu'un événement se produise moyennant l'apparition d'un autre de la même séquence. Cette étape est à la base du système de détection d'anomalies.

3.2.2.1 Exploitation des relations temporelles pour la prédiction

Selon Jakkula et Cook [Jakkula et Cook, 2007], seul un sous ensemble des relations temporelles d'Allen représentent une utilité pour prédiction et la détection d'anomalie. Ce sous-ensemble est formé des relations : *before*, *contains*, *overlaps*, *meets*, *starts*, *started-by*, *finishes*, *finishes-by*, et *equals*. Les relations temporelles restantes (*after*, *during*, *overlapped-by* et *meets-by*) ne sont donc pas prises en compte dans le processus de reconnaissance et d'apprentissage. De façon sommaire, le processus inférentiel proposé par Jakkula et Cook se déroule, une fois l'apprentissage terminé, comme suit. Pour chaque événement Z , on calcule la probabilité que celui-ci a de se produire (dans ce cas, on vérifie la borne inférieure de l'intervalle de temps de cet événement) et on effectue le renforcement de cette probabilité en se basant sur la présence d'autres événements observés, par exemple Y , qui ont une relation temporelle inférée par apprentissage avec Z . Pour ce faire, on considère la probabilité d'apparition de Z donnée par la détection du début de l'intervalle temporel de l'événement Y . La formule qui permet de calculer cette probabilité de l'événement Z est donnée par l'équation (1) ci-dessous. Il faut noter que cette équation est basée sur la fréquence d'observation entre Y et Z . La nécessité d'additionner les valeurs des relations temporelles observées est de trouver la probabilité totale pour laquelle les événements Z et Y arrivent en même temps et que la relation qui a la valeur la plus grande est la relations la plus probable qui peut être associé entre les événements Z et Y .

Équation (1) :

$$P(Z|Y) = | \text{Before}(Z, Y) | + | \text{Contains}(Z, Y) | + | \text{Overlaps}(Z, Y) | + | \text{Meets}(Z, Y) | + | \text{Starts}(Z, Y) | + | \text{StartedBy}(Z, Y) | + | \text{Finishes}(Z, Y) | + | \text{FinishedBy}(Z, Y) | + | \text{Equals}(Z, Y) | / |Y|$$

Jakkula et Cook introduisent ensuite un algorithme de prédiction (reconnaissance) basée sur l'*ActiveLeZi* [Gopalratnam, 2003]. Pour ce faire, ils considèrent que n'importe quelle séquence d'événements fréquents qui ont été identifiés (les activités apprises) peut être modélisée par un processus stochastique. Ainsi, l'algorithme emploie des modèles Markov [Rabiner, 1990] pour prévoir le symbole (événement) suivant dans n'importe quel séquence stochastique d'une façon optimale. Dans leur récent travail, ils complètent le renforcement de leur approche pour qu'une action particulière utilise les informations incluant des relations temporelles.

3.2.2.2 Implémentation et validation de l'approche

L'approche de Jakkula et Cook a été implémenté et testé à l'intérieur de l'infrastructure d'habitat intelligent du laboratoire MavLab [Youngblood et al., 2005]. Le laboratoire consiste en une cuisine, des bureaux d'étudiant et une pièce facultative avec plus de 100 capteurs, utilisant principalement la technologie X10 qui permet de transmettre des données à travers le réseau électrique conventionnel d'une résidence. De plus, un générateur de données capteurs synthétiques a été créé pour valider cette approche. Le système de collecte de données consiste en un tableau de capteurs de mouvement, qui rassemblent des informations en utilisant les dispositifs X10 et le réseau de capteurs internes. L'ensemble des données est collecté en vue d'obtenir un échantillon sur deux mois. Jakkula et Cook ont développé un modèle de *pattern* d'un utilisateur qui consiste en un certain nombre d'activités différentes dans plusieurs pièces. Dans ce travail, ils ont produit un ensemble de données synthétiques contenant environ 4,000 événements sur deux mois d'activités. Les paramètres des ensembles de données réels et synthétiques sont donnés dans la table ci-dessous.

Ensemble de données	Nombre de jour	Nombre d'événements	Intervalles temporelles trouvées
Réel	60	17	1640
Synthétique	60	081	1738

Table 3.1 Paramétrage de l'ensemble de données utilisé pour trouver des contraintes

La première étape de l'expérimentation a permis de traiter les données telles qu'elles sont (à l'état brut) pour trouver les intervalles temporels. Ceci est réalisé en utilisant un outil simple qui marque le temps dans lequel l'événement s'est produit et basé sur l'état MARCHE/ARRÊT qui forme les intervalles. Ensuite, ces données sont transmises à un outil analyseur de temps qui

identifie les intervalles temporels basés sur les contraintes formées. Le pseudo code de cet outil analyseur de temps développé est décrit dans l'algorithme suivant :

```

Algorithme Analyseur-de-temps
    Entrées : marqueur-de-temps, état et nom-événement
    Répéter : Tantque [Evenement && Evenement+1 trouvé]
        Trouver les paires d'événements
        « MARCHE » ou « ARRÊT » dans les données pour
        déterminer le rang temporel.
        Lire l'événement suivant, trouver rang temporel.
        Identifier le type de relation entre une paire
        d'événements à partir des types de relations
        possibles. (voir Table 2.5).
        Enregistrer les types de relations et données.
        Incrémenter le pointeur d'événements
    Revenir jusqu'à fin des entrées.

```

Algorithme 3.1 Analyseur d'intervalles temporels

Une fois les relations temporelles inférées, on passe à l'étape suivante où on identifie les règles d'association qui peuvent être utilisées pour la prédiction. Pour ce faire, Jakkula et Cook exploite la plateforme *Weka*, qui fournie une implémentation de plusieurs algorithmes de *datamining* dont un de type *Apriori*, qui est utilisée ici afin de réduire itérativement le support minimal jusqu'à ce qu'il trouve le nombre exigé de règles. La table 3.2 récapitule les paramètres qui ont été mis en évidence et le nombre de règles produites avec une « confiance » minimale indiquée pour l'ensemble de données réels. La table 3.3 récapitule de la même manière l'ensemble de données synthétiques.

Exécuter #	Support minimal	Confiance minimale	Nombre de règles trouvées
1	0.00	0.5	100
2	0.01	0.5	006
3	0.02	0.5	002
4	0.05	0.5	001

Table 3.2 Paramétrage et génération de règles dans Weka pour l'ensemble de données réelles

Exécuter #	Support minimal	Confiance minimale	Nombre de règles trouvées

1	0.00	0.5	100
2	0.01	0.5	010
3	0.02	0.5	005
4	0.05	0.5	003

Table 3.3 Paramétrage et génération de règles dans Weka pour les données synthétiques

Une fois cette tâche complétée, l'étape finale concerne l'ajout des relations temporelles aux données en entrée en réalisant une étude comparative de la performance, avec et sans les règles. L'algorithme 3.2 spécifie l'amélioration apportée au niveau temporelle.

Algorithme Améliorer-prédiction

Entrée : Sortie de l'agent de prédiction ActiveLezi a, Meilleurs règles r, Données temporelles

Répéter

Si a != null

Répéter

Mettre r1 au premier événement dans la règle des relations

Si (r1==a) alors

Si (relation != « After ») alors

Calculer le renforcement (utiliser l'équation1)

Si renforcement > (Mean + 2 STD. Dev.) est noté

Alors

Rendre l'événement relié à r1 dans la meilleure

règle (autorité) comme la sortie suivante de l'agent de prédiction

Sinon

*Prendre l'événement prédit suivant et chercher la relation temporelle dans la base des relations temporelles

Si la relation temporelle est « After » alors passer à *

Jusqu'à ce qu'on retrouve plus de relation « After »

Calculer le renforcement

Si renforcement > (Mean = 2 STD. Dev)

Alors prédire ;

Sinon

Calculer le renforcement et si le renforcement > (Mean + 2Std. Dev.) alors prédire cet événement basé sur la relation ;

Fin si

```

Jusqu'à la fin des règles
Fin si
Revenir jusqu'à fin des entrées.

```

Algorithme 3.2. Prédition améliorée des règles temporelles

Ensemble de données	Exactitude %	Erreur %
Réelles(sans règles)	55	45
Synthétiques(sans règles)	64	36
Réelles(avec règles)	56	44
Synthétiques(avec règles)	69	31

Table 3.4 Comparaison des prédictions à base d'ActiveLezi avec et sans les règles temporelles

La table 3.4 affiche l'exactitude observée de la performance de prédition concernant les ensembles de données réelles et synthétiques. On note une amélioration de performance de prédition de 1.86 % pour ce qui est des données réelles et une amélioration de 7.81 % en ce qui concerne les données synthétiques. Il s'agit d'une amélioration de la prédition des activités du résident de l'environnement intelligent. La raison principale d'un taux d'erreur significatif est la petite quantité de données utilisées. La recherche de connaissances à base de règles temporelles constitue un nouveau secteur de recherche au niveau des habitats intelligents. On remarque que l'utilisation de relations temporelles fournie une nouvelle approche unique pour la prédition. Cette approche a été expérimentée sur de petits ensembles de données, en attendant l'exécution de l'algorithme de prédition sur des ensembles de données plus importantes. Cependant, pour une étude plus vaste, l'approche de Jakkula et Cook aurait besoin d'une méthode générale servant à regrouper les événements utilisés.

3.2.3 Approche de Weida [1993]

Weida [Weida, 1993] propose de s'attaquer au problème de la reconnaissance d'activités en utilisant un système de représentation de la connaissance exploitant la logique de description [Baader et al., 2007] afin de modéliser les états et les événements, tels que les activités avec leurs constituants. Les systèmes traditionnels de gestion de la connaissance en LD, comme KL-ONE [Brachman et Schmolze, 1985], sont largement utilisés pour résoudre des problèmes et raisonner sur la connaissance conceptuelle dans le cadre de la création d'applications « intelligentes ».

Cependant, ces systèmes sont limités par le fait que les logiques de description souffrent d'une incapacité à manipuler les compositions complexes des concepts [Haton, 1991]. Pour pallier à ce problème, Weida propose d'étendre le formalisme de la LD en introduisant la notion de réseaux de contraintes, où les nœuds du réseau représentent des concepts au sens de la logique des descriptions et où les arrêtes représentent des contraintes de nature temporelle. En ce sens, Weida s'est intéressé directement à la problématique de la reconnaissance temporelle d'activités. Pour celui-ci, une activité est d'abord décrite en termes d'un ensemble d'actions, défini chacune par un concept dans une base de connaissance en LD. Cet ensemble est ensuite organisé en un réseau de contraintes, dans lequel les actions font offices de nœuds. Les arcs introduits entre les actions induisent quant-à elles des contraintes temporelles entre les actions de l'activité.

3.2.3.1 Formalisation d'une activité sous forme d'un réseau de contraintes

Formellement, Weida propose de caractériser la définition d'une activité dans la librairie par un réseau de contraintes prenant la forme d'une structure de graphe $G = \langle V, E \rangle$, dans lequel V est l'ensemble des nœuds, chacun correspondant à un concept défini en logique de description et représentant formellement un événement (une action), et dans lequel E est l'ensemble des arcs représentant chacun une contrainte de nature temporelle entre deux événements inclus dans V . Ces contraintes correspondent en fait à des relations binaires simples d'Allen [Allen, 1991] ou à une disjonction de celles-ci. Par exemple, si l'on prend le cas d'une activité (un plan) P comprenant trois actions décrit par des concepts, de sorte que $V = \{FairePoulet, FaireMarinade, MélangerPouletMarinade\}$. On pourrait alors décrire un ensemble de contraintes temporelles en introduisant des arrêtes dans E , tel que $E = \{\text{Après } (FairePoulet, FaireMarinade), \text{ Avant } (FairePoulet, MélangerPouletMarinade), \text{ Avant } (FaireMarinade, MélangerPouletMarinade)\}$. Cet exemple est illustré visuellement à la figure 3.3. Dans le cas de cet exemple, les contraintes permettent de spécifier que le poulet doit être mis à cuire après avoir préparé la marinade, que la marinade doit avoir été mélangé avec le poulet avant de le cuire et que, évidemment, la marinade doit avoir été préparé avant de pouvoir la mélanger avec le poulet. La base de connaissances d'un agent de reconnaissance est alors formée, dans le modèle de Weida, d'un ensemble de structure de plans encodés selon ce format et d'une relation d'ordre partielle (subsumption de plans) permettant d'organiser ces plans par niveau de généralité de façon à donner naissance à une taxonomie.

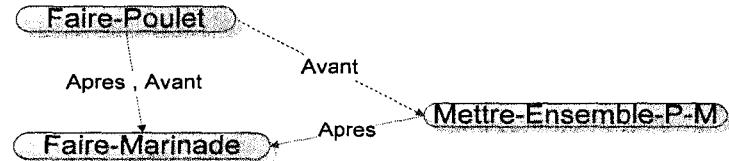


Figure 3.3. Un exemple de réseau de contraintes temporelles simples selon Weida [Weida, 1993]

3.2.3.2 Processus inférentiel et subsomption entre réseaux de contraintes

À partir de là, le processus d'inférence pour la reconnaissance d'activités proposé par Weida peut être caractérisé comme une correspondance entre les plans préétablies de la librairie de l'agent observateur et l'activité observée. Cette correspondance est basée sur la relation de subsomption entre le réseau de contrainte décrivant une activité et l'activité observée. Elle s'appuie sur la relation de subsomption de la logique de description¹ qui représente un fondement de la notion de classification. Formellement, un plan P de la librairie subsumera un plan observé O, si les nœuds (concepts) de G_p (le graphe représentant P), vérifient en totalité la relation de subsomption avec les nœuds correspondant dans G_o (le graphe représentant O) et que chaque arc (chaque contrainte temporelle) de G_o admet, au moins, une relation temporelle semblable (s'il s'agit d'une disjonction de relations) dans l'arc qui lui correspond dans G_p .

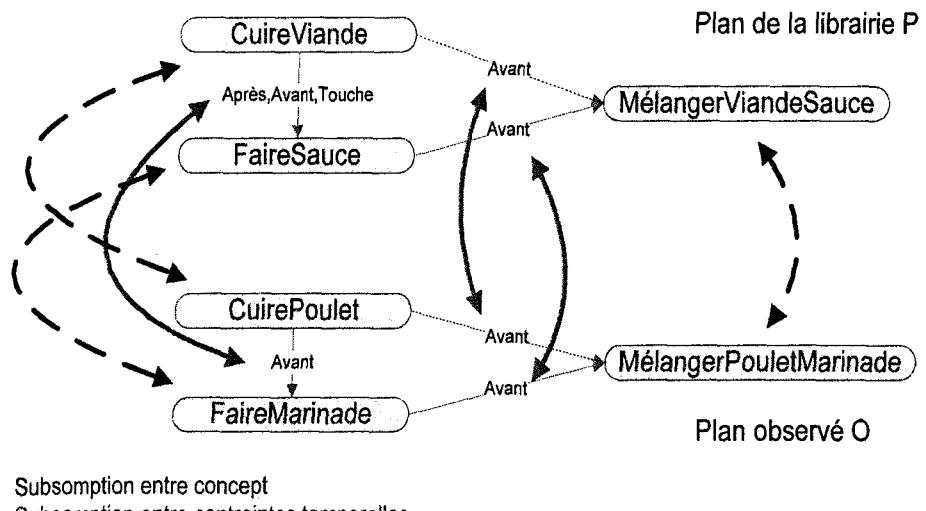


Figure 3.4. Exemple de subsomption entre un plan de la librairie et un plan observé

¹ Une synthèse des principaux éléments de la logique de description est disponible à l'annexe 1.

Par exemple, prenons P et O, respectivement, le plan de la librairie de l'agent observateur et le plan observé qui sont illustrés à la figure 3.4, où le plan P de la librairie est défini tel que :

$$P = (\{Avant, Après, Touche\} \{CuireViande, FaireSauce\}, \{Avant\} \{CuireViande, MélangerViandeSauce\}, \{Avant\} \{FaireSauce, MélangerViandeSauce\})$$

et le plan O et défini tel que :

$$O = (\{Après\} \{CuirePoulet, FaireMarinade\}, \{Avant\} \{CuirePoulet, MélangerPouletMarinade\}, \{Avant\} \{FaireMarinade, MélangerPouletMarinade\}).$$

D'après la notion de classification basée sur la relation de subsomption, chaque concept représentant une action de P doit admettre un subsumé dans O : *CuireViande*, *FaireSauce* et *MélangerViandeSauce* subsume donc respectivement *CuirePoulet*, *FaireMarinade* et *MélangerPouletMarinade* suivant la taxonomie d'actions. De même, chacune des relations des contraintes temporelles de O vérifie une seule relation de chaque contrainte temporelle disjonctive ou non-disjonctive (une contrainte équivaut à une seule relation) de P.

Par exemple, la contrainte *(Après)(CuirePoulet, FaireMarinade)* de O vérifie la contrainte *(Avant, Après, Touche)(CuireViande, FaireSauce)* de P, vu que la relation *Après* existe dans la disjonction *(Avant, Après, Touche)* de même pour le reste des contraintes. Ainsi on peut parler ici de relation de subsomption entre P et O, dans laquelle P est le subsumant et O le subsumé. La figure 3.4 résume l'exemple précédent en montrant la correspondance entre le plan P de la librairie et le plan observé O. Dans ce cas, une relation de subsomption serait détectée, indiquant à l'agent de reconnaissance que l'activité observée est potentiellement de type P.

3.2.3.3 Implémentation du système de Weida

Weida a implémenté son approche à l'intérieur d'un prototype de système de reconnaissance d'activité, appelé T-REX (Terminological Recognition System) [Weida, 1993]. Ce système a servi de plateforme d'essai pour valider l'approche. T-REX est en fait composé de deux sous systèmes, l'un pour gérer les concepts (nœuds du réseau) de la logique de description (K-Rep - Knowledge Representation) [Mays et al, 1991], et l'un pour gérer les contraintes de nature temporelle (MATS [Kautz et Ladkin, 1991]), comme le montre le diagramme de la figure 3.5. Une fois qu'une activité est définie par l'usager, T-REX vérifie sa cohérence syntaxique, normalise sa définition en dérivant l'information implicite et en la classifiant dans la librairie de plans à l'aide de la relation de subsomption. Cette librairie de plans se caractérise par une section renfermant l'ensemble de toutes les actions fournies par le système K-REP et une autre contenant

l'ensemble des contraintes temporelles (métriques et symboliques d'Allen) gérés par le système MATS et qui sont issues des plans définies et des observations. Dès qu'une observation se présente, T-REX reconnaît plusieurs ensembles de plans candidats correspondants aux modalités nécessaire, possible ou impossible, à partir du processus inférentiel vu à la section précédente. On parle alors ici d'une approche purement logique permettant de dégager de la librairie un ensemble d'activités possibles. Cependant, on ne dispose pas ici d'un moyen quantitatif de discriminer les différents plans possibles une fois ceux-ci identifiés.

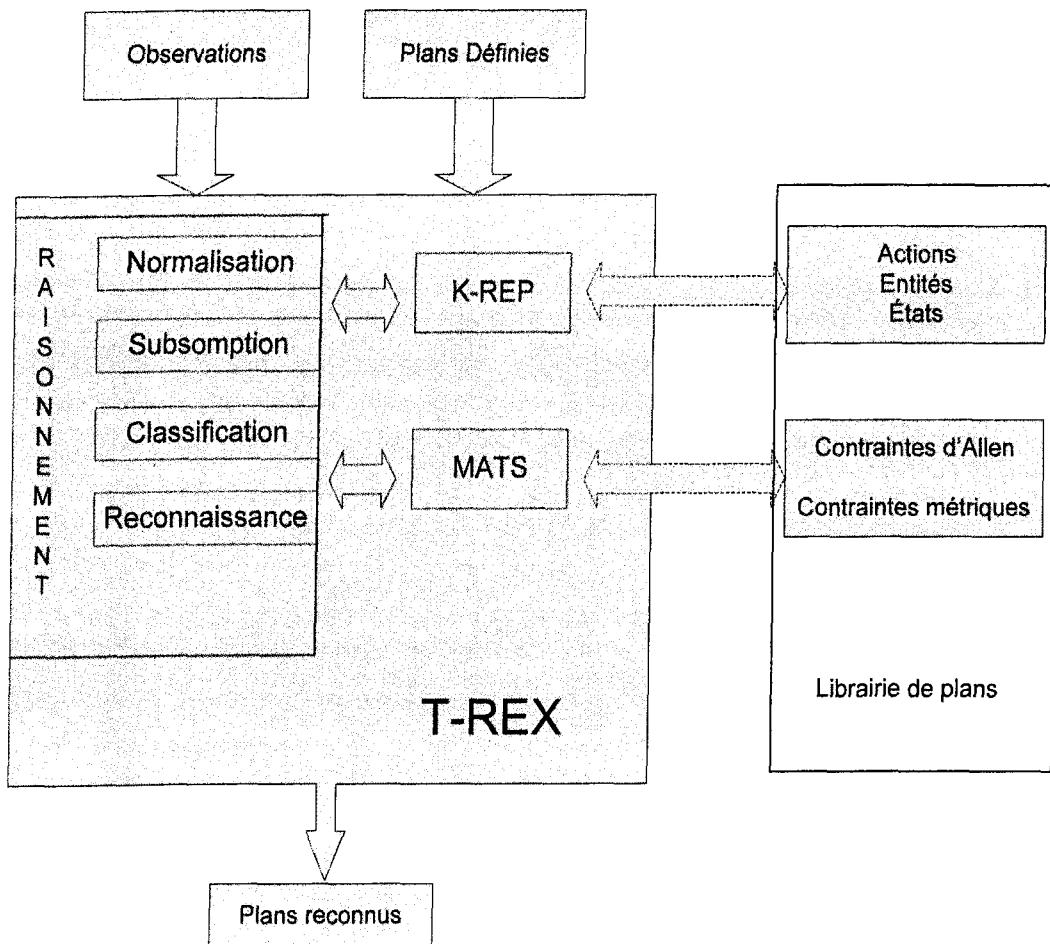


Figure 3.5. Le système T-REX [Weida, 1993]

3.2.3.4 Bilan sur le travail de Weida

Weida [Weida, 1993] propose dans son travail d'exploiter la logique de description pour mettre en œuvre un système de reconnaissance d'activités basée sur la notion de réseau de

contraintes temporelles. Ces réseaux sont formés de concepts au sens de la LD et de contraintes temporelles tirées du modèle d'Allen [Allen, 1991]. La relation de subsomption est ici étendue par Weida pour s'appliquer sur les réseaux de contraintes servant à décrire une activité. Cette façon de faire est ingénieuse car elle permet d'exploiter les algorithmes de classification de la logique de description pour mettre en œuvre le processus de reconnaissance d'activités. L'approche de Weida est aussi avantageuse car elle permet également de tenir compte, à l'intérieur du processus de reconnaissance, des contraintes temporelles entre les différentes actions d'une activité. La principale faiblesse de l'approche de Weida découle de l'utilisation du modèle d'Allen pour la représentation des contraintes temporelles entre les événements. Les relations d'Allen permettent uniquement de tenir compte de l'aspect qualitatif des contraintes temporelles liées à l'ordonnancement des actions à l'intérieur d'une activité, en négligeant l'aspect quantitatif lié aux durées des actions et aux délais entre ces actions, qui permet de cerner plus efficacement les erreurs cognitives. En ce sens, le modèle proposé par Weida s'avère très intéressant mais limité au niveau de la reconnaissance des éléments liés au temps.

3.3 Conclusion

Ce chapitre avait pour but d'introduire le lecteur à la problématique de la reconnaissance d'activités et de présenter un état de l'art sur les principales approches pour la reconnaissance d'activités qui ont été proposées dans la littérature et qui intègrent l'aspect temporel, à savoir, les modèles de Kautz [Kautz, 1986], Jakkula et Cook [Jakkula et Cook, 2006] et Weida [Weida, 1993]. La théorie de Kautz possède un fondement logique dans le processus inférentiel qui se base sur un algorithme de type CSP prenant en compte les contraintes temporelles. Dans le deuxième modèle étudié qui est celui de Jakkula et Cook on a considéré la reconnaissance de plans à l'intérieur d'un habitat intelligent, en tenant compte des erreurs temporelles qui peuvent avoir lieu dans une réalisation d'activités d'un habitant. Le troisième modèle, c'est celui qui va nous intéresser le plus, vu qu'il s'inscrit dans une approche hybride qu'on voudrait étendre, on a vu que Weida se base sur la logique de description afin de créer sa librairie de plans pour pouvoir appliquer la reconnaissance de plan grâce à la notion de classification basée sur la relation de subsomption. Ce chapitre a permis de mettre en lumières les forces et les faiblesses de chacune des approches, permettant ainsi de bien situer notre contribution dans le reste du mémoire. Nous avons pu constater, à travers l'état de l'art présenté aux chapitre 2 et 3, que l'ensemble des informations temporelles observables (durée d'une action, laps de temps entre deux actions, etc.) dans un habitat intelligent doté de capteurs se révèle d'une importance capitale dans le processus

de reconnaissance, notamment pour identifier certains types précis d'erreurs de réalisation. Nous avons aussi pu constater que ces éléments temporels sont souvent sous-exploités dans les travaux antérieurs, en raison de l'utilisation massive de la théorie temporelle d'Allen [Allen, 1983], qui ne prend en considération qu'à éliminer l'ordonnancement des actions dans le temps. Le prochain chapitre présente notre contribution : un modèle temporel de reconnaissance qui tient compte des erreurs provenant des délais et des durées entre les actions. Ce modèle se veut une extension de l'approche proposée par Weida [Weida, 1993], que nous avons décrit dans ce chapitre.

CHAPITRE 4

Modèle temporel pour la reconnaissance d'activités : extension de l'approche de Weida

4.1 Introduction

Nous avons pu constaté, au chapitre précédent, que l'approche de Weida [Weida, 1993] s'appuie sur la logique de description pour définir un processus inférentiel de reconnaissance d'activités tenant compte, d'une façon limitée, des contraintes temporelles. Cette méthode explorée constitue, à notre avis, l'approche la plus intéressante à exploiter dans notre contexte de recherche. La principale raison justifiant l'utilisation de l'approche de Weida comme fondement de notre contribution est qu'elle nous permet d'utiliser la relation de subsomption pour transformer le problème de reconnaissance d'activités en un problème de classification pouvant être résolu de façon efficace. En fait, la relation de subsomption permet ainsi d'exploiter le paradigme de classification pour effectuer une correspondance efficace entre les événements observés et les actions/activités de la taxonomie [Bouchard et al., 2007]. Une autre raison justifiant l'utilisation de cette approche est sa proximité algébrique avec les modèles antérieures que nous avons élaborés et implantés [Bouchard et al., 2007][Roy et al., 2009], permettant ainsi de fournir une contribution compatible avec les modèles théoriques et les outils logiciels déjà en place au laboratoire.

Au niveau temporel, en explorant le modèle de reconnaissance d'activités de Weida [Weida, 1993], nous avons remarqué que la prise en compte du temps est caractérisée par des contraintes temporelles basées sur le modèle d'Allen et correspondant à des disjonctions temporelles de nature symbolique et qui peuvent décrire un ordre temporel d'une action par rapport à une autre [Allen, 1983]. Tel que nous l'avons mentionné dans l'introduction, dans le domaine de l'assistance cognitive, la quantification et la précision de l'information temporelles sont essentielles afin de mieux superviser et assister un patient possédant un déficit cognitif. Ceci nous permet de conclure que les relations temporelles symboliques induites dans l'approche de Weida ne permettent pas de reconnaître toutes les situations erronées qui nous intéressent, en l'absence de la prise en compte des durées et des délais réelles des événements. Ces informations sont pourtant vitales dans notre cas, où l'on fait face à un comportement erratique de la part d'un patient souffrant d'un déficit cognitif. Afin d'illustrer clairement cette problématique dans le contexte particulier du modèle de Weida [Weida, 1993], nous allons réutiliser le scénario simple décrivant une recette de cuisine de poulet mariné, spécifié sous la forme d'un graphe temporel de Weida à la figure 4.1.

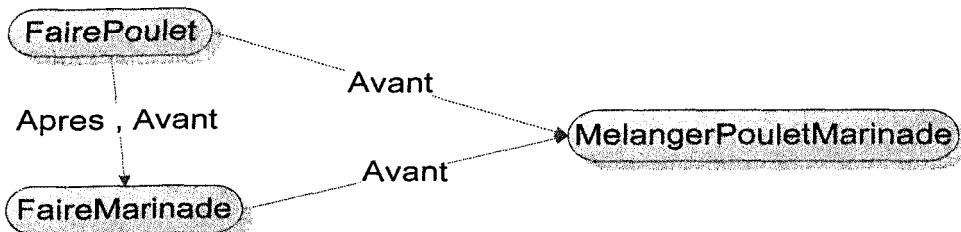


Figure 4.1. Graphe temporel, scénario du poulet mariné [Weida, 1993]

Dans ce scénario, la seule information pouvant être encodée dans le modèle de l'activité concerne l'ordre d'une action par rapport à une autre dans le temps. L'action *FairePoulet*, par exemple, peut se réaliser avant ou après l'action *FaireMarinade*, tout en sachant que ces deux actions ne peuvent se réaliser qu'avant l'action *MélangerPouletMarinade*. On remarque, à partir de ce scénario, qu'il y a une absence totale de la connaissance quantitative de l'aspect temporel, comme, par exemple, la durée minimale de temps nécessaire pour assurer la cuisson du poulet ou bien pour préparer la marinade. Ainsi, dans le cas de cette recette, l'action *FairePoulet* devrait avoir une durée minimale d'environ 20 minutes. De façon plus précise, cette action devrait durer au moins 20 minutes, mais ne devrait pas non plus dépasser 60 minutes. Supposons qu'un patient Alzheimer réalise cette action et que celle-ci se prolonge pendant 100 minutes. Il

s'agirait alors clairement d'une erreur pouvant générer un incendie et qui est, en fin de compte, un oubli lié à sa déficience cognitive menant à ce qu'on peut qualifier de *débordement de la durée normal du temps* requis à l'exécution de cette action. Par conséquent, le graphe temporel proposé par Weida ne permet pas d'encoder l'information quantitative permettant de reconnaître une telle erreur. Il en va de même pour les informations liées aux délais entre les actions. Par exemple, supposons que la marinade doit absolument être préparée avant le poulet et que celle-ci doit macérer au moins 10 minutes avant d'être mélangée à celui-ci. Il est encore une fois impossible de représenter correctement cette situation avec le modèle original de Weida.

Le constat est donc évident : les relations temporelles du graphe comportent une faiblesse considérable au niveau de la précision quantitative qui est pourtant nécessaire, voir même vitale, afin d'identifier les anomalies de réalisations reliées au temps. De là, on peut dire que l'approche de Weida est insuffisante, du point de vue de la perspective temporelle adoptée dans le domaine de l'assistance cognitive, pour reconnaître adéquatement les activités d'un patient Alzheimer. Également, on note que l'approche de Weida suppose qu'on peut directement observer les actions de bas niveau servant de nœuds dans le graphe décrivant une activité. Cependant, dans le cas l'habitat intelligent, on peut observer seulement des événements de bas niveau (ex. déclenchement d'un détecteur de mouvement et d'un tapis tactile) constituant des indices sur l'action en cours (ex. déplacement de la cuisine vers le salon). L'approche souffre donc également d'un manque au niveau de la représentation des actions de bas niveau permettant de faire l'appariement entre les événements observés et l'action véritablement en cours.

Afin de palier à ces limitations, nous proposons dans ce chapitre une version améliorée du modèle de reconnaissance d'activités proposé par Weida [Weida, 1993]. Celui-ci permet de modéliser les actions de bas niveau de façon à pouvoir faire l'appariement entre celles-ci et les événements observés, et de prendre en considération les durées et les délais au niveau quantitatif et de placer la représentation du temps dans un référentiel numérique. L'amélioration de l'aspect temporel est basée sur l'approche numérique de Dechter [Dechter et al, 1991] qui utilise le formalisme de graphe de contraintes, où les nœuds représentent des événements ponctuels et les arcs représentent des intervalles numériques, appelés aussi fenêtres temporelles, délimitant quantitativement le moment de départ et le moment de fin d'une action. Cette approche numérique est utile lors du processus de reconnaissance, où l'observation en provenance des capteurs nous fournit directement les données temporelles liées au déclenchement des capteurs, nous permettant ainsi d'inférer les durées et les moments de début et de fin des activités observées. Dans notre modèle, ces données temporelles tirées des observations pourront être formellement comparées aux intervalles correspondant aux activités prédéfinies de notre librairie

(base de connaissances). Par exemple, la description d'une action/étape *FairePoulet* à l'intérieur d'une activité *PréparerPouletMariné* pourra se voir attribuée une durée minimale de 20 minutes et une durée maximale de 60 minutes, représenté par l'intervalle [20, 60]. Lors de l'observation de la réalisation de cette activité, il faudra alors que la durée observée par l'agent assistant soit incluse dans cet intervalle, sinon il y a une anomalie temporelle. De la même manière, le délai minimum et le délai maximum entre deux actions impliquées dans la réalisation d'une activité sera également borné par un intervalle. Il s'agira alors, sommairement, de vérifier la relation de subsumption (inclusion ensembliste) entre les intervalles de temps modélisés dans la nouvelle du graphe temporel. Si les contraintes numériques ne sont pas satisfaites, on générera alors une erreur, soit un débordement de la durée normal du temps, soit un délai incohérent entre deux actions.

Les prochaines sections du chapitre serviront à décrire formellement cette nouvelle extension du modèle de Weida. Elles décriront également l'implémentation réalisée de ce nouveau modèle ainsi que les efforts déployés pour en faire la validation. La suite du chapitre se divise en trois parties. La première présente l'apport théorique proposé au modèle de Weida. La seconde présente notre contribution pratique, prenant la forme de l'implémentation de ce nouveau modèle de reconnaissance à travers une application réelle. La dernière partie présente une étude comparative des performances et apports de ce nouveau modèle amélioré par rapport au modèle précédent de Weida. Nous conclurons le chapitre sur un bilan de nos travaux.

4.2 Extension théorique du modèle de Weida

La première extension que nous proposons au modèle de Weida concerne les concepts utilisés pour décrire les actions de bas niveau. En fait, Weida représente les actions en utilisant carrément des concepts au sens de la logique de description [Baader et al., 2007]. Ces concepts sont en fait primitifs et reviennent à un simple ensemble de termes organisés par niveau de généralité dans une taxonomie. Cette vision de l'action s'avère suffisante si l'on suppose, comme Weida, que les « capteurs » du système sont capables d'observer directement ces événements / actions. Cependant, dans le contexte de l'habitat intelligent, la notion de concept en logique de description se revêt statique et insuffisante pour décrire la dynamique d'un comportement (i.e. les changements résultant de l'application de cette action dans l'environnement) [Bouchard et al., 2007]. C'est pourquoi les nœuds de notre graphe temporel ne peuvent être représentés par de simple concept en logique de description. On doit donc proposer l'utilisation d'une structure de concept « augmenté » permettant de représenter adéquatement les actions d'une activité.

Dans notre extension du modèle de Weida, les nœuds du graphe correspondront dorénavant à des concepts dynamiques d'actions [Bouzouane, 2005] formalisées en logique de description en définissant une paire $\langle W, A \rangle$ qui représente un modèle état-transition. À l'intérieur de cette structure algébrique, W représente l'ensemble des états possibles du monde et A représentant l'ensemble des actions qui s'applique sur ces états. Une action $a \in A$ sur l'ensemble W est une relation binaire $a \subseteq W \times W$, tel que $\langle w, e \rangle \in a$ si et seulement si $a(w) = \{e | (w, e) \in W \times W\}$, où w et e représentent, respectivement, l'état suivant. Ainsi, les actions sont déterministes et opèrent sur les formules d'assertions, qui sont des cas particuliers de formules de la logique du premier ordre [Borgida, 1996]. Si les expressions des concepts et les assertions en logique de descriptions des faits à propos de l'état du monde, qui signifie l'environnement du patient, elles peuvent alors être satisfiables ou non, en fonction de l'état courant. Par conséquent, les états du monde peuvent correspondre à des structures sémantiques. Soit $w = \langle \text{Dom}(w), (.)^{lw} \rangle$ une structure sémantique tel que $\text{Dom}(w)$ représente le domaine d'interprétation, c'est-à-dire l'ensemble non vide d'objets appelés individus existant dans le monde à l'état w à un moment spécifique. La fonction $(.)^{lw}$ exprime une fonction d'interprétation associée à w , assignant à chaque symbole de concept C un sous ensemble de domaine $\text{Dom}(w) \times \text{Dom}(w)$ de manière à ce que les formules qui définissent la sémantique de la logique de description soient satisfaites [Baader et Nutt, 2003].

Soit les symboles C et D , qui désignent des concepts et r le nom d'un rôle au sens de la logique de description. La relation de subsumption existante entre les concepts est dénoté comme suit : $D < C$ ou C subsume D , ce qui signifie que $D^{lw} \subseteq C^{lw}$ dans l'état W . Une assertion de la forme $C(i)$ spécifie que l'individu i est une instance du concept C et une assertion $r(i, j)$ indique que le couple d'individus (i, j) se trouve dans l'extension de r . Les actions interviennent au niveau assertionnel qui correspond à l'extension des concepts et rôles. Dans le but d'associer une interprétation aux assertions, la fonction $(.)^{lw}$ est étendue aux individus. Par exemple, $C(i)$ est satisfiable par w , on peut noter alors $w \models C(i)$, si et seulement si $i \in C^{lw}$.

4.2.1 Sémantique de l'action en logique de description

Une action $a(w)$ est une structure utilisant une formulation tiré de celle proposée dans le travail de Fikes et Nilsson [Fikes et Nilsson, 1971] concernant le langage STRIPS qui utilise les notions de préconditions et de postconditions.

- La précondition $pre(a)$ de l'action a consiste en une conjonction de formules d'assertions qui concernent les objets de concepts ainsi que les rôles qui s'associe avec ces objets. L'ensemble des états dans lequel l'action $a(w)$ peut être effectuée correspondant au

domaine : $\text{Dom}(a) = \{w \in W \mid w \models \text{pre}(a)\}$. Par conséquent, toutes les assertions qui composent $\text{pre}(a)$ doivent satisfaire chacun des états w tel que $a(w) \neq \emptyset$.

- La postcondition $\text{pos}(a)$ représente les effets produits par l'exécution de l'action $a(w)$ qui peut être exprimé par les formules d'assertion induites dans $\text{post}(a)$ représentant les ajouts à l'interprétation des concepts et des rôles qui concerne l'action $a(w)$, ainsi, pour les assertions qui spécifient les suppressions à l'interprétation des concepts et des rôles, qui est dénoté par $\text{pos}^{\sim}(a)$. L'ensemble des états résultant de l'exécution de l'action a correspond au co-domaine $\text{CoDom}(a) = \{e \in W \mid e \models \text{pos}(a)\}$.

4.2.2 La subsomption entre les actions

Après avoir étudié la sémantique de la structure d'action en logique de description, on définit maintenant la relation de subsomption qui va permettre d'organiser les nouveaux concepts d'actions en une taxonomie. Soit a et b , deux structures d'actions construites suivant le modèle que nous venons de présenter dans la section précédente, il existe une relation de subsomption entre ces deux actions, où b est le subsumant et a le subsumé, si le domaine et le co-domaine de l'action a sont respectivement inclus dans le domaine et le co-domaine de l'action b . D'une façon formelle, une action b subsume une action a , qui se dénote par $a <_a b$ si et seulement si $\text{Dom}(a) \subseteq \text{Dom}(b)$ et $\text{CoDom}(a) \subseteq \text{CoDom}(b)$. Le symbole $<_a$ indique la relation de subsomption entre les concepts d'actions. La sémantique de cette relation de subsomption d'actions est une extension directe de la relation de subsomption traditionnelle qui provient de la logique de description. Si $a <_a b$ alors pour tout état $w \subseteq W$: $w \models \text{pre}(a) \rightarrow w \models \text{pre}(b)$. De même pour les préconditions : $\text{Dom}(a) \subseteq \text{Dom}(b)$ et $\text{CoDom}(a) \subseteq \text{CoDom}(b)$. Cette extension de la relation de subsomption peut être vue comme un élément de base sur lequel s'appuie le processus de reconnaissance d'activités qu'on veut étendre et que nous définirons plus loin dans ce chapitre.

4.2.3 Extension numérique des contraintes temporelles de Weida

Du point de vue temporel, le modèle original de Weida utilise les treize contraintes issues du modèle d'Allen. Ces contraintes sont représentées à l'aide d'arrêtes dans le graphe servant à décrire une activité dans la librairie. Comme nous l'avons vu, une activité est définie comme étant structure de graphe $G = \langle V, E \rangle$ dans lequel les nœuds V correspondent à des concepts d'action dans le sens de la logique de description, et dans lequel les arrêtes E représentent des contraintes temporelles tirées du modèle d'Allen permettant d'ordonnancer ces actions dans le

temps. Dans le cas de notre extension, les nœuds V du graphe correspondent, comme nous venons de le voir, non pas à un concept primitif en logique de description mais bien à une structure d'action dynamique telle que définie dans les sections précédentes. Grâce à la relation de susbsomtion définie entre ces concepts d'actions, ceux-ci forment une taxonomie où chaque type d'action peut être vu comme un plan atomique. Cette taxonomie inclus tous les types d'actions appartenant au plan observé durant le processus de reconnaissance. Elles permettent, dans notre contexte applicatif, d'effectuer un appariement efficace entre les observations et les actions de la librairie. Lors du déclenchement d'une série d'événements, le type d'action en présence peut alors être identifié en faisant la correspondance entre l'état w précédent ces événements et le domaine des concepts d'actions de la librairie, ainsi qu'entre l'état e résultant et le co-domaine de celles-ci. D'où l'avantage de ce premier apport au modèle de Weida.

Toutefois, du coté des relations temporelles utilisées dans le modèle original, leur forme symbolique souffre, comme on l'a vu, d'une faiblesse au niveau de l'expressivité quantitative du temps. En d'autres termes, elles ne permettent pas de donner des précisions métriques concernant le temps comme le moment de début et de fin d'une action, la durée d'une action ou bien le temps écoulé entre deux actions. Par conséquent, après avoir modifié les nœuds V , nous proposons une modification des contraintes temporelles symbolique incluses dans E , pour les transformer en contraintes temporelles numériques. Il s'agit d'une alternative bien intéressante puisque ces nouvelles contraintes vont nous permettre de mieux quantifier le temps d'exécution des actions et ainsi être en mesure de cerner les anomalies issues d'un comportement erratique d'un patient lors du processus de reconnaissance.

Concrètement, notre proposition consiste à introduire une nouvelle représentation des contraintes temporelles du modèle de Weida, inspirées du travail de Dechter [Dechter et al, 1991]. Celui-ci propose un raisonnement temporel numérique et qui va permettre de reformuler la structure des contraintes dans la description d'une activité (un graphe temporel). Chaque action va ainsi se voir attribuer une contrainte de durée, représentée par un intervalle numérique. De la même façon, chaque relation entre deux actions du graphe va se voir attribuer une contrainte de délais. Ainsi, lors d'une inférence, et après avoir appliquée la notion de classification sur les concepts d'actions de l'activité observée, on vérifiera alors la correspondance entre les durées réelles observées de chaque action et les intervalles numériques attribués à ces actions dans le graphe temporel servant à encoder le modèle de l'activité dans la librairie. De cette manière, dans le cas où les durées réelles ou les délais ne vérifient pas les intervalles correspondants, une erreur temporelle pourra être détectée.

Formellement, l'approche suit la perspective de Dechter [Dechter, 1991] où les actions sont considérées comme étant des propositions. À chaque proposition P_i est associée à un intervalle $I_i = [a_i, b_i]$, exprimant la durée de l'action. Les points du temps sont considérés comme des variables sur laquelle on peut établir des contraintes. Ces points peuvent représenter le point (moment) de début ou le point (moment) de fin d'une action quelconque. De la même manière, on définit les délais se trouvant entre chaque couple de propositions (actions) pour caractériser les relations temporelles d'une manière numérique et de pallier au problème que pose les relations symbolique d'Allen dans la représentation temporelle des plans de Weida. En d'autre mots, on introduit deux types d'intervalles temporels, le premier est propre à l'action pour exprimer une durée, le deuxième est celui qui entretient l'action avec une autre pour exprimer un délai.

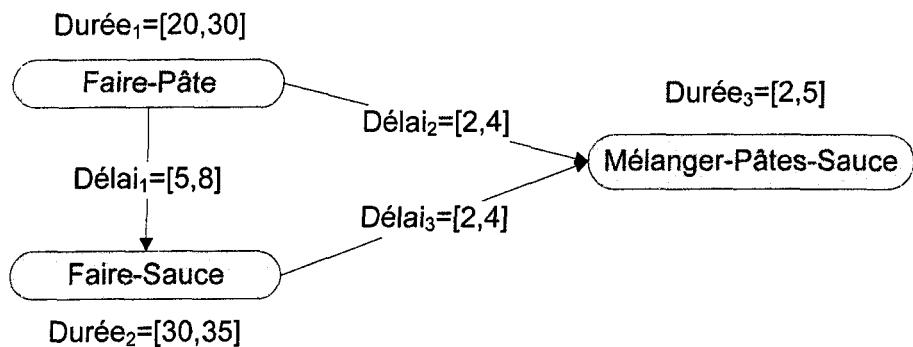


Figure 4.2. Exemple de scénario impliquant les nouvelles contraintes temporelles

Afin d'illustrer cette approche, considérons le scénario que l'on peut voir à la Figure 4.2. Le graphe temporel représenté ici est issu de notre nouveau modèle étendu. Il permet de caractériser précisément le déroulement temporel inhérent à la réalisation d'un plat de pâtes. Il spécifie qu'on doit d'abord faire bouillir la pâte pendant une durée de 20 à 30 minutes. Par la suite, on doit préparer la sauce, ce qui prendra entre 30 à 35 minutes. Cette sauce *Alfredo* doit être prête de 5 à 8 minutes après la réalisation du *Spaghetti*, afin que les pâtes aient juste assez de temps pour se départir de l'eau de cuisson, mais pas trop pour devenir sec. Enfin, après avoir laissé reposer les pâtes pendant au moins 2 à 4 minutes, on mélange l'ensemble pour avoir des *Spaghetti* avec sauce *Alfredo*. L'action *MélangerSpaghettiSauce* devrait durer de 2 à 5 minutes pour assurer un bon mélange.

Pour représenter une telle situation, on utilise une structure qui se fond sur un formalisme de réseau de contraintes temporelles, à savoir, les problèmes temporels simples (STP), où les contraintes sont des intervalles simples et non disjonctifs. Notre STP dans cette proposition peut être défini comme un système impliquant un ensemble d'intervalles temporels, $X = \{x_1, \dots, x_n\}$

qui sont attribués à des événements ou des actions tel que $x_i = [x_{i-\text{début}}, x_{i-\text{fin}}]$ et un ensemble de contraintes binaires sur les points du temps $C = \{c_1, \dots, c_m\}$ qui correspondent à chaque couple d'élément de l'ensemble X , c'est-à-dire, une contrainte temporelle entre deux actions. Chaque arc du réseau de contraintes $i \rightarrow j$ admet un intervalle $[a_{ij}, b_{ij}]$, tel que a_{ij} et b_{ij} représentent les délais minimaux et maximaux qui existe entre deux éléments de X . Dans le cas de notre exemple, on peut définir A_1 comme étant la proposition caractérisant l'action « *FaireSpaghetti* », A_2 la proposition caractérisant l'action « *FaireSauceAlfredo* », et A_3 la proposition caractérisant l'action « *MelangerSpaghettiSauce* ». On peut définir les intervalles correspondant aux durées : $[x_1, x_2] [x_3, x_4] [x_5, x_6]$, tel que x_1 et x_2 représentent respectivement le temps minimal et maximal correspondant à l'action *FaireSpaghetti*, x_3 et x_4 ceux de l'action *FaireSauceAlfredo* et x_5 et x_6 ceux de l'action *MelangerSpaghettiSauce*. On peut aussi définir les intervalles correspondant aux délais : $[y_1, y_2] [y_3, y_4] [y_5, y_6]$ tel que y_1 et y_2 représentent respectivement le temps minimal et maximal qui borne le délai entre les actions *FaireSpaghetti* et *FaireSauceAlfredo*. De même, y_3 et y_4 borne le délai existant entre *FaireSpaghetti* et *MelangerSpaghettiSauce*, et y_5 et y_6 borne le délai existant entre *FaireSauceAlfredo* et *MelangerSpaghettiSauce*. L'ensemble de ces contraintes temporelles sont prédefinies dans les modèles d'activités de la librairie. Ces modèles d'activités sont organisés en taxonomie, tout comme les actions, grâce à la relation de subsomption.

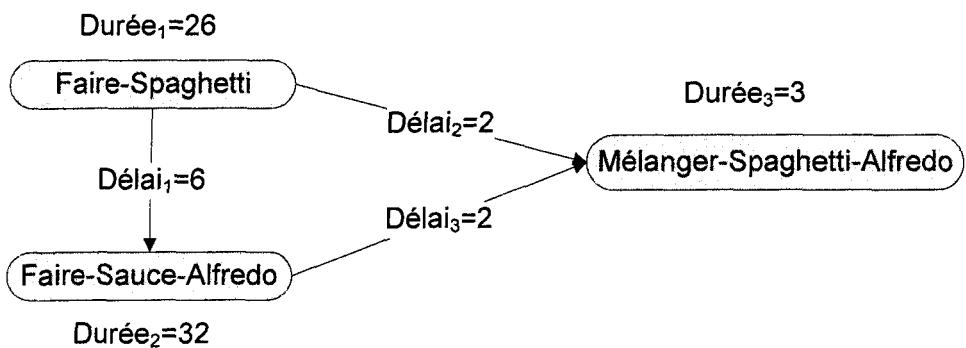


Figure 4.3 Exemple de l'observation de la réalisation de l'activité *PréparerSpaghettiAlfredo*

Lorsque l'observation d'une séquence d'actions est détectée, celle-ci est aussi organisée sous la forme d'un réseau de contraintes et est comparée par classification aux activités de la librairie. Par exemple, on peut voir à la figure 4.3 un réseau représentant l'observation d'une réalisation de l'activité *PréparerSpaghettiAlfredo* en termes d'un réseau de contraintes temporelles. On remarque que dans ce réseau, la durée attribuée aux actions (nœuds) ainsi qu'aux délais entre les actions est représentée par un temps et non pas par un intervalle de temps. Par exemple, on peut

voir que le spaghetti a pris 26 minutes à cuire exactement et que la sauce Alfredo a pris 32 minutes à préparer. Pour des fins de comparaison, au sens de la subsomption, avec les modèles d'activités de la librairie, cela revient à dire que la borne inférieure et supérieure de l'intervalle de temps observé est identique, par exemple, pour l'action *FaireSpaghetti* qui a pris 26 minutes, l'intervalle est [26, 26].

4.2.3.1 Définition du processus de reconnaissance d'activités

Après avoir défini la structure des modèles d'activités de la librairie dans un cadre temporel, on est maintenant en mesure de définir le nouveau processus de reconnaissance. Ce processus est une extension de celui proposé par Weida qui utilise la classification en exploitant la relation de subsomption de la logique de description. Le principe est d'encoder la séquence d'actions en cours d'observation sous la forme d'un réseau de contraintes compatibles avec les modèles d'activités de la librairie, comme nous venons de le voir, et de tenter de classifier ce réseau dans la taxonomie formant la librairie d'activités de l'agent observateur. Cette classification nous permettra d'identifier les activités potentiellement en cours de réalisation, c'est-à-dire celles qui subsument le réseau de contraintes correspondants à la séquence observée. Pour ce faire, nous devons définir formellement la relation de subsomption entre deux réseaux de contraintes.

Définition : Subsomption entre deux (activités) réseaux de contraintes temporelles

Un réseau de contraintes temporelles P , correspondant à un modèle d'activités dans la librairie, subsume un autre réseau de contraintes O , si et seulement si : (i) pour chacune des actions (nœuds) $o_i \in O$, il existe une action correspondante $p_i \in P$ pour laquelle $o_i <_a p_i$, et (ii) si l'intervalle $[xo_1, xo_2]$, spécifiant la durée de O , est subsumée par l'intervalle $[xp_1, xp_2]$, spécifiant la durée de P , de sorte que $[xo_1, xo_2] \subseteq [xp_1, xp_2]$, et (iii) si pour chaque arc $<(o_i, o_j), [yo_1, yo_2]> \in O$, où o_i et o_j représentent respectivement le domaine et le co-domaine de la relation temporelle et où $[yo_1, yo_2]$ caractérise le délai entre ces deux actions, il existe un arc correspondant $<(p_i, p_j), [yp_1, yp_2]> \in P$, pour laquelle $o_i <_a p_i$, $o_j <_a p_j$ et $[yo_1, yo_2] \subseteq [yp_1, yp_2]$.

Afin de mieux comprendre cette définition, on verra à la Figure 4.4 un exemple concret de vérification de la relation de subsomption qui reprend les deux exemples précédents. Le réseau en haut de la figure représente le modèle de l'activité *PréparerPâtesSauce* alors que la partie du bas de la même figure représente l'observation d'une séquence d'actions correspondant à la

préparation de spaghetti Alfredo, qui se veulent un sous type de plat de pâtes à la sauce. On peut voir la correspondance effectuée entre chacun des nœuds des réseaux respectifs, ainsi qu'entre les arcs. On peut également constater la vérification de l'inclusion des contraintes temporelles.

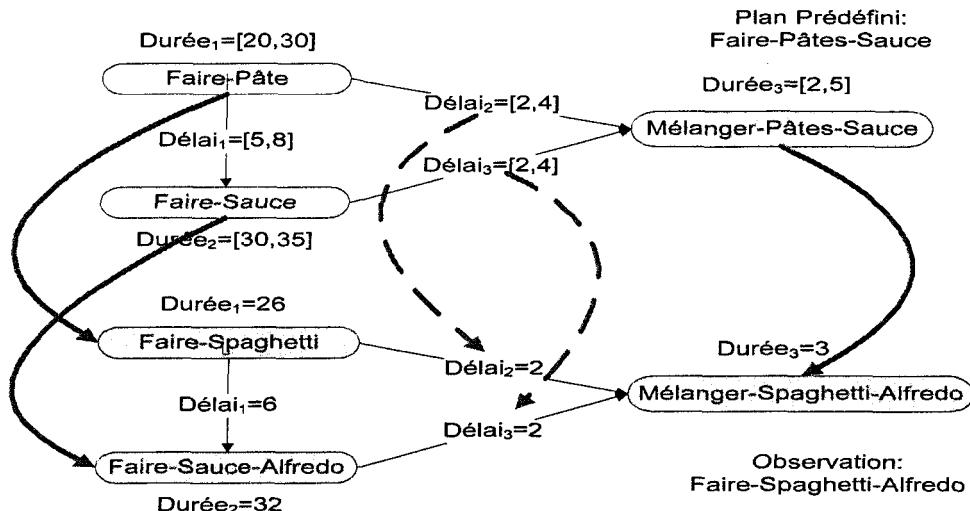


Figure 4.4. Relation de subsumption entre un modèle d'activité et un plan observé

Dans cette figure, on remarquera particulièrement trois types de flèches :

- Les fines flèches droites (noirs) : elles représentent les arcs définis dans les différents réseaux de contraintes temporelles.
- Les flèches courbées pleines (rouges) : elles caractérisent les relations de subsumption existantes entre les actions (noeuds) des différents réseaux et, aussi, entre les intervalles représentant les durées attribuées à chacune de ces actions. Le nœud d'origine de la flèche est le subsumant et celui de destination est le subsumé.
- Les flèches pointillées (bleus) : elles représentent la relation de subsumption entre les différents arcs des réseaux de contraintes.

La vérification de la relation de subsumption entre deux activités est la pierre angulaire de notre processus inférentiel. Les plans de la librairie de l'agent observateur sont organisés par niveau de généralité, ce qui permet, au fur et à mesure de la vérification des relations de subsumption existantes, d'identifier avec précision l'activité la plus susceptible d'être en cours de réalisation. La figure 4.5 nous montre, par exemple, le positionnement de la séquence d'actions

observées à l'exemple précédent au sein d'une taxonomie de modèle d'activités constituant la librairie d'un agent observateur.

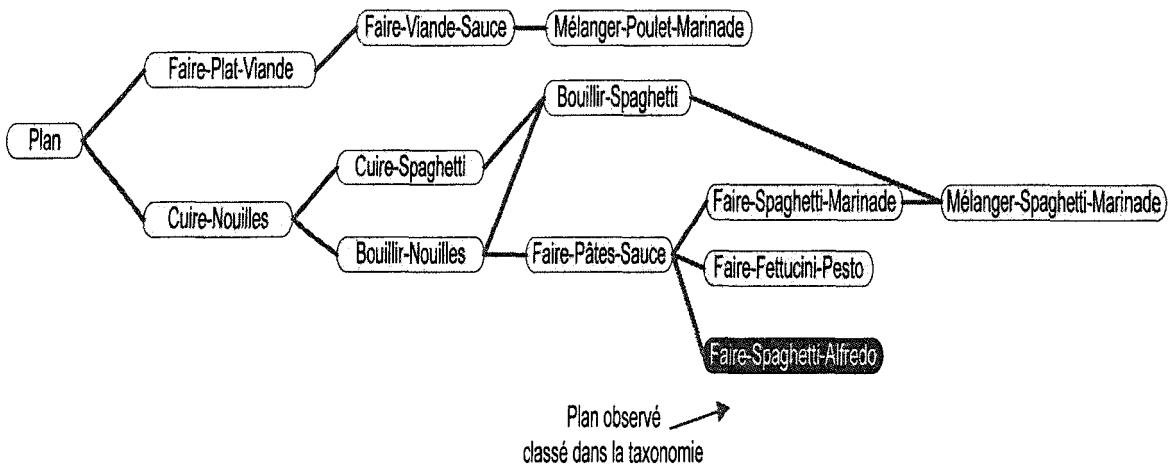


Figure 4.5. Classification d'une activité observée dans la taxonomie de modèle d'activités

En fait, à chaque observation, un modèle partiel de l'activité en cours est construit et classifié dans la hiérarchie. Plus il y a d'observation, plus ce modèle partiel se complète, ce qui permet de la positionner plus précisément dans la hiérarchie et ainsi éliminer des branches de l'arbre. Il est à noter que l'approche de Weida suppose qu'une seule activité est réalisée à la fois et que notre extension s'en tient à la même limitation. Cependant, les travaux du laboratoire sur les modèles de reconnaissance logiques [Bouchard et al., 2007], probabilistes [Roy et al., 2009] et possibiliste [Roy et al., 2010] s'attaquent quant à eux à ce défi particulier de combiner la reconnaissance de plusieurs activités à la fois. Également, notre approche, tout comme celle de Weida, suppose que la librairie est complète, c'est-à-dire que toutes les actions et activités que l'on peut potentiellement observées existent nécessairement dans la librairie.

4.2.3.2 Exemple de détection d'une erreur de nature temporelle

Afin d'illustrer comment ce processus inférentiel permet de mieux reconnaître les erreurs de nature temporelle, nous allons reprendre l'exemple précédent mais cette fois, en introduisant des erreurs liées au temps dans la séquence d'actions observées. Cet exemple, décrit à la Figure 4.6, reprend dans la partie du haut le même modèle d'activité décrivant la réalisation d'un plat de pâtes à la sauce. La partie du bas reprend quant à elle la séquence d'actions observées d'un plat de pâtes à la sauce Alfredo. Cependant, la durée de cuisson observée des spaghettis est de

seulement 4 minutes dans ce nouvel exemple. Par ailleurs, le délai observé entre la réalisation de la sauce Alfredo et le mélange de celle-ci avec les spaghetti de 100 minutes.

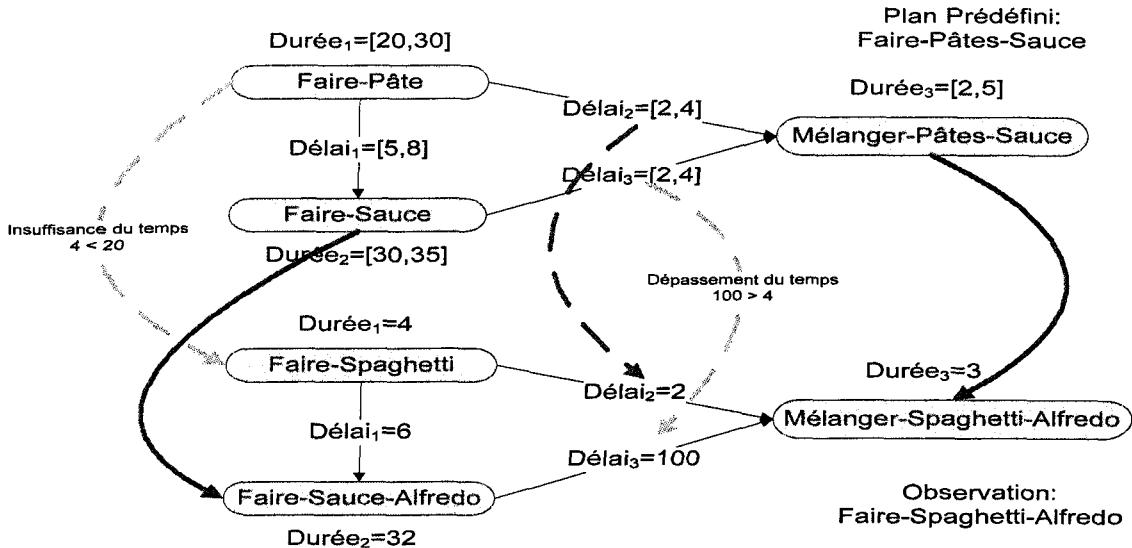


Figure 4.6. Identification d'erreurs de types dépassement et d'insuffisance du temps

On peut noter sur la Figure 4.6 que la vérification des relations de subsomption permet d'identifier les deux erreurs de nature temporelle. Dans le cas de la durée de cuisson des pâtes, la comparaison de la durée observée avec l'intervalle définie pour l'action *FairePâte* permet de détecter une erreur de type « Insuffisance du temps de réalisation ». Effectivement, il est impossible que les spaghetti aient eu le temps de cuire en seulement 4 minutes. De la même façon, dans le cas du délai observé entre la réalisation de la sauce Alfredo et le mélange de celle-ci avec les spaghetti, on peut ici détecter un « dépassement incohérent du délai normal » entre ces deux actions. Ces deux détections d'erreurs sont représentées dans la figure ci-haut par des flèches pointillées en jaune avec une étiquette marquant l'erreur identifiée.

4.2.4 Bilan de notre apport théorique

Cette première section du chapitre avait comme objectif de présenter la contribution théorique du mémoire, qui correspond à une extension simple du modèle de reconnaissance temporel de Weida [Weida, 1993], basé sur une formalisation de l'action en logique de description et sur une approche de raisonnement numérique sur les contraintes temporelles inspirée du travail de Dechter [Dechter et al, 1991]. L'extension de l'approche de Weida au

niveau de la représentation des actions permet de tenir compte de la réalité de l'habitat intelligent, dans lequel on ne peut identifier directement l'action en cours. L'extension temporelle proposée au modèle permet quant à elle d'orienter le processus de reconnaissance d'activités afin d'être apte à détecter les anomalies liées au temps qui, tel que discuté en introduction, sont d'une importance capitale dans notre contexte d'assistance cognitive. En ce sens, nous pouvons dire que notre modèle hérite la base structurelle du modèle de Weida, en lui rajoutant une précision numérique au niveau du cadre temporelle, ce qui implique l'augmentation de l'efficacité du processus de reconnaissance et par se fait, l'agent assistant sera en mesure d'exploiter ses différentes opportunités afin de mieux assister un patient. La prochaine section présente la seconde partie de notre contribution, soit l'implémentation et la validation de ce nouveau modèle étendu.

4.3 Implémentation et validation de notre nouveau modèle étendu

Afin de vérifier la viabilité et l'opérationnalité de notre modèle, nous avons réalisé une implémentation de celui-ci au sein d'un contexte de simulation. Nous avons également implémenté, pour fins de comparaisons, deux versions du modèle de reconnaissance proposé par Weida [Weida, 1993], l'une intégrant les contraintes temporelles et l'autre faisant complètement abstraction des notions liées au temps. L'objectif étant de démontrer d'une part, le caractère essentiel de l'aspect temporel et, d'autres part, de montrer que notre prise en compte du temps permettait d'obtenir de meilleurs résultats que l'approche originale de Weida. Par la suite, nous avons procédé à une série d'expérimentation avec les trois modèles en simulant un ensemble de scénarios tirées de cas réels servant comme données d'entrée pour les observations au sein d'un habitat intelligent virtuel. Pour ce faire, nous avons développé un simulateur 3D d'habitat intelligent appelé SIMACT [Bouchard et al, 2010]. Ce logiciel permet de simuler un environnement d'appartement en 3D, de définir des capteurs virtuels et de brancher un agent logiciel à l'écoute de ces déclenchements virtuels. Ainsi, la virtualisation de l'environnement devient transparente pour l'agent, puisque celui-ci reçoit approximativement les mêmes données d'entrée que lorsque qu'il écoute les événements en provenance d'un habitat réel. L'avantage est d'avoir un environnement contrôlé permettant de rejouer des scénarios préenregistrés modélisés à partir d'expérimentation antérieures réelles. Les résultats de ce processus de simulation sont, par ailleurs, facilement comparables car les scénarios sont toujours identiques et produisent les mêmes déclenchements de capteurs à chaque fois au même moment.

Dans la première partie de cette section, nous allons présenter l'architecture logicielle et l'implémentation réalisée de notre modèle. Cette partie correspond à notre contribution pratique au domaine scientifique. Ensuite, nous allons présenter notre protocole expérimental ainsi que les résultats obtenus aux termes d'une expérimentation basée sur des simulations de scénarios de cas réels. Cette partie correspondra à notre apport expérimental au domaine.

4.3.1 Architecture logicielle de notre implémentation

L'architecture logicielle de l'implémentation proposée est décrite à la Figure 4.7. Ce système se compose en deux parties principales :

- Module de simulation SIMACT [Bouchard et al., 2010]
- Module de reconnaissance temporel

Cette architecture représente une chaîne acyclique, où les relations entre les différents modules peuvent être en un seul sens, comme en double sens. Les relations ayant un sens unique signifie que la donnée ou l'ensemble de données fournies sont non modifiables et que le module fournisseur peut présenter une indépendance du reste du système, comme par exemple, le module de simulateur d'activités, peut fonctionner de manière autonome à l'égard du reste des modules. Alors que les relations en double sens signifient que les parties qui entretiennent de telles relations, peuvent présenter une indépendance mutuelle et que les données inter-changées sont indispensables à chacun d'eux, qui est le cas de notre implémentation avec la librairie d'activités.

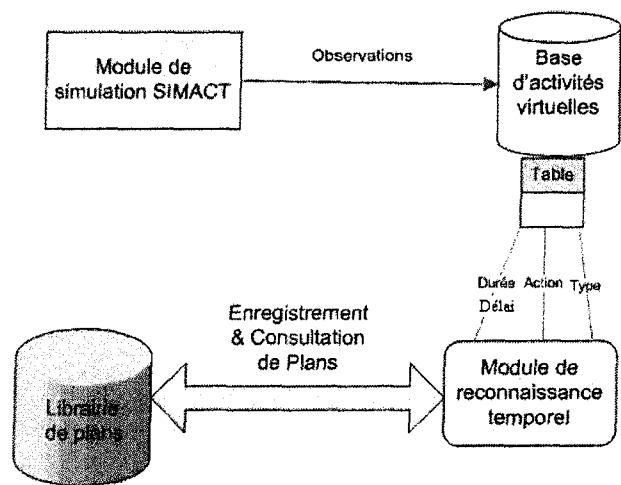


Figure 4.7 Architecture logicielle de l'implémentation proposée

De façon synthétique, les différents composants de l'architecture interagissent comme suit. Le module de simulation SIMACT charge un scénario et en fait la simulation. Une visualisation 3D est produite de la séquence d'événements et, pour chaque déclenchement d'un capteur virtuel, une entrée est envoyée à la base de données. Cette partie est indépendante. L'agent de reconnaissance constitue un composant à part. SIMACT est conçu afin de pouvoir interagir de façon simple et efficace avec différentes applications produites avec différents langages. Ainsi, une application n'a qu'à se connecter à la base de données d'événements afin d'extraire en temps réel les événements de déclenchement inscrits par SIMACT. Ce découplage est très avantageux au niveau applicatif. De son côté, l'agent de reconnaissance se connecte donc à la base de données de SIMACT et va extraire, à toute les demi-secondes, les nouveaux événements de la base de données. L'agent possède une librairie d'activités. Chaque activité de la librairie est modélisée selon le modèle vu précédemment. Les activités sont organisées en taxonomie grâce à la relation de subsumption. Cette librairie comprend une sous-librairie d'actions. La librairie est construite comme étant une extension logicielle du système Powerloom [Powerloom, 2002]. Ce système permet la représentation et la gestion de connaissances en logique de description et implémente un algorithme pour la vérification des relations de subsumption. Notre librairie encapsule Powerloom, lui déléguant l'ensemble des définitions en LD aux niveaux des concepts et des rôles de bas-niveau. Les modèles d'activités et d'actions sont quant à eux gérés dans une couche logicielle supérieure, implémentée suivant le modèle théorique présenté dans la première partie du chapitre. Nous allons maintenant décrire plus précisément les détails concernant l'implémentation de chacune des parties du système.

4.3.2 Le simulateur d'habitat intelligent SIMACT

Afin de nous permettre de conduire des expérimentations dans un environnement simulé avec des données réelles, le laboratoire de recherche [Bouchard et al, 2010] a conçu un outil logiciel appelé SIMACT, permettant de simuler les infrastructures d'un habitat intelligent avec tout ses senseurs. L'architecture de ce simulateur permet à un *agent de reconnaissance intelligent* de se connecter facilement sur une base de donnée pour recevoir, en temps réel, les entrées d'un ensemble de senseurs simulés. Des scénarios préenregistrés sont inclus dans l'outil, permettant ainsi de rejouer des épisodes tirés de cas réel. Ceux-ci proviennent des essais réels effectués avec des sujets humains à l'été 2010 [Laprise et al., 2010] et d'études antérieures effectuées au Centre de Recherche sur les Habitats Intelligents de l'Université de Sherbrooke. De nouvelles expérimentations, visant entre autre à avoir plus de scénarios significatifs, sont en cours de réalisation à l'UQAC dans le courant de l'été et de l'automne 2010, sous la supervision d'Audrey

Potvin, étudiante au doctorat en psychologie effectuant son projet de recherche au laboratoire. L'objectif de l'outil SIMACT est de fournir un outil permettant de tester des algorithmes de reconnaissance dans un environnement virtuel contrôlé. Le simulateur SIMACT a été réalisé comme étant un logiciel « open source » qui sera disponible bientôt en téléchargement sur le site web du laboratoire. Ainsi, les chercheurs du domaine pourront exploiter cet outil pour leurs expérimentations, donnant de ce fait une base de comparaison idéale pour la performance des algorithmes. et qui nous a permis de l'intégrer dans l'application que nous avons développé. Nous espérons que, dans le futur, cet outil sera largement utilisé par les chercheurs ne possédant pas d'infrastructure réel et ayant un accès restreint à de véritables patients. La figure 4.7 illustre l'interface utilisateur de l'outil SIMACT.

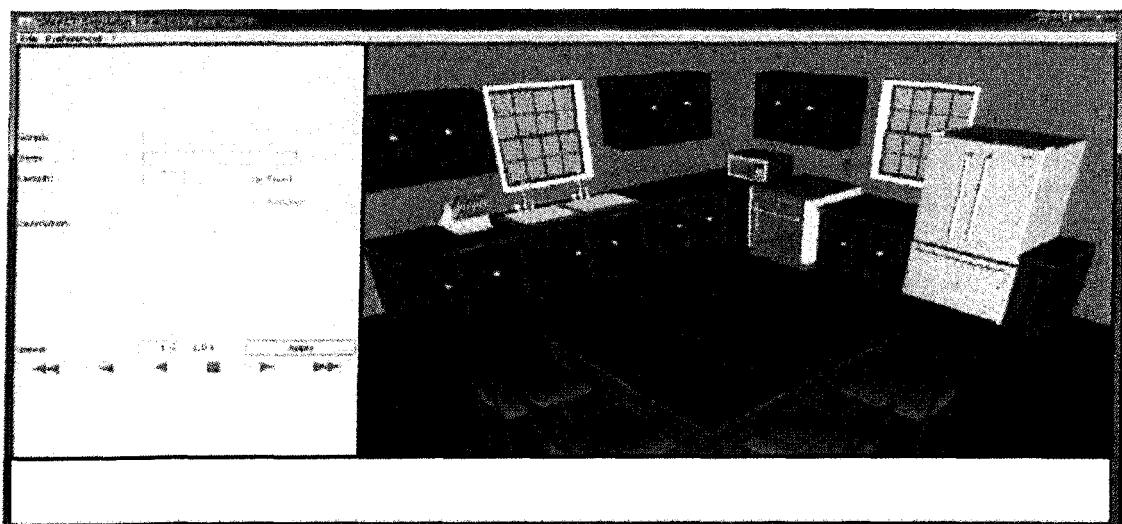


Figure 4.8. L'outil de simulation d'habitat intelligent SIMACT

Le principal bénéfice de ce type d'outil, est le fait qu'il est capable de simuler n'importe quel activité effectuée dans un habitat intelligent, en reprenant les mêmes séquences d'actions et déclenchement de capteurs. Nous avons donc utilisé l'outil SIMACT comme plateforme logiciel d'expérimentation pour tester l'agent de reconnaissance développé avec le nouveau modèle étendu de Weida.

4.3.3 Modèle d'activité de la vie quotidienne utilisée pour les tests

Pour tester adéquatement notre approche de reconnaissance d'activités, il faut d'abord se poser la question : quelles activités modélisées ? Dans notre contexte, nous souhaitons bien sûr disposer d'activités représentatives de la réalité quotidienne d'un patient Alzheimer. La littérature

sur les habitats intelligents nous fournit plusieurs pistes intéressantes concernant le choix d'activités représentatives. Par exemple, des activités d'hygiène dans la salle de bain comme le lavage des mains [Mihailidis et al., 2007], des activités de préparation d'un repas tel que des spaghetti [Bouchard et al., 2007], des activités propre à la routine du matin comme la préparation d'un thé ou d'un café [Pigot et al., 2003], etc. Les modèles d'activités choisis pour les expérimentations doivent également être simples et faciles à expérimenter dans un contexte de recherche. En ce sens, ces activités doivent inclure un nombre suffisant d'étapes mais limité d'étapes. Autrement dit, de petites activités représentatives mais simples. Ces activités devraient aussi être réalisables dans une durée de temps raisonnable dans l'éventualité de tests avec des sujets réels. Un temps d'environ 10 minutes maximum serait l'idéal [Laprise et al., 2010]. Finalement, il faut posséder des données réelles concernant les activités choisies. Afin de satisfaire tous ces besoins, nous avons basé notre choix d'activités sur les recommandations du *Naturalistic Action Test* (NAT) [Schwartz et al, 2002]. Ce test cognitif éprouvé est conçu pour évaluer la performance de l'individu en exécutant des tâches simples effectuées dans une cuisine en utilisant des objets légers, facilement déplaçables. On peut disposer les différents objets sur une table de 1 mètre par 1,5 mètres et effectuer les expérimentations dans n'importe quel local. Pour les fins d'expérimentations de cette étude, deux activités ont été sélectionnées : *faire du café* et *préparer des rôtis*. Ces activités sont intéressantes car elles se réalisent en moins de 10 minutes, comporte un nombre d'étapes suffisants mais limité, et on peut faire interagir les deux activités ensemble. Les objets impliqués dans ces activités sont : un carton de lait, un pot de sucre, une carafe d'eau chaude, un pot de café instantané, un sac de pain, un grille-pain, un couteau à beurre, un beurrier avec du beurre. Dans la configuration expérimentale réelle [Rocher et al., 2010], chacun de ces objets est « étiqueté » avec un « tag RFID² » et ses déplacements sont suivis grâce à deux antennes, disposés sur les coins de la table, qui utilisent la puissance du signal pour trianguler la position et les déplacements de l'objet dans les différentes zones. Dans les expérimentations traitées dans ce mémoire, le déclenchement des capteurs liés aux différents objets est simplement simulé par SIMACT sur la base de tests réels antérieurs.

La figure 4.8 schématise, étape par étape, le modèle résumant les différentes façons de réalisées les deux activités choisies. Ce modèle a été conçu à la lumière des expérimentations réalisées avec des sujets réels normaux. Il est utilisé afin de déterminer si l'exécution de l'activité est correcte. Plus précisément, les informations recueillies lors de ces expérimentations

² La radio-identification, souvent appelée technologie RFID (de l'anglais Radio Frequency IDentification), est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés (« RFID tag »).

antérieures permettront de vérifier si l'ordre des étapes est correcte, si la durée des étapes et les délais entre celles-ci est adéquate (le paramètre qui nous intéresse), etc.

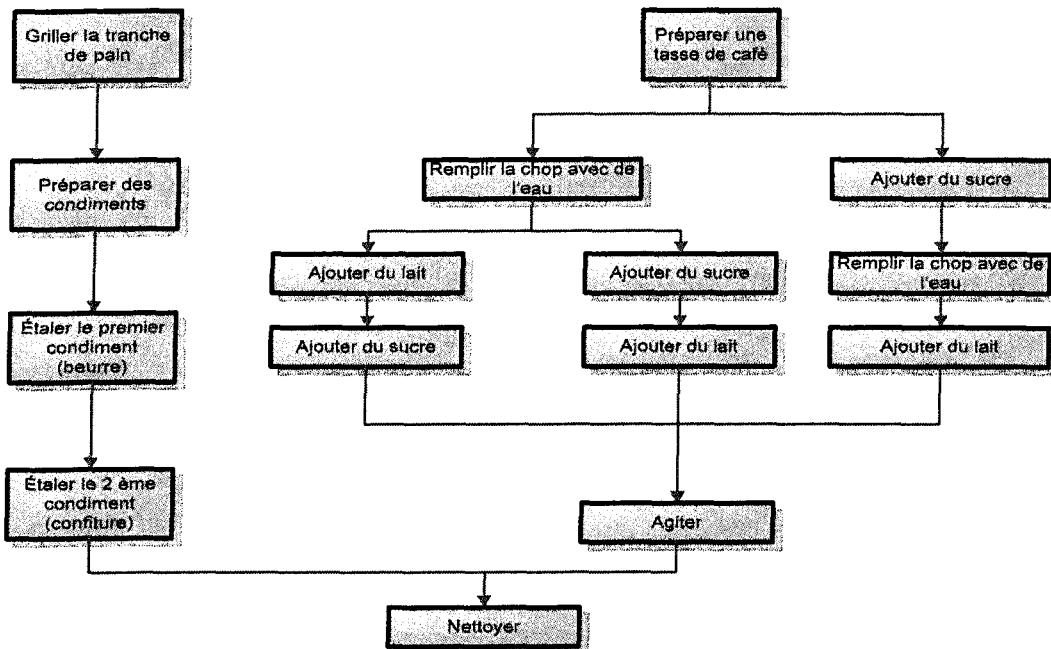


Figure 4.9. Modèle des activités choisies : *faire du café et préparer des toasts*

4.3.4 Approche expérimentale utilisées pour collecter les données de SIMACT

L'approche qui a été adoptée dans SIMACT pour collecter les données utiles peut être divisée en deux phases. La première a consisté à conduire des essais cliniques en utilisant les deux activités choisies du NAT avec des sujets normaux. L'objectif de cette première phase était de rassembler les données appropriées concernant l'exécution normale de chaque étape de l'activité, l'ordre habituel des étapes, les erreurs commises par les gens normaux, les hésitations, la différence entre les sexes, les âges, notamment au niveau des temps de réalisation des étapes et des délais entre celles-ci. Ces données sont celles utilisées pour tester notre algorithme et pour la comparaison avec le comportement anormal pouvant être observé. La deuxième phase consiste à mener une expérience identique à la première mais cette fois-ci, avec des patients atteints d'Alzheimer. Le but de cette deuxième phase est de collecter des données spécifiques concernant les exécutions erronées des réalisations de ces activités causées par des troubles cognitifs. Ceci permet de mieux comprendre le comportement de tels patients et d'avoir un outil concret afin de leur venir en aide pour adapter la technologie d'assistance aux besoins particuliers de ce genre de

patients. Dans le cadre de ce mémoire, se sont uniquement les données en provenance de la première étude [Laprise et al., 2010] était disponible. Nous avons donc utilisé les scénarios de réalisation de chacun des sujets normaux pour calculer les intervalles « normaux » de réalisation des étapes et des délais. Nous avons ensuite repris ces scénarios de réalisation et modélisé un ensemble de scénarios temporellement erronés au niveau des délais et durées, afin de tester l'approche.

4.3.4.1 Expérimentation réalisée avec les 20 sujets normaux

Les résultats de cette expérimentation [Laprise et al., 2010] réalisée au laboratoire constitue la base de notre protocole de tests. Le laboratoire a effectué un recrutement, au printemps 2009, d'un ensemble de sujets normaux sur une base volontaire à l'UQAC. Les participants devait être âgés entre 18 et 77 ans. Le critère le plus important été l'absence de tout déficit cognitif. Les critères permettant d'exclure une personne étaient, par exemple, les cas psychiatriques ou neurologiques, l'utilisation de médicaments délivrés sur ordonnance, dépression, etc. Les participants n'avaient pas le droit de consommer des boissons alcooliques durant les 24 heures qui précédent le test. Un total de 20 participants ont été sélectionné pour exécuter les deux activités décrites précédemment. Le protocole expérimental se déroulait alors comme suit : le participant prend place sur une chaise, tout le matériel requis pour la complétude des activités est déposé sur la table qui est devant lui. Le participant est libre d'utiliser n'importe quel moyen afin de compléter les deux tâches, dans l'ordre souhaité, D'abord, les participants sont appelés à préparer une tasse de café avec le sucre et du lait et préparer un toast avec le beurre et la confiture. Chaque essai à été enregistré à l'aide d'une caméra vidéo et temporisé. Les vidéos ont préservé l'anonymat du participant, en masquant son visage sur l'enregistrement. Ceux-ci peuvent être rejoués afin d'en extraire des données utiles. Une fois qu'un essai est terminé, l'ordre et le temps de chaque étape sont évalués et collectés de manière individuelle. L'analyse statistique (l'analyse modale et moyenne, la comparaison moyenne et corrélation) a été effectuée pour déterminer s'il y avait une différence de sous-groupe, par exemple, s'il s'agissait d'une différence dépendant du genre ou l'âge du patient. À titre d'exemple, une différence significative a été notée concernant les temps de réalisation des hommes versus celui des femmes. Ceci peut être lié à divers facteurs, dont l'habitude dans la cuisine. Ces éléments doivent cependant être pris en compte afin de ne pas détecter un délai anormal chez un groupe de gens dont la rapidité d'exécution est plus lente. À ce titre, on note que le temps d'exécution nécessaire semble plus important chez les groupes plus âgés.

4.3.4.2 Expérimentation avec 20 patients Alzheimer

Le LIARA a signé récemment une entente de collaboration formelle avec le CSSS Clephas-Claveau de l'arrondissement La Baie. Ce centre est en mesure de fournir un groupe adéquat de patients Alzheimer pour les tests car il s'occupe de diagnostiquer les cas de toute la région. Une équipe multidisciplinaire composée d'étudiants en psychologie et d'une chercheure en neuropsychologie supervisera cette seconde phase de test, actuellement en cours.

4.3.5 L'application de reconnaissance temporelle

Maintenant que nous avons fait le tour d'horizon de la partie simulateur et données, nous allons discuter de l'implémentation réalisée de l'approche temporelle. L'application développée permet de faire une démonstration de viabilité de l'approche proposée. Celle-ci est composée d'une interface graphique permettant la *Création de plans*, comme on peut le voir à la Figure 4.9. Cette interface constitue l'outil avec lequel l'utilisateur peut construire et modifier sa base de connaissance d'actions (taxonomie d'actions) représentant les étapes possibles des activités. Par la suite, il peut utiliser les concepts d'actions et de cette base de connaissance afin de former des plans temporels et d'imposer des limites temporelles entre chaque couple d'actions. Cet outil graphique de modélisation d'activités est construit en Java. Elle peut être divisée en deux principales sections :

- La première section se charge de la création et la modification de la base de concepts d'actions.
- La seconde section qui se focalise sur la modélisation des modèles d'activités temporelles qui pourront être, par la suite, enregistrés dans la base afin d'être consulter plus tard par le module *Gestion de plans*.

Dans la première section, on remarque que la base de concepts d'actions est construite sous forme d'arbre grâce au composant *JTree* de Java, qui peut être considéré comme un vrai aperçu, pour l'utilisateur, de la base d'actions terminologique et qui nous a permis de gérer les actions comme étant des nœuds, étant donnée que nous considérons dans cette base que chaque concept d'action admet qu'un seul subsumant, c'est-à-dire, un seul père dans l'arbre d'actions, ce qui rend la manipulation des actions moins complexe. Chaque représentation structurelle des actions (nœuds de l'arbre) est transformée en une série de définitions terminologiques équivalentes en utilisant la logique de description. Cette traduction est en partie envoyée à PowerLoom à l'aide d'un API, permettant ainsi le dialogue direct entre celui-ci et la base de connaissances. Ainsi, la

partie LD pure est complètement déléguée à Powerloom. Chaque fois une définition de concept est envoyée à PowerLoom, celui-ci classe ce nouveau concept parmi la taxonomie de concepts déjà existants. Si on prend à titre d'exemple les actions se trouvant dans l'arbre de la figure 4.9, on peut dire alors que le concept d'action *mouvement* est de type *Action* et que le concept d'action *Le-gâteau-est-déposé* est de type *mouvement*.

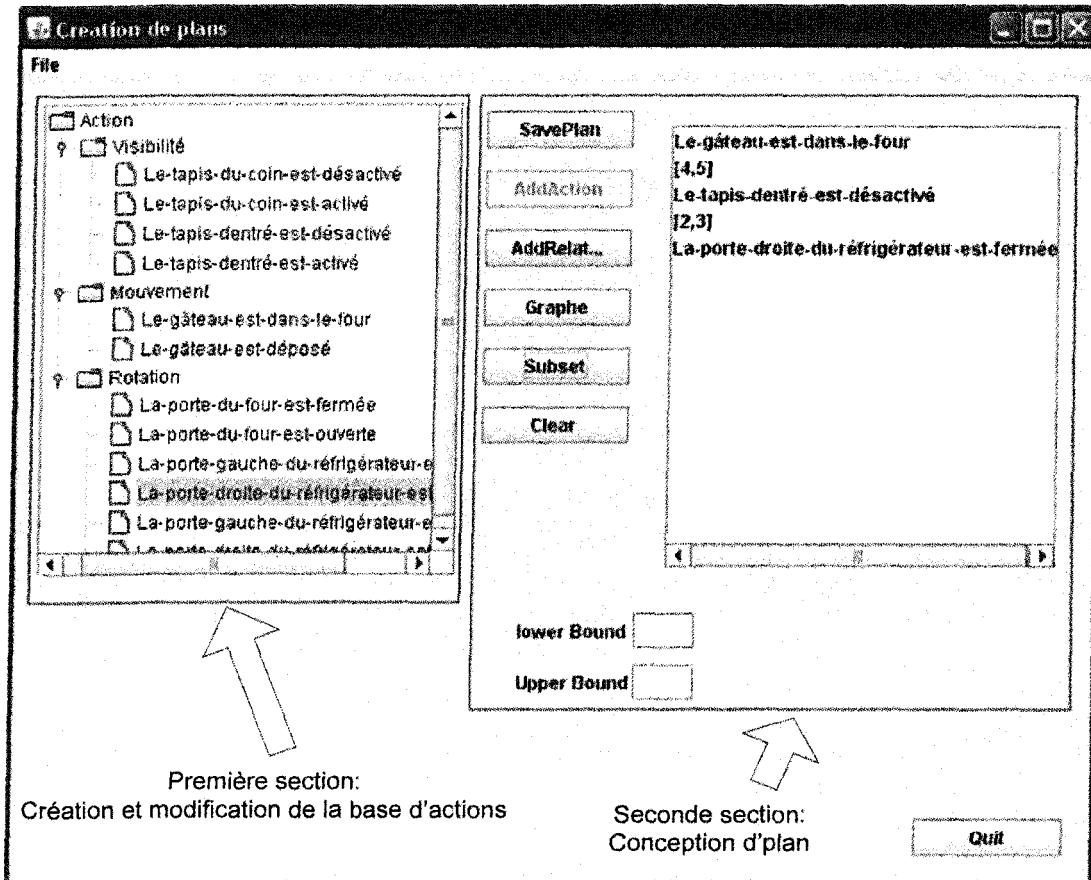


Figure 4.10. L'application de création des plans

La deuxième section de l'application *Création des plans* a comme objectif de permettre la modélisation d'activité au niveau temporelle, c'est-à-dire une structure regroupant un ensemble de concepts d'actions créés dans la première section et un ensemble d'intervalles concernant les délais et les durées à partir des champs de saisie *Lower Bound* et *Upper Bound* de chacun de ces concepts d'actions. Grâce aux boutons *addAction* et *addRelation*, l'utilisateur effectue un ajout séquentiel pour former le plan voulu. Une fois modélisé, l'utilisateur peut valider le plan à l'aide du bouton *savePlan* qui est relié avec une fonction qui permet de vérifier l'exactitude syntaxique du plan, par la suite, il l'enregistre dans librairie d'activités, qui va gérer les actions, les relations

temporelles et les plans qui les référents d'une manière indépendante. Afin de visualiser le plan formé sous forme de graphe, les éléments (actions et relations temporelles) de celui-ci sont envoyés comme paramètres à l'aide du bouton *Graphe* et qui va permettre de générer le graphe du plan modélisé, comme le montre la figure 4.10, grâce à l'outil JGraph.

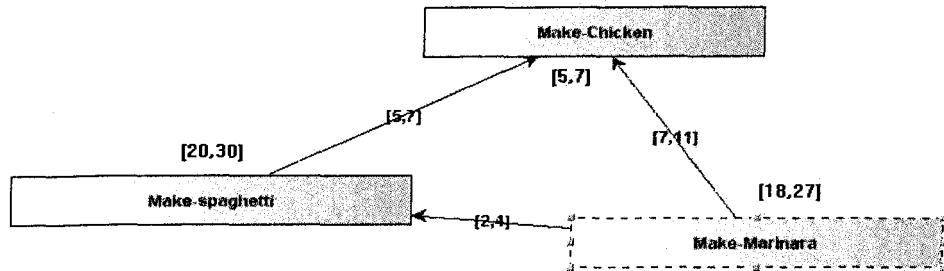


Figure 4.11. Un graphe généré à partir d'un modèle de plan

D'un point de vue général, l'application de création de plans permet ainsi de gérer notre base de connaissances concernant les actions et les activités. Elle permet d'introduire les contraintes temporelles au niveau des étapes des activités modélisés, de façon à ce par la suite, le système de reconnaissance puisse utiliser ces informations pour déterminer la cohérence d'un plan observé par rapport aux informations incluses dans la librairie.

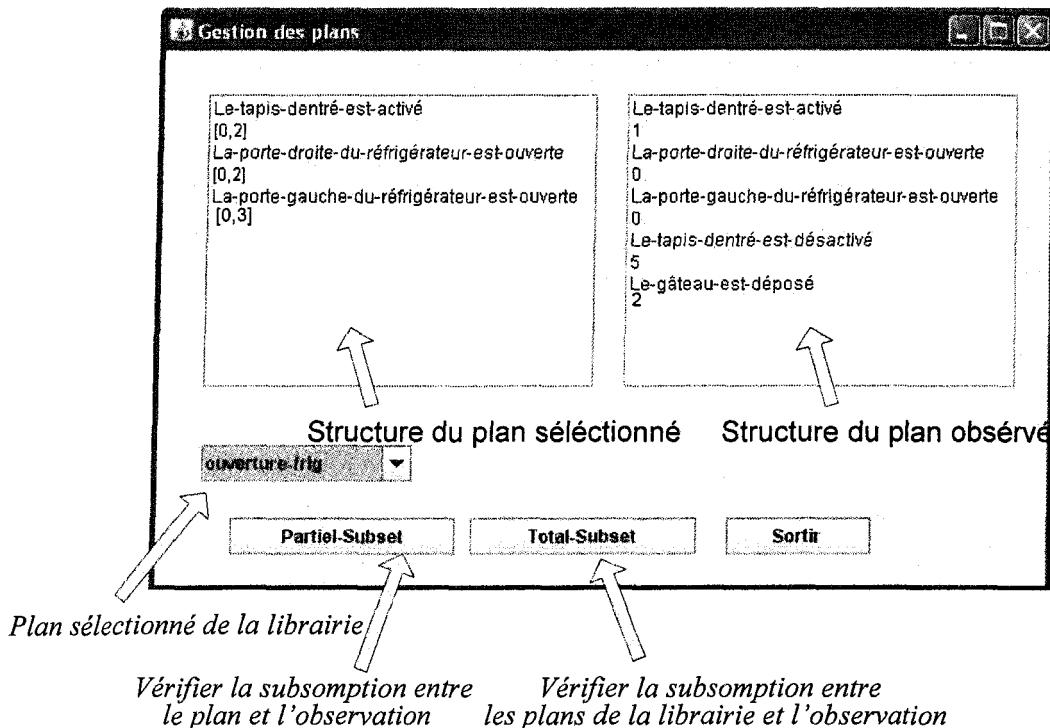


Figure 4.12. Fenêtre de gestion des plans

L'application propose également une fenêtre de gestion des plans, illustrée à la Figure 4.11, qui permet de vérifier manuellement la relation de subsomption entre un modèle d'activité de la librairie et une séquence d'actions observée. Cette section de l'application est à la base du module de reconnaissance temporelle, puisque cette vérification de la relation de subsomption entre un plan la librairie et une séquence d'actions observées est à la base du processus inférentiel. Ce processus se déroule comme suit : le module de simulation SIMACT fourni, lors du déroulement d'un scénario, la nouvelle action (étape) observée au processus d'inférence avec les temps de réalisation et de délais. Cette action est ajouté au modèle partiel d'activité construit à partir de la séquence d'actions observées. Le système exploite alors la relation de subsomption définie entre modèles d'activités. L'algorithme 4.1 résume cette partie du processus inférentiel.

```

Subsomption (P, O)
    Décomposer le plan P en n triplets
         $T_p(n) \leftarrow \text{Décomposer}(P)$ 
        Pour i (qui réfère le triplet courant) de 1 à n
            Extraire la première action de  $T_p(i)$ 
                Action1  $\leftarrow$  élément 1 de  $T_p(i)$ 
                Si Action1 ne subsume aucune des actions de  $o_j \in O$ 
                    Alors le plan P ne subsume pas l'observation O
                    Retourner faux
                Sinon extraire la deuxième action de  $T_p(i)$ 
                    Si l'intervalle temporelle de Action1 ne subsume pas celle de  $o_j$ 
                        Retourner faux + erreur temporelle de durée
                    Action2  $\leftarrow$  élément 3 de  $T_p(i)$ 
                    Si Action2 ne subsume  $o_k \in O$  qui le succédant  $o_j$  subsumé par Action1
                        Si pas d'élément  $o_k$ , alors passer au triplet  $i+1$  suivant
                        Sinon le Plan P ne subsume pas l'observation O
                        Retourner faux
                    Sinon extraire l'intervalle temporel de  $T_p(i)$ 
                        Si l'intervalle temporelle de Action2 ne subsume pas celle de  $o_k$ 
                            Retourner faux + erreur temporelle de durée
                        Intervalle  $\leftarrow$  élément 2 de  $T_p(i)$ 
                        Si les bornes inférieure/supérieure ne sont pas vérifiées
                            Alors Retourner faux + erreur temporelle de délai.
                    Boucle -> Passer au triplet suivant
    Fin

```

Algorithme 4.1. Vérification de subsomption d'un plan de la librairie avec une observation

Afin de vérifier la relation de subsomption entre un plan de la librairie P et un plan observé O, nous avons géré P comme étant une séquence composée d'un ou plusieurs triplets. Chaque triplet est composé de deux actions reliées par un intervalle temporel de la manière suivante : *action intervalle action*. Il faut vérifier la correspondance entre chaque triplet issu de P avec son correspondant provenant de O. Ainsi, pour la première action de chaque triplet de P, on cherchera une action correspondant (subsumée) dans O. Lorsque c'est fait, on vérifie si la durée de cette action concorde avec l'intervalle définie dans P. Si ce n'est pas le cas, on générera alors une erreur de type « durée anormale » pour l'action observée. Sinon, on comparera la deuxième action du triplet avec le successeur de l'action identifiée dans O. S'il n'existe pas de successeur, c'est que le plan observé n'est encore rendu à cette étape. Puisqu'il s'agit d'un plan partiellement complété, et que la première moitié subsume et semble correcte, on considérera alors le résultat positif. S'il existe un successeur dans O à l'action trouvée à la première étape dans P, on effectuera alors la vérification de la relation de subsomption entre celle-ci et la seconde action du triplet. Si c'est négatif, on génère une erreur, il n'y pas de subsomption. Si c'est positif, on vérifie alors si la durée de l'action observée est incluse dans l'intervalle définie dans P. Si ce n'est pas le cas, on génère une erreur de « durée anormale ». Enfin, on vérifie la correspondance entre l'intervalle représentant le délai entre les deux actions du triplet dans P, et le délai observé entre les deux actions dans O. Si ce délai observé n'est pas inclus dans l'intervalle, on génère alors une erreur de « délai incohérent entre deux actions ».

4.3.6 Étude comparative des performances du nouveau modèle

Afin de mesurer l'apport temporel du modèle proposé dans ce mémoire, nous avons procédé à une phase d'expérimentation et effectuée une étude comparative des performances du modèle. Pour ce faire, nous avons implanté, pour fins de comparaisons, deux autres modèles de reconnaissance. Nous avons évidemment implanté le modèle original de Weida [Weida, 1993] intégrant les contraintes temporelles symboliques qui s'appuie sur l'approche temporelle d'Allen. Nous avons aussi implanté une version du modèle de Weida faisant complètement abstraction de l'aspect temporel. L'objectif expérimental était d'en arriver à démontrer, d'une part, le caractère essentiel de l'aspect temporel et, d'autres part, de montrer que notre prise en compte du temps permettait d'obtenir de meilleurs résultats que l'approche originale de Weida. Par conséquent, nous avons procédé à une série d'expérimentations avec les trois modèles en simulant un ensemble de scénarios tirées de cas réels servant comme données d'entrées pour les observations au sein d'un habitat intelligent virtuel. Les simulations de scénarios ont été

réalisées grâce à l'outil SIMACT, que nous avons présenté précédemment. Les données expérimentales (scénarios) que nous avons utilisées proviennent des expérimentations effectuées à l'été 2009 par notre équipe avec des sujets réels.

Pour les fins d'expérimentation, une librairie composée de 30 modèles d'activités reliées à des tâches de cuisine (préparation de repas, faire du café, du thé, etc.) a été constituée. Plusieurs actions sont partagées à travers plusieurs activités. Par exemple, l'action *faire bouillir de l'eau* est présente dans le modèle des activités : *faire du café*, *faire du thé*, *préparer des spaghetti alfredo*, etc. Pour la simulation, 10 scénarios différents de réalisations correctes des activités *faire du café* et *préparer des rôtis* ont été joués avec SIMACT. Par la suite, 10 scénarios de réalisation des mêmes activités, mais cette fois temporellement erronées (des erreurs de durée et de délai ont été introduites dans chacun des scénarios), ont été joué avec SIMACT. Le premier objectif expérimental était de voir, avec les dix réalisations correctes, si le nouveau modèle permettait de mieux circonscrire les plans possibles au terme du processus de reconnaissance grâce au raffinement temporelle introduit dans le mémoire. Par la suite, en jouant les 10 réalisations erronées, le second objectif était de voir si le modèle était apte à reconnaître les erreurs de nature temporelle et de les exhibées.

4.3.7 Présentation des résultats

La table 4.1 résume les résultats obtenus pour la première phase expérimentale réalisée en jouant les 10 scénarios de réalisations correctes. Rappelons que lors du déroulement d'un processus de reconnaissance logique, comme c'est le cas du modèle ici, le résultat de la reconnaissance correspond à un sous ensemble de plans de la librairie [Bouchard et al., 2007]. En d'autres mots, le processus permet de circonscrire à travers tous les plans possibles de la librairie, avec plus ou moins de précision, un sous ensemble de plans plausibles en fonction des observations. En ce sens, le tableau 4.1 présente le nombre moyen de plans inclus dans ce sous ensemble résultant pour chacun des modèle testés. Dans un premier cas, celui du modèle de Weida sans prise en compte du temps, on note que le résultat moyen est de 19 modèles d'activités plausibles au terme du processus de reconnaissance. Autrement dit, en faisant seulement la correspondance entre les actions observées dans les scénarios joués et les modèles d'activités de la librairie, on arrive à éliminer 11 hypothèses, mais 19 plans de la librairie reste, en moyenne, plausible au terme du processus. On peut donc voir que le fait de ne pas tenir compte des éléments temporels observés rend le processus de reconnaissance imprécis. En appliquant cette fois le modèle temporel original de Weida, on voit que le nombre d'hypothèses

plausibles au terme du processus connaît une réduction significative, passant en moyenne à 8 / 30. Ceci est explicable par le fait que la prise en compte de l'ordonnancement temporel émanant des relations d'Allen permet d'éliminer un nombre significatif d'hypothèse en vérifiant si les scénarios observés respectent ou non une séquence correcte. Par ailleurs, lorsque l'on introduit notre version étendue du modèle, on obtient une moyenne de 3 plans possibles sur 30 au terme du processus, soit une circonscription très précise de l'espace des plans de la librairie. Ceci est dû essentiellement au pouvoir quantitatif d'un raisonnement basé sur des contraintes numériques. Ainsi, la vérification de la validité des durées et des délais vient s'ajouter à la validation du séquencement correct des actions pour aboutir sur un résultat plus raffiné.

	Modèle de Weida (sans prise en compte du temps)	Modèle de Weida (relations d'Allen)	Modèle étendu (relations numériques)
Moyenne du # d'activités possibles	19 / 30	8 / 30	3 / 30

Table 4.1. Moyenne du nombre d'activités possibles au terme du processus de reconnaissance pour chacune des versions du modèle de reconnaissance

Dans l'expérimentation du premier modèle, on note qu'en moyenne 11 plans de la librairie ont pu être rejetés, soit environ 36,5%. Ces plans rejetés comportaient tous des actions qui ne vérifiaient pas, en totalité, une relation de subsumption avec les actions qui les correspondent à la séquence observée. La figure 4.12 permet d'illustrer visuellement la proportion de plans plausibles retenus au terme du processus versus ceux rejetés par celui-ci.

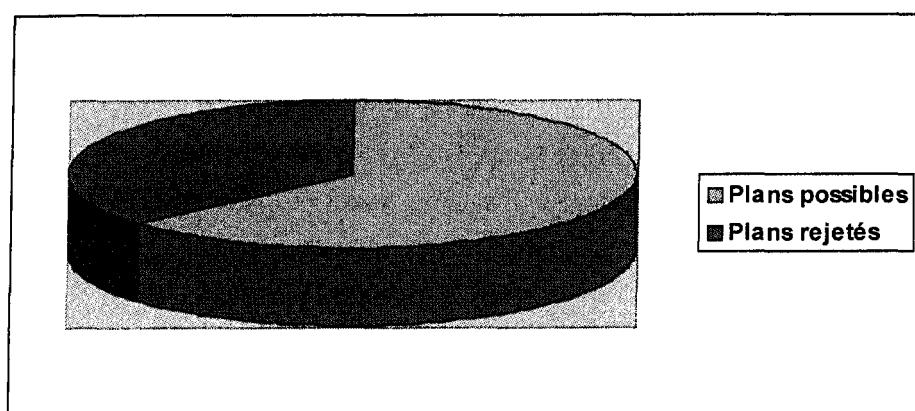


Figure 4.13. Proportion d'hypothèses rejetées versus celles identifiées comme plausibles pour le modèle de Weida sans prise en compte du temps

Dans le cas du deuxième modèle expérimenté, soit la version temporelle originale de Weida, 22 plans (73%) ont pu être rejetés par le processus de reconnaissance, augmentant ainsi significativement la précision du modèle. La figure 4.13 nous permet de comprendre cette amélioration de la précision en montrant la proportion de ces plans qui ont été éliminé grâce à la vérification des contraintes temporelles d'Allen introduites sur les modèles d'activités de la librairie. Ainsi, 11 plans ont été rejetés (36,5%) grâce à la correspondance (subsumption) entre l'ensemble des actions observées et celles incluses dans les modèles de plans de la librairie et 11 autres ont pu être écartés grâce aux contraintes temporelles à caractère qualitatif qui s'appuient sur les relations d'Allen. Celles-ci ont donc permis de filtrer 36,5% des hypothèses possibles. L'apport de la prise en compte du temps est donc clairement mis en lumière à ce stade.

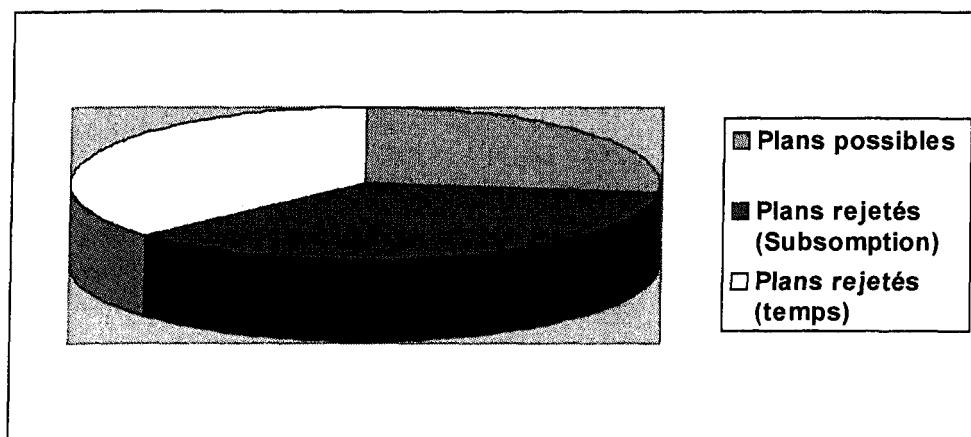


Figure 4.14. Proportion d'hypothèses rejetées en raison des contraintes temporelles versus les autres rejetées et celles identifiées plausibles pour le modèle original de Weida

Enfin, dans le cas de l'expérimentation de notre nouvelle version étendue du modèle de Weida, 27 hypothèses en provenance de la librairie ont pu être éliminé, soit environ 90% de la totalité des plans de la librairie. La Figure 4.14 résume la répartition de cette performance. Ainsi, on peut voir qu'en moyenne 11 plans ne vérifiaient pas la relation de subsumption entre actions (36,5%). Par la suite, une moyenne de 4 autres ont pu être écartés en raison d'anomalies de nature temporelle (13,5%) liées aux durées des actions (insuffisance de temps). De plus, une moyenne de 12 autres plans ont pu être éliminés en raison d'anomalies liées aux délais (débordement) entre les actions observées (40%). En ce sens, on peut voir que l'apport des contraintes temporelles numériques est très utile car il permet d'augmenter la précision du processus de reconnaissance par rapport à l'utilisation des contraintes symboliques.

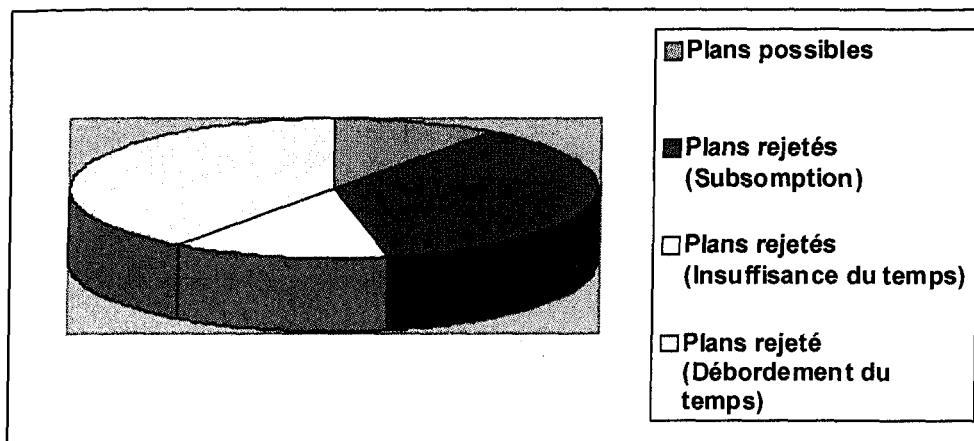


Figure 4.15. Proportion d'hypothèse rejetées en raison des contraintes temporelles numériques versus les autres rejetées et celles identifiées plausibles pour le modèle du mémoire

Dans cette première phase expérimentale, les 10 scénarios joués étaient corrects, c'est-à-dire que l'activité simulée comportait une réalisation cohérente avec les bonnes actions, réalisées à l'intérieur de délais et de durées inclus dans des intervalles « normaux ». L'objectif de cette première phase était de mesurer le gain en précision du processus de reconnaissance obtenu grâce à la vérification des contraintes de temps numériques.

La deuxième partie de notre expérimentation visait à vérifier la capacité de notre approche à identifier les réalisations erronées d'une activité. En ce sens, nous avons rejoués les 10 scénarios de la première phase en introduisant, dans chacun d'entre eux, une erreur temporelle de durée (par exemple une durée d'action excessivement longue) ou de délai (un délai insuffisant ou trop long entre deux actions). Pour cette seconde phase, nous avons seulement testé notre nouvelle version étendue du modèle car les deux autres versions implémentées n'avaient tout simplement pas la capacité de reconnaître ce genre d'erreur. Le résultat de cette phase fut tel qu'attendu, c'est-à-dire que toutes les erreurs ont pu être identifiées par le processus de reconnaissance. La façon de procéder était simple : si au moins un plan de la librairie subsume entièrement la séquence d'actions observées, c'est donc qu'il existe au sein de celle-ci un modèle d'activité pour lequel une réalisation correct est en cours. Ainsi, le processus regroupe l'ensemble des hypothèses plausibles (les subsumants de la séquence observée) et déclare qu'une activité correctement réalisée est en cours et que celles-ci correspondent à l'une des hypothèses identifiées. Dans le cas où aucun modèle d'activités de la librairie ne subsume entièrement la séquence observée, pour des raisons de dérogation aux contraintes temporelles numériques imposées, alors le processus déclare qu'une réalisation temporellement erronée d'une activité est en cours. De cette façon, les erreurs de nature temporelle peuvent être identifiées par le processus.

4.4 Positionnement de ce travail au sein du LIARA

Ce travail d'initiation à la recherche avait pour but d'identifier les limitations des approches de reconnaissance existantes au niveau de la prise en compte du temps et de dégager des pistes de solutions intéressantes pour intégrer cette aspect au sein d'un modèle de reconnaissance basé sur la logique de description. Il avait également pour objectif de faire des essais d'implémentation et de validation de ces pistes de solutions. En ce sens, ce travail s'inscrit dans une perspective plus large dans le grand schème des projets en cours au laboratoire LIARA. Il constitue en fait l'une des briques dans le projet de développement d'une système de reconnaissance et d'assistance adapté permettant de fournir des services à une personne en perte d'autonomie dans un habitat intelligent. Ce système, que l'on peut voir à la Figure 4.15, est développé conjointement par notre équipe et celle du Centre de Recherche sur les Habitats Intelligents (CRHI) de l'Université de Sherbrooke. Il représente un résultat majeur au niveau de la recherche et du développement.

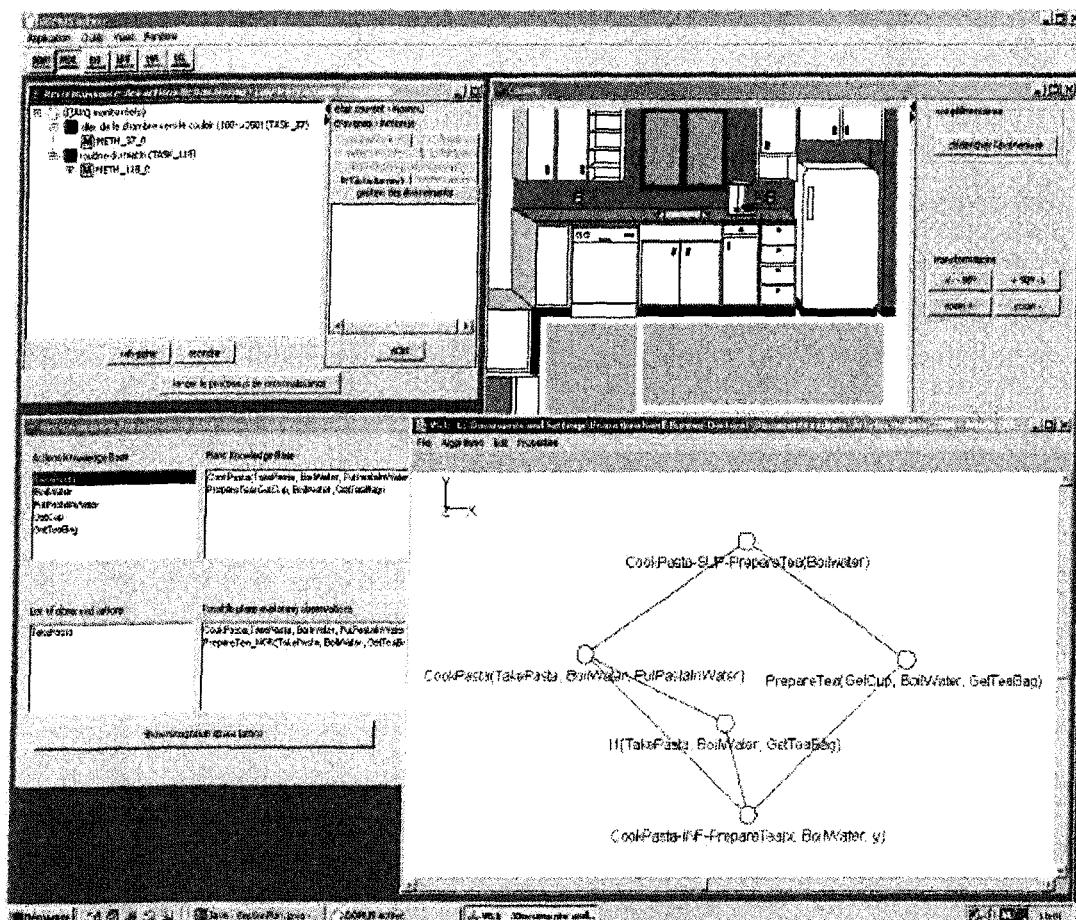


Figure 4.16. Système de reconnaissance d'activités développées par notre équipe. L'image montre l'application utilisée dans les infrastructures du CRHI de l'Université de Sherbrooke.

Le projet réalisé dans ce mémoire fournira ainsi une piste pour l'intégration de l'aspect temporel à l'intérieur de ce système qui fonctionne actuellement dans les infrastructures prototype du CRHI. Ce système exploite actuellement un modèle de reconnaissance basée sur la logique de description et sur la théorie des possibilités [Roy et al., 2009]. Il suit les traces des modèles hybrides de reconnaissance et peut être considéré comme étant une extension possibiliste du modèle logique élaboré à l'origine par [Bouchard, 2006], lequel est basé sur la théorie des treillis [Davey et al, 2002] et un cadre d'action formel [Bouzouane et al, 2005] fondé sur la logique de description [Baader et al, 2003]. Ce modèle exploite ainsi la logique de description, comme nous le faisons dans ce mémoire, afin de transformer la problématique de reconnaissance en un problème de classification basé sur la relation de subsumption, permettant d'effectuer la correspondance entre un événement observé et son type d'action qui lui correspond. Dans ce modèle, le processus de correspondance organise au sein d'une structure de treillis l'ensemble des plans plausibles, permettant de rendre la relation incertaine entre ces plans possibles plus explicites. Ceci permet de générer d'une façon dynamique un ensemble d'hypothèses qui porte sur les comportements erronés ou entrecroisés d'un patient souffrant d'Alzheimer, causé par son déficit cognitif [Bouchard et al., 2007].

Toutefois, le système actuellement en cours d'utilisation au CRHI exploitant ce modèle possède certaines faiblesses. Parmi elles, le fait qu'on ne tienne pas compte des durées et des délais des actions et donc, qu'on ne prenne en considération que partiellement l'aspect temporel. Pourtant, si on veut établir un modèle efficace de reconnaissance d'activités, il faut qu'il soit en mesure de reconnaître l'ensemble des erreurs de réalisation et en particulier celles de nature temporelle [Jakkula & Cook, 2007], qui constituent une nécessité conforme avec la réalité de l'habitat intelligent, surtout avec un patient affecté par un déficit cognitif qui peut perdre la notion du temps causant, par exemple, un incendie. En ce sens, l'état de l'art réalisé dans ce mémoire sur cette question particulière et la proposition faite contribue à l'avancement de cet aspect important dans le cadre du développement du système de reconnaissance et d'assistance du LIARA.

4.5 Bilan de la contribution de ce mémoire

Nous pouvons synthétiser le contenu de ce chapitre en faisant un bilan de la contribution de ce travail de recherche. Celle-ci suit les traces des approches de reconnaissance temporelles et dégage une problématique nouvelle liée à la prise en compte de l'aspect numérique du temps au niveau des durées et des délais entre les actions. Cette contribution revêt un volet théorique (un nouveau modèle étendu, un volet pratique (une implémentation) et un volet expérimental.

D'un point vue théorique, ce chapitre propose une extension du modèle de reconnaissance d'activités de Weida [Weida, 1993] dans laquelle nous avons incorporé, d'une part, la notion de concepts d'action en logique de description, qui permet de faciliter l'appariement des événements observés et, d'autre part, de tenir compte de l'aspect temporel lié aux durées et aux délais des actions, qui était absent du modèle original. La nouvelle structure de contraintes temporelles quantitatives proposée pour améliorer le modèle s'appuie sur les travaux de Dechter [Dechter, 1991], qui offrent un moyen efficace d'exploiter des intervalles de temps fermés en utilisant la notion d'instant. L'extension proposée permet ainsi d'augmenter la performance et la précision du modèle original de Weida tout en permettant l'identification de nouvelles anomalies liées au temps.

D'un point de vue pratique, ce chapitre propose une implémentation de l'approche avancée afin de démontrer sa faisabilité technique. Deux versions du modèle Weida ont également fait l'objet d'une implémentation, l'un intégrant l'aspect temporel avec les contraintes d'Allen et l'autre non. Ces modèles ont ainsi pu être comparés lors d'une expérimentation réalisée à l'aide d'un simulateur d'habitat intelligent [Bouchard et al., 2010], ainsi qu'à l'aide de données synthétiques tirées d'une étude clinique réalisée avec des sujets humains. Cette contribution pratique et expérimentale a permis de bien cerner les gains concrets au niveau de la performance, de la précision, et de la capacité de reconnaissance des erreurs liées aux durées et aux délais.

4.6 Conclusion du chapitre

Ce chapitre avait comme objectif de présenter notre contribution qui prend la forme d'un nouveau modèle étendu de reconnaissance d'activités temporelles, issu du modèle original proposé par Weida [Weida, 1993]. Dans un premier temps, nous avons introduit formellement les ajouts théoriques proposés comme extension au modèle de Weida. Cette section a permis de bien cerner l'apport théorique du mémoire et de comprendre les différentes opportunités que la logique de description nous permet d'établir. Cette partie a aussi démontré l'intérêt, à l'aide d'exemples simples, de notre proposition par son pouvoir de détection accru des erreurs de nature temporelle.

La seconde partie du chapitre a permis de présenter l'implémentation et la validation de ce nouveau modèle. En premier lieu, les différents aspects de l'architecture logicielle de cette implémentation ont été discutés. Le fonctionnement des différents composants a été présenté, ainsi que tous les outils permettant de réaliser les expérimentations avec ce modèle, notamment le simulateur SIMACT [Laprise et al., 2010]. Le système a été développé en suivant le modèle

théorique proposé en début de chapitre, afin d'étendre le modèle de Weida pour qu'il puisse prendre en considération les erreurs issues du temps de réalisations des actions.

La dernière partie du chapitre a permis de décrire le volet expérimental du projet. Le protocole expérimental et les résultats ont ainsi pu être présentés. Le choix des activités sélectionnée a aussi été discuté de même que la façon dont les données réelles ont été collectées lors de l'élaboration du simulateur d'activités d'un habitat intelligent virtuel SIMACT [Bouchard et al, 2010]. La présentation des résultats obtenus pour chaque phase de tests a permis de montrer l'apport de la nouvelle approche proposée au niveau de la reconnaissance d'activités. Finalement, nous avons montré comment la contribution du mémoire s'inscrit dans une perspective plus large et contribuera au développement d'un système d'assistance réalisé conjointement par notre équipe et celle du CRHI de l'Université de Sherbrooke.

CHAPITRE 5

CONCLUSION GÉNÉRALE

Le projet de recherche faisant l'objet de ce mémoire de maîtrise apporte des pistes de solutions à la problématique énoncée en introduction, qui porte sur la reconnaissance temporelle des activités de la vie quotidienne d'une personne en perte d'autonomie. Nous avons pu constater, à travers l'état de l'art présenté aux chapitre 2 et 3, que l'ensemble des informations temporelles observables (durée d'une action, laps de temps entre deux actions, etc.) dans un habitat intelligent doté de capteurs se révèle d'une importance capitale dans le processus de reconnaissance, notamment pour identifier certains types précis d'erreurs de réalisation. Nous avons aussi pu constater que ces éléments temporels sont souvent sous-exploités dans les travaux antérieurs [Singla et al., 2008], en raison de l'utilisation massive de la théorie temporelle d'Allen [Allen, 1983], qui ne tiennent compte que de l'ordonnancement des actions dans le temps. Afin de trouver des réponses à cette problématique, ainsi qu'aux limites des approches antérieures citées, nous avons proposé dans ce mémoire une extension au modèle de reconnaissance d'activités de Weida [Weida, 1993], dans lequel nous avons incorporé, d'une part, la notion de concept d'action en logique de description, qui permet de faciliter l'appariement des événements observés et des actions de librairie et, d'autre part, dans laquelle nous avons introduit une nouvelle structuration temporelle augmentée permettant de tenir compte de l'aspect quantitatif lié aux durées et aux délais des actions, qui était absent du modèle original. Cette nouvelle structure de contraintes temporelles quantitatives proposée s'appuie sur les travaux de Dechter [Dechter, 1991] et offre un moyen efficace d'exploiter les intervalles de temps fermés en utilisant la notion d'instants. Elle permet ainsi d'augmenter la performance et la précision du modèle original de Weida tout en permettant la détection de nouvelles anomalies.

comportementales liées au temps. Ce projet s'est effectué à travers la réalisation de trois principales phases de recherche, comportant chacune des objectifs bien définis.

5.1 Réalisation des objectifs fixés en introduction

Le premier objectif du projet visait l'acquisition d'une connaissance approfondie de la problématique de recherche afin d'être apte à proposer des solutions pertinentes. Cet objectif a été réalisé en effectuant une revue des travaux existants sur la problématique de la reconnaissance d'activités en général [Bouchard et al, 2007] [Pollack, 2005] [Patterson et al, 2005] [Carberry, 2001] et plus particulièrement les approches exploitant l'aspect temporel [Jakkula et Cook, 2006] [Weida, 1993] [Kautz, 1991]. D'une façon parallèle, une étude des principaux outils formels employés en IA pour modéliser le temps, notamment les théories d'Allen [Allen, 1983] et de Dechter [Dechter, 1991], a aussi été réalisée afin d'être en mesure de comprendre les travaux explorés. Le premier volet a permis d'avoir une vue d'ensemble du domaine de la reconnaissance d'activités, particulièrement dans un contexte applicatif lié à l'assistance des personnes en perte d'autonomie. Il a aussi permis de mieux cerner les enjeux en cause et les besoins spécifiques d'un modèle de reconnaissance destiné à cet usage à l'intérieur d'un habitat intelligent. Le second volet de cette première étape a permis de réaliser un état de l'art ciblé sur les modèles de reconnaissance existants qui exploitent les informations temporelles dans leur processus de reconnaissance. Il a permis d'aboutir à des pistes de solutions qui ont conduit à la contribution proposée dans le mémoire.

Le deuxième objectif du projet consistait à définir les bases théoriques d'un nouveau modèle de reconnaissance d'activités capable de répondre à la problématique temporelle citée en introduction, en se fondant sur les pistes dégagées lors de la revue de la littérature. Pour ce faire, nous avons adopté une approche qui consiste à proposer une extension d'un modèle existant que nous avons exploré, soit celui de Weida [Weida, 1993]. Ce modèle répondait bien à nos besoins applicatifs car il intègre, à la base, une gestion limitée des contraintes temporelles pouvant être étendue et il est compatible avec nos outils applicatifs actuels car il se fonde sur la logique de description [Baader et al., 2007] qui, comme il a été démontré dans [Bouchard, 2006] et [Roy et al., 2009], permet d'exploiter la relation de subsomption pour transformer le processus de reconnaissance en une opération de classification dans une ontologie. Cette phase du projet a permis la définition théorique d'un nouveau modèle de reconnaissance étendue permettant d'identifier les anomalies liées aux délais et aux durées d'exécution d'une activité de la vie quotidienne d'un patient. Il apporte ainsi des réponses à la problématique soulevée.

Enfin, l'objectif final du projet consistait à valider le modèle de reconnaissance temporelle développée au cours de l'étape précédente, de façon à vérifier l'aspect opérationnel de celui-ci. Pour cette étape, une implémentation en Java du modèle original de Weida a été produite puis, une version améliorée a été implémentée en intégrant les propositions théoriques formulées lors du volet précédent. En marge de cette étape, un simulateur d'habitats intelligents virtuels, nommé SIMACT, a été développé par notre équipe [Bouchard et al., 2010]. Il a permis la simulation de scénarios de cas réels avec une visualisation en 3-dimensions du résultat. Ce simulateur a servi de plateforme expérimentale pour tester, avec des données synthétiques tirées d'une étude clinique, les performances du nouveau modèle proposé, notamment au niveau de la précision et de la capacité à reconnaître les erreurs liées à l'aspect temporel. Ces résultats ont ensuite pu être comparés à ceux obtenus avec la version originale du modèle de Weida et ont permis de démontrer l'apport des travaux du mémoire.

Cette dernière partie du travail a fait l'objet d'une publication scientifique importante dans le cadre de la conférence internationale Ubiquitous Computing and Multimedia Applications (UCMA, 2010), dont l'éditeur est la réputée maison Springer-Verlag. Le taux d'acceptation de cette conférence était de 10% (30 articles sélectionnés / 300 soumissions). Cette réalisation vient appuyer les conclusions du mémoire ainsi que l'importance des travaux réalisés par notre équipe et des résultats obtenus.

5.2 Bilan sur le modèle développé

Sommairement, notre approche temporelle de reconnaissance consiste à renforcer un modèle existant celui de Weida [Weida, 1993], qui est basé sur un modèle d'action en logique de description et une approche de représentation temporelle qui est celle de Dechter [Dechter et al, 1991]. L'utilisation du modèle d'action permet de transformer le problème de reconnaissance en un problème de classification basé sur la relation de subsumption. Ce modèle temporel permet d'orienter le processus de reconnaissance d'activités d'un patient atteint de la maladie d'Alzheimer, à la détection d'anomalies issues du temps de réalisation des actions, dans un habitat intelligent.

Le modèle original de Weida proposait un raisonnement et une représentation temporelle exploitant la théorie d'Allen [Allen, 1983], où les contraintes temporelles des plans prédefinis de la librairie et celles des plans observés sont des relations symboliques, décrivant la position d'un événement (action) par rapport à un autre. Cependant, ces relations sont faibles au niveau de l'expressivité quantitative du temps réel (ex. les durées) correspondant à la réalisation d'une

action. Par conséquent, nous avons opté pour une solution à caractère numérique basé sur l'approche de Dechter pour redéfinir les activités prédéfinies de la librairie et de considérer des durées fixes pour les actions des plans observés, étant donné que l'agent observateur fournira préalablement de tels informations. Grâce à cette amélioration, le processus d'inférence temporel est désormais capable de détecter les anomalies provenant des éléments temporels, tel que les durées, en vérifiant la correspondance entre les contraintes temporelles d'un plan de la librairie et une observation tel que leur actions vérifient la relation de subsomption. Les résultats expérimentaux présentés au chapitre 4 ont d'ailleurs illustré cette amélioration lors des essais comparatifs effectués avec l'implémentation du modèle.

5.3 Limitations et perspectives de développement futur

Malgré les résultats prometteurs obtenus à la suite de la vérification lors des différentes phases expérimentales en simulant des scénarios tirés de cas réel, le modèle temporel de reconnaissance proposé dans ce mémoire renferme certaines limitations, lesquelles ont été identifiées lors des différentes étapes de validation. À titre d'exemple, on note que le modèle ne tient pas compte des contraintes temporelles disjonctives où une contrainte représente une disjonction d'intervalles afin d'exprimer l'incertitude au niveau de la durée minimale et maximale d'une action. Bien que cette représentation de contraintes basée sur un seul intervalle, qui est inspiré des Problèmes Temporelles Simples (STP), assoupli la complexité de traitement de la vérification de la subsomption lors du processus de reconnaissance, il reste que cette perspective relève une incapacité à exprimer des approximations sur les intervalles (incertaines ou imprécises). Par exemple, un individu pourrait prendre un taxi pour aller à son travail, lequel va lui prendre de 30 à 40 minutes ou bien prendre l'autobus, qui prendra de 45 à 50 minutes, d'où on peut présenter la contrainte temporelle comme suit : $SortirDeChezSoi [30,40] \cup [45,50]$ *ArriverAuTravail*. Cet exemple a été l'objet d'étude dans le travail de Dechter [Dechter et al, 1991] concernant le problème de satisfaction de contraintes. Dans une perspective de développement futur, on peut penser à étendre le modèle afin d'admettre ce type de disjonctions de façon à ce que l'approche puisse mieux gérer l'incertitude et l'imprécision temporelles au niveau des durées des actions.

Une autre limitation du modèle que nous pouvons dégager est la rigidité des intervalles temporels qui sont définis à l'intérieur de notre modèle. En fait, les intervalles introduits dans la librairie de plans de notre modèle de reconnaissance sont générés de manière statique. Par exemple, pour une durée moyenne d'une action de 4 minutes, on génère un intervalle [3,5], pour

une durée de 40 minutes, on génère un intervalle [30,50], etc. Cette rigidité dans la fixation des bornes inférieures et supérieures des durées acceptables des actions de la librairie constitue une limitation importante. Celle-ci pourrait être partiellement palliée par l'élaboration d'un modèle d'apprentissage capable d'ajuster dynamiquement les intervalles temporels suivant le profil du patient observé qui varie d'un patient à l'autre. Cependant, cette piste, quoiqu'intéressante, est insuffisante pour régler entièrement ce problème. Qu'arrivera t-il lorsqu'une action ayant une durée normale définie dans la librairie entre 3 et 5 minutes se réalise en 2 minutes 59 secondes? Le système considéra alors cette action comme étant échouée car elle n'a pas été réalisée à l'intérieur du laps normal de temps. Pourtant, la différence entre 2 minutes 59 secondes et 3 minutes est négligeable dans ce cas et probablement que l'action a été réalisée correctement. Une piste intéressante à explorer pour répondre à cette problématique dans les développements futurs consiste à investiguer du côté de la logique floue et de la théorie des possibilités, qui sont des voies qui permettront de définir des intervalles de durée d'action « flou » tenant compte de l'imprécision pouvant émaner de la réalisation d'une action par un humain. Par ailleurs, il serait à ce titre intéressant d'introduire la notion de subsomption partielle [Jaeger, 1994] au niveau des éléments temporels. Cette notion a été introduite dans le cadre des travaux sur les logiques de description possibilistes [Heinson, 1994] de façon à caractériser l'appartenance partielle d'une instance ou d'un concept à une classe d'objets. Ainsi, on ne dirait plus simplement que l'action a été réalisé dans un temps correct ou non, mais bien que celle a, par exemple, été réalisé dans un laps de temps répondant partiellement (par exemple à 90%) aux critères de normalité quant à la durée raisonnable escomptée d'une certaine action.

Enfin, on note que le modèle proposé fonctionne bien avec les actions à durée plutôt constante. Dans le cas d'actions à durée variable, comme bouillir de l'eau, qui varie selon le contexte, le modèle pourrait ne pas être adéquat. Une piste de solution à cette problématique pourrait être encore une fois l'utilisation de durée « floue » et une intégration des méthodes d'apprentissages pour ajuster les durées.

5.4 Bilan personnel sur ce travail de recherche

En conclusion finale, je terminerai par un bref bilan personnel de mon expérience d'initiation à la recherche. Le cheminement effectué tout au long du projet a été très enrichissant, tant au niveau de l'acquisition de nouvelles connaissances ciblées dans un domaine d'expertise (le domaine de l'habitat intelligent et de la reconnaissance d'activités), qu'au niveau du développement de nouvelles compétences professionnelles (méthodologie de la recherche,

implémentation, habiletés communicationnelles, etc.). Cette expérience gratifiante m'a de plus permis de contribuer modestement à l'amélioration des connaissances dans mon domaine de recherche. En ce sens, j'aimerais beaucoup poursuivre dans un avenir rapproché mon expérience au troisième cycle et entamer des études doctorales en lien avec mon domaine d'expertise.

ANNEXE A

Introduction aux logiques de description

Cette annexe présente une introduction aux formalismes de représentation de connaissances appelés logiques de description (ou logiques terminologiques) [Baader et al., 2007]. Ce type de formalisme est particulièrement bien adapté pour gérer l’interprétation et la classification des connaissances. Cela peut se traduire concrètement par la conception d’ontologies ou la mise en place de systèmes de représentations de la connaissance terminologique (SRCT).

Dans un premier temps, nous définirons les éléments de base des logiques de description, notamment la notion de concept et de rôle. Ensuite, nous présenterons le langage terminologique TF ainsi que sa sémantique, de manière à illustrer l’interprétation sémantique des différents composants de ce type de logique. En troisième lieu, nous présenterons en détails la relation de subsomption permettant d’organiser les concepts et les rôles par niveau de généralité. Finalement, nous présenterons une brève comparaison du pouvoir expressif de la logique de description par rapport à la logique du premier ordre, avant de conclure l’annexe.

A.1 Éléments de base des logiques de description

Une logique de description (LD) est un formalisme de représentation des connaissances se divisant en deux parties bien distinctes : la partie terminologique (TBox), qui permet de définir un ensemble de concepts ainsi que leurs rôles, et la partie assertionnelle (ABox), qui permet la création d’individus appartenant aux différents concepts et rôles de la partie terminologique [Nebel, 1990]. La partie assertionnelle du formalisme permet les opérations de classification et d’inférence. La classification, qui constitue la base du raisonnement en logique de description, s’appuie sur la relation de subsomption qui permet d’organiser les concepts et les rôles en une taxonomie.

A.1.1 La notion de concept en logique de description

D'un point de vue ensembliste, on peut voir un concept comme étant une série de définitions qui permettent la création d'un ensemble, en spécifiant les conditions nécessaires d'appartenance à celui-ci. Un concept³ en logique de description peut être vu comme une conjonction de caractéristiques qui sont définies par ses rôles. Il existe deux types de concepts :

- **Concept primitif** : les concepts primitifs servent de base à la construction de concepts définis. Ce type de concept représente le niveau le plus atomique de la connaissance en logique de description. Ces concepts peuvent posséder certains rôles qui les définissent, mais leur description est incomplète et représente des conditions nécessaires mais insuffisantes pour déterminer le type d'un individu (une instance) : il n'est pas possible de reconnaître un représentant d'un concept primitif à la seule vue de ses rôles [Haton et al., 1991]. Ainsi, si un concept nommé Personne et un autre nommé Animal sont tous deux définis avec le rôle *date-de-naissance*, il sera vrai de dire que toutes les instances du concept Personne auront une *date-de-naissance*, mais l'inverse sera faux.
- **Concept défini** : les concepts définis sont construits à partir des rôles qu'ils entretiennent avec des concepts primitifs ou avec d'autres concepts définis. Ce type de concept représente les niveaux de connaissances plus complexes. Contrairement aux concepts primitifs, les concepts définis possèdent une définition complète et suffisante pour en déduire l'appartenance certaine d'une instance [Napoli, 1997]. Si l'on prend, par exemple, un concept nommé Parent qui entretient un rôle nommé enfant, il sera possible d'affirmer qu'une instance de type Parent aura automatiquement un ou des enfant et qu'une instance ayant un ou des enfant, appartient automatiquement au concept Parent.

A.1.2 La notion de rôle

Un rôle représente une relation binaire entre deux concepts. Cette relation a pour fonction de caractériser le premier concept en précisant sa relation avec le second. Le concept non caractérisé par le rôle est appelé le co-domaine. Des restrictions de cardinalité peuvent être attribuées aux rôles. Par exemple, le rôle enfant, caractérisant le concept Parent, devrait être défini avec une cardinalité $n \geq 1$. Donc, les rôles servent à caractériser les concepts et permettent de spécifier leurs descriptions grâce aux relations qu'ils entretiennent les uns avec les autres.

³ Habituellement, chaque terme désignant le nom d'un concept débute par une lettre majuscule et chaque terme désignant le nom d'un rôle commence par une lettre minuscule.

A.1.3 Formalisme terminologique et assertionnel

Une logique de description peut être vue comme la conjonction d'un langage terminologique et d'un langage assertionnel.

Un langage terminologique consiste en un ensemble d'opérateurs (appelés des constructeurs) et de symboles qui permettent la définition de concepts et de rôles. Par exemple, le langage terminologique TF, que nous utiliserons pour illustrer la théorie, peut être défini comme suit [Nebel, 1990] :

$$TF = \langle \top, \perp, =, \prec, \text{and}, \text{all}, \text{atleast}, \text{atmost} \rangle$$

où $\{ \top, \perp, =, \prec \}$ sont des symboles et $\{ \text{and}, \text{all}, \text{atleast}, \text{atmost} \}$ sont des opérateurs. La sémantique de ce langage sera décrite dans les sections suivantes.

Un langage assertionnel peut être défini comme étant un ensemble de symboles et d'opérateurs permettant la création d'instances issues de concepts et de rôles décrits par un langage terminologique. Par exemple, le langage terminologique AF est le complément assertionnel de TF, et se définit comme suit :

$$AF = \langle c, r, \geq, \leq, \text{atleast}, \text{atmost} \rangle$$

où $\{\geq, \leq\}$ sont des symboles et $\{c, r, \text{atleast}, \text{atmost}\}$ sont des opérateurs. Il est à noter que c et r représentent ici deux termes désignant respectivement un concept c et un rôle r .

A.2 La sémantique des logiques de description

À l'instar de la logique classique, une sémantique est associée aux descriptions de concepts et de rôle dans les logiques de description. Les concepts sont interprétés comme des sous-ensembles d'un domaine d'interprétation et les rôles comme des sous-ensembles d'un produit entre deux domaines d'interprétations [Baader et al., 2007].

A.2.1 La notion d'interprétation

La notion d'interprétation, symbolisée par I , est surtout décrite dans les articles de Baader et Hollunder [Baader et Hollunder, 1991] [Baader et al., 1991] et de Falquet [Falquet et Mottaz, 2000]. Pour la définir, il faut se donner un domaine sémantique Δ^I , qui correspond à l'ensemble de tous les individus de la terminologie étudiée (pas seulement ceux qui sont effectivement

existants), et une fonction \cdot^I , dont le rôle est d'associer chacun des termes désignant un concept à un sous-ensemble de Δ^I et chaque nom de rôle à un sous-ensemble de $\Delta^I \times \Delta^I$. La fonction \cdot^I est appelée fonction d'interprétation et elle s'utilise en respectant la terminologie introduite par le langage terminologique sous-jacent. Prenons, par exemple, les concepts suivants : {*Chaise*, *Dossier*, *Pied*} et le rôle partie qui relie une composante d'un meuble à celui-ci. Le domaine sémantique Δ^I sera alors constitué de l'ensemble de tous les individus issus des concepts *Chaise*, *Dossier* et *Pied* ainsi que toutes les parties possibles (respectivement c_1, d_1, p_1 pour représenter les concepts et des couples de ceux-ci pour représenter le rôle partie, en respectant sa définition). Donc, en appliquant une fonction d'interprétation, on aurait, par exemple :

$$\begin{aligned} \text{Chaise}^I &= \{ c_1, c_2, c_3, c_4 \} \\ \text{Dossier}^I &= \{ d_1, d_2, d_3 \} \\ \text{Pied}^I &= \{ p_1, p_2, p_3, p_4 \} \\ \text{partie}^I &= \{ (c_1, d_1), (c_1, p_1), (c_1, p_2), (c_1, p_3), (c_1, p_4), (c_3, d_2) \} \end{aligned}$$

La notion d'interprétation signifie littéralement que l'on interprète, via la fonction d'interprétation \cdot^I , les concepts et les rôles donnés par la terminologie. Il peut donc y avoir plusieurs interprétations I possibles pour une terminologie donnée.

A.2.2 Illustration ensembliste du langage TF

Il est possible de réécrire tous les opérateurs d'un langage terminologique avec un symbolisme ensembliste. C'est ce qu'on appelle la sémantique d'un langage terminologique. On distingue souvent une terminologie de son interprétation en utilisant les termes définition par intention et définition par extension pour l'interprétation. Voici, à titre d'exemple, la sémantique du langage TF, sous le format « constructeur = sémantique » :

$$\begin{aligned} T^I &= \Delta^I \\ \perp^I &= \emptyset^I \\ (C \text{ and } D)^I &= C^I \cap \Delta^I \\ (\text{all } R \text{ } C)^I &= \{ i \in \Delta^I \mid \forall j : (i, j) \in r^I \rightarrow j \in C^I \} \\ (\text{atleast } n \text{ } R)^I &= \{ i \in \Delta^I \mid \#(j \in \Delta^I : (i, j) \in r^I) \geq n \} \quad (\text{atmost } n \text{ } R)^I = \{ i \in \Delta^I \mid \#(j \in \Delta^I : (i, j) \in r^I) \leq n \} \end{aligned}$$

Le symbole C et D désignent ici deux concepts, R un rôle et # désigne la cardinalité d'un ensemble d'éléments. Ainsi, la définition d'un concept D comme étant un sous-concept de C,

c'est-à-dire $D \prec C$, s'écrit sémantiquement avec le symbole d'inclusion des ensembles $D \subseteq C$. Quant à la définition d'un concept C à l'aide du symbole $=$, elle est équivalente à la création d'un nouvel ensemble $C \subseteq \Delta^1$. Ceci s'applique également aux rôles, mais en sachant qu'un rôle est une fonction dans $\Delta^1 \times \Delta^1$. À partir de là, il est tout à fait possible de présenter graphiquement, d'une façon ensembliste, la sémantique d'un langage terminologique, puisqu'elle est uniquement composée d'opérations sur des ensembles. Il suffit d'appliquer les opérations ensemblistes : l'intersection pour le and, l'union pour le or (non disponible en TF, mais il se retrouve dans plusieurs autres langages terminologiques), etc. Cette équivalence ensembliste s'applique également à la relation de subsomption. Par conséquent, une relation de subsomption entre un concept D et un concept C peut être présentée comme l'inclusion d'un ensemble D dans un ensemble C ou vice-versa.

A.3 La relation de subsomption en LD

La relation de subsomption permet d'organiser les concepts et les rôles par niveaux de généralité. Intuitivement, un concept C subsume un concept D si C (le subsumant) est plus général que D (le subsumé). Cette réalité peut être exprimée de trois différentes façons :

Définition ensembliste en extension : un concept C subsume un concept D si et seulement si l'ensemble des représentants dénoté par C contient nécessairement l'ensemble des représentants dénoté par D [Levesque et Brachman, 1987]. Dans cette définition, chaque concept est représenté par un ensemble et chaque relation de subsomption est représentée par une inclusion ensembliste. Par exemple, le concept qui dénote les mammifères subsume celui dénotant la famille des chats.

Définition ensembliste en intention : un concept C subsume un concept D si toute entité représentée par D l'est aussi par C . Ainsi, chaque entité dont la description est définie par D possède les caractéristiques qui sont spécifiées par C [Finin, 1986]. Par exemple, le concept dénotant les chats possède toutes les caractéristiques de celui qui dénote les mammifères en plus de celles qui sont propres aux chats. Un concept se compose donc d'une description propre à lui-même et d'une description partagée ou commune avec ses subsumants.

Définition logique : un concept C subsume un concept D si un élément qui est décrit par D implique logiquement que cet élément est décrit par C . Ainsi, l'implication $(\text{Chat}(x) \rightarrow \text{Mammifère}(x))$ est vraie pour tout x [McGregor, 1988].

En utilisant la notion d'interprétation, on peut définir la relation de subsomption de la manière suivante : un concept D est subsumé par un concept C (respectivement C subsume D), ce qui se note $D \prec C$, si et seulement si $D^1 \subseteq C^1$ pour toute interprétation ¹ de C et D . La subsomption est à la base de la classification en logique de description. C'est une relation d'ordre partiel qui permet d'organiser les concepts en une taxonomie. Chaque concept possède une description qui permet de le comparer aux autres et de déterminer les relations de subsomption qu'il entretient avec les autres concepts. La subsomption est une relation réflexive ; un concept C est donc subsumé par lui-même. Cette relation est également transitive, ce qui signifie que si un concept C subsume un concept D et que ce concept D subsume un concept E , alors le concept C subsume automatiquement le concept E . La relation de subsomption est aussi antisymétrique, ce qui signifie que si un concept C subsume un concept D et que le concept D subsume à son tour le concept C , alors $C = D$. Les différentes relations de subsomption qu'entretient un concept permettront de déduire l'emplacement que ce concept devrait occuper dans la taxonomie. Un concept maximal, noté \top , représentant la totalité du domaine d'interprétation, est présent au sommet de la taxonomie. Ce concept représente l'ensemble le plus général et subsume tous les autres, permettant ainsi d'obtenir un point de départ pour la taxonomie. Inversement, un concept minimal, noté \perp , est présent au bas de la hiérarchie. Ce concept est subsumé par tous les autres ; il représente en fait le vide (aucune instance du domaine).

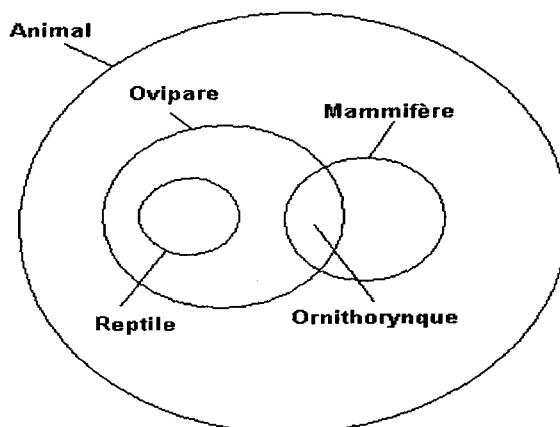


Figure A.1 – Représentation ensembliste d'une série de concepts

Prenons, par exemple, les concepts *Animal*, *Mammifère*, *Ovipare*, *Reptile* et *Ornithorynque*, tels que représentés de façon ensembliste sur la Figure A.1. On peut voir que le concept *Ovipare* subsume le concept *Reptile*, car le fait d'être un *Reptile* implique le fait d'être un *Ovipare*. On peut également noter, sur la même figure, que le concept *Ornithorynque* est subsumé par le concept *Mammifère* et par le concept *Ovipare*, mais de façon plus explicite, par la conjonction des deux concepts. Le fait d'être un *Ornithorynque* implique obligatoirement le fait d'être un *Mammifère* et un *Ovipare*. Chacun des concepts est donc délimité par l'ensemble des individus qu'il représente par rapport aux autres concepts. On peut donc en conclure que la classification d'un nouveau concept se réduit à déterminer où devra se situer l'ensemble qu'il représente par rapport aux autres ensembles déjà existants.

A.3.1 Subsomption versus Héritage

Il est très important de faire la distinction entre une relation d'héritage (telle que l'on connaît dans les langages objets) et une relation de subsomption. À première vue, les deux relations peuvent sembler identiques, mais même si elles s'apparentent l'une à l'autre, elles ne le sont pas. En vérité, on peut dire que la relation de subsomption subsume ou est plus générale que la relation d'héritage, car elle porte sur les concepts, sur les individus, mais aussi, de façon duale, sur des clauses : les concepts s'apparentent à des conjonctions de littéraux de la logique du premier ordre [Hamberger et Richards, 2002] alors que les clauses sont des disjonctions de littéraux [Borgida, 1996]. La relation de subsomption relève donc d'un niveau d'abstraction plus élevé que la relation d'héritage. On peut voir l'héritage comme un mécanisme de partage de la connaissance qui peut être considéré comme une façon d'implanter la subsomption [Haton et al., 1991]. De plus, l'héritage est une relation tout ou rien, ce qui n'est pas le cas de la subsomption.

A.3.2 Inférence des relations de subsomption

Afin de pouvoir identifier une relation de subsomption entre deux concepts, il faut les comparer « composante par composante ». Il faut se rappeler qu'un concept en logique de description peut être vu comme une conjonction de caractéristiques définies par des rôles et que des restrictions de cardinalités peuvent être attribuées à ces caractéristiques. Autrement dit, pour vérifier si un concept *C* subsume un concept *D*, il faut s'assurer que pour chaque rôle du concept *D*, il existe un rôle équivalent ou plus général défini par le concept *C*. Afin de mieux comprendre la procédure de détection, voici une version simplifiée de l'algorithme utilisé dans KL-ONE

[Schmolze et Lipkis, 1983], un des premiers langages de description proposé par Schmolze et Lipkis, pour déterminer si un concept C subsume un concept D :

```

Si tous les concepts qui sont subsumants de C sont aussi subsumants de D
Alors retourner vrai
Pour tous les rôles  $c_i$  de C
    Pour tous les rôles  $d_j$  de D
        Si l'un des rôles  $d_j$  de D exprime le même rôle que  $c_i$  de C (si
             $c_i = d_j$ ) OU
        Si l'extension du rôle  $c_i$  est incluse par l'extension du rôle
             $d_j$  (si  $c_i$  subsume  $d_j$ )
        Alors passer à ( $c_i + 1$ )
        Sinon retourner échec ou inconnu
    Retourner vrai

```

Il faut maintenant préciser comment il est possible de déterminer si un rôle c_i subsume un rôle d_j . Pour cela, il faut répondre aux deux questions suivantes :

- Est-ce que le co-domaine du rôle c_i subsume le co-domaine du rôle d_j ?
- Est-ce que la cardinalité du rôle c_i est égale ou supérieure à celle du rôle d_j ?

L'algorithme de vérification d'une relation de subsomption présenté ci-haut peut retourner deux réponses : vrai ou inconnu. Dans le cas où l'algorithme retourne vrai, cela signifie que le concept C subsume avec certitude le concept D . Cependant, lorsque l'algorithme retourne inconnu, cela ne signifie pas nécessairement que le concept C ne subsume pas le concept D , mais plutôt qu'avec les informations connues sur les deux concepts, il est impossible de le prouver. Le problème de la détection des relations de subsomption est d'une importance capitale pour la classification dans une hiérarchie de concepts.

A.4 Comparaison de la LD avec la logique du premier ordre (LPO)

La logique de description et la logique du premier ordre sont très proches l'une de l'autre. Par ailleurs, les concepts que l'on retrouve en logique de description peuvent être vus comme des prédictats unaires, et les rôles comme des prédictats binaires [Haton, 1991]. Il est toujours possible d'effectuer la conversion d'une réalité modélisée avec une logique de description en LPO, mais l'inverse n'est pas toujours vrai. On peut donc en conclure que la LPO subsume la logique de description ou encore que les logiques de description un fragment de la logique du premier ordre. Cependant, la LPO est une logique mathématique difficilement utilisable de façon concrète pour

la classification, contrairement à la logique de description. Une comparaison entre le pouvoir d'expression des logiques de description et celui de la logique du premier ordre est donnée par Baader et Borgida [Borgida, 1996] [Baader, 1999]. Leurs comparaisons établissent que tous les logiques de description proposés jusqu'à présent peuvent exprimer toutes les notions exprimables dans la LPO avec trois variables au plus, et seulement celles-là [Borgida, 1996].

A.5 Conclusion

Cette annexe a permis d'introduire les notions fondamentales inhérentes aux logiques de description. Cette introduction a été réalisée grâce à une définition des éléments de base constituant ce type de logique, suivie d'une présentation d'un langage terminologique simple et de sa sémantique, pour finalement terminer avec comparaison du pouvoir expressif de la LD versus celui de la LPO. La plupart des explications ont été appuyées par des exemples concrets, dans le but de comprendre plus facilement la théorie. Les exemples ont notamment été illustrés à l'aide du formalisme terminologique TF, défini par Nebel [Nebel, 1990].

Les logiques de description possèdent plusieurs avantages en tant que formalisme de représentation de la connaissance. Elles sont reconnues pour être particulièrement efficaces pour le raisonnement par classification. Ce type de logique possède une sémantique bien définie. Ceci implique que l'existence des instances de concepts n'est pas donnée de façon opérationnelle ou par l'implémentation elle-même. Ce serait l'inverse si on affirmait que l'assertion « John est un père » était valide uniquement parce que le système a répondu « père » à la question « Qu'est-ce que John ». L'existence des instances est plutôt donnée par des modèles et des descriptions qui permettent aux logiques de description de s'articuler dans un univers bien défini, et non d'être une simple suite de définitions statiques de références. Il faut également souligner que les logiques de description constituent une famille de formalismes qui se veulent plus simple et intuitive d'utilisation que la logique du premier ordre, entre autres parce qu'il n'y a pas de quantificateur. De plus, il a été prouvé que les logiques de description sont complètes et correctes [Baader et al., 1991], ce qui permet de les utiliser dans plusieurs domaines connexes à l'intelligence artificielle. Par exemple, les travaux de Schmidt [Schmidt, 2000] ont permis de voir les logiques de description comme étant une solution possible pour la compréhension du langage naturel. On peut également voir les applications des logiques de description dans le domaine des systèmes multi-agents, pour tenter de résoudre le problème de l'hétérogénéité sémantique [Valentia, 2000] concernant l'utilisation de termes différents pour décrire une même réalité. Les

logiques de description ont aussi été proposées comme support théorique pour la modélisation de bases de données [Borgida, 1996].

Malgré tous les avantages découlant de l'utilisation des logiques de description, leur principal problème est leur dualité d'expressivité versus leur complexité [Nebel, 1990]. Autrement dit, plus une formalisme de description permet de modéliser un grand nombre de réalités, plus sa complexité théorique augmente au niveau des opérations d'inférences, en particulier pour la détection des relations de subsomption [Schmidt, 1991].

Finalement, cette annexe constitue uniquement une introduction à la théorie concernant les logiques de description. Il faut savoir qu'il existe plusieurs points qui n'ont pas été présentés en profondeur, notamment les mécanismes pour supporter les définitions cycliques et récursives, les différents problèmes reliés à la complexité des SRCT, la notion de modèle à l'intérieur d'un SRCT, le fonctionnement de plusieurs services dans un SRCT, etc. Cependant, cette annexe se veut une introduction suffisamment complète pour permettre au lecteur de comprendre l'ensemble des éléments discutés dans ce mémoire. Pour une présentation beaucoup plus détaillée de la théorie concernant les logiques de description, le lecteur est prié de consulter l'ouvrage de référence écrit par Baader et al. [Baader et al., 2007] et intitulé « *The Description Logic Handbook: Theory, Implementation, and Applications* ».

BIBLIOGRAPHIE

- [Allen et Ferguson, 1994] James F. Allen, & George Ferguson (1994). *Actions and Events in Interval Temporal Logic*. Technical Report 521.
- [Allen, 1981] J. F. Allen : *An interval-based representation of temporal knowledge*. Proceedings of the 7th IJCAI, pp. 221-226, 1981.
- [Allen, 1983] James Allen. *Maintaining knowledge about temporal intervals*. Communications of the ACM 26(11).832-843, 1983.
- [Aspvall et Shiloach, 1980] B. Aspvall and Y. Shiloach, *A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality*, SIAM J. Comput. 9 (4). 827-845 ,1980.
- [Baader et al., 1991] F. Baader, H.J. Bürckert, J. Heinsohn, B. Hollunder, J. Müller, B. Nebel, W. Nutt, et H.J. Profitlich H.J. *Terminological Knowledge Representation : A proposal for Terminological Logic*. Technical Report TM-90-04, DFKI (Deutshes Forshungszen- trum für Künstliche Intelligenz), Allemagne, 1991.
- [Baader et Hollunder, 1991] F. Baader et B. Hollunder. *A Terminological Knowledge Representation System with Complete Inference Algorithms*. In Proc. of the first International Workshop on Processing Declarative Knowledge (PDK-91), Lecture Notes in Artificial Intelligence, Springer-Verlag, volume 567, 67–86, 1991.
- [Baader, 1999] F. Baader. *Artificial Intelligence Today : Recent Trend and Developments, chapter Logic Based Knowledge Representation*. Lecture Notes in Computer Science, Springer-Verlag, M.J. wooldridge et M. veloso editors , 13–41, 1999.

[Baader et Nutt, 2003] Baader, F., and Nutt, W. Basic description logics. In *The Description Logic Handbook*, F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. Eds. Cambridge University Press, 2003, pp. 43-95.

[Baader, 2005] Baader, F., Lutz, C, Milicic, M., Sattler, U., and Wolter, F. Integrating description logics and action formalisms : First results. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*. 2005, pp. 572-577.

[Baader et al., 2007] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), *The Description Logic Handbook: Theory, Implementation, and Applications*, 2nd Edition, Cambridge University Press, 2007.

[Basseville, 1996] M. Basseville, M-O. Cordier : *Surveillance et diagnostic de systèmes dynamiques : approches complémentaires du traitement de signal et de l'intelligence artificielle*, Technical Report 1004, IRISA, Mars 1996.

[Bassière et Regin, 1997] Bessiere, C. and J. C. Regin. *Arc consistency for general constraint net-works: preliminary results*. International Joint Conference on Artificial Ingelligence (IJCAI'97), Nagoya, Japan, 1997.

[Björn et al., 2006] Björn Gottfried, Hans W. Guesgen, & Sebastian Hübner. *Spatiotemporal Reasoning for Smart Homes*. Designing Smart Homes. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Pg 16-34, Volume 4008/2006.

[Borgida, 1996] Borgida, A. *On the relative expressiveness of description logics and predicate logics*. *Artificial Intelligence* 82 353-367, 1996.

[Bouchard et al., 2007] Bouchard, B., Bouzouane, A., and Giroux, S. *A keyhole plan recognition model for Alzheimer's patients : First results*. *Applied Artificial Intelligence* 22, 7. August 2007, 623-658.

[Bouchard, 2006] Bouchard, B. *Un modèle de reconnaissance de plans pour les personnes atteintes de la maladie d'Alzheimer basé sur la théorie des treillis et sur un modèle d'action en logique de description*. PhD thesis, Université de Sherbrooke, 2006.

[Bouchard et al., 2010] Bouchard K., Ajroud A., Bouchard B., Bouzouane A.: *SIMACT: a 3D Open Source Smart Home Simulator for Activity Recognition*, Proc. of International Conference

on Ubiquitous Computing and Multimedia Application (UCMA'10), Springer, Lecture Notes in Computer Science (LNCS), June 23-25, Miyazaki, Japan, pp. 524-533, 2010 (taux d'acceptation de seulement 10%).

[Bouzouane, 2005] Bouzouane A.: *Action description logic for smart home agent recognition*. In Proceedings of the International conference on Human- Computer Interaction (HCI) of the International Association of Science and Technology for Development (IASTED), pp. 185-190, 2005.

[Carberry, 2001] Carberry, S. Techniques for plan recognition. *User Modeling and User-Adapted Interaction 11*, 2001, 31-48.

[Davey et al, 2002] Davey, B. A., and Priestley, H. A. *Introduction to Lattice and Order*, second ed. Cambridge University Press, 2002.

[Dean et McDermott, 1987] T. L. Dean and D. V. McDermott. *Temporal Data Base Management*. AI, 32 :1-55, 1987.

[Dechter et al., 1991] R. Dechter, I. Meiri, J. Pearl : *Temporal constraint networks*, Artificial Intelligence n°49, 1991, pp. 61-95.

[Dousson, 1994] C. Dousson : *Suivi d'évolutions et reconnaissances de chroniques*, Thèse du LAAS, 1994.

[Eude, 1999] Virginie. Eude : *Modélisation spatio-temporelle floue pour la reconnaissance d'activités militaires*, Thèse de doctorat de l'université Paris 6, 1999.

[Fontaine, 1993] D. Fontaine : *Reconnaissance de scénarios temporels*, Rapport interne U.R.A C.N.R.S 817 Heudiasyc, UT Compiègne, Juin 1993.

[Fikes et al., 1971] Fikes, R. E., and Nilsson, N. J. *Strips : A new approach to the application of theorem proving to problem solving*. *Artificial Intelligence 2*, 189-208, 1971.

[Finin, 1986] T. W. Finin. *Interactive Classification : A technique for acquiring and maintaining Knowledge Bases*. Proc. IEEE 74, 1414-1421, 1986.

[Garlatti, 2002] Serge, Garlatti. *Les logiques non-standards*. Enst-Bretagne, France, 2002.

[Ghallab, 1989] M. Ghallab. *Représentation et gestion de relations temporelles*, Actes du 7ème congrès RFIA, AFCET, Paris, pp. 3-20, 1989.

[Ghallab, 1992] M. Ghallab : *Managing temporal knowledge : a survey on problems and approaches*, International Symposium on Artificial Intelligence, Robotics and Automation in Space, Toulouse, 1992.

[Gopalratnam et Cook, 2003] Gopalratnam K. and Cook D., Active LeZi: *An Incremental Parsing Algorithm for Sequential Prediction*, Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference, 2003, pp 38-42.

[Gopalratnam et Cook, 2004] K. Gopalratnam & D. J. Cook (2004). *Active LeZi: An Incremental Parsing Algorithm for Prediction*. International Journal of Intelligence Tools. 14(1-2):917-930, 2004.

[Hamburger et Richards, 2002] H. Hamburger et D. Richards. *Logic and Language Models for Computer Science*. Upper Saddle River, 2002.

[Hatou, 1991] Hatou : *Le raisonnement en Intelligence Artificielle*, inter-éditions, Paris, 1991.

[Heinsohn, 1994] J. Heinsohn, *Probabilistic description logics*, in Proc. of the 10th Annual Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufmann, 1994, pp. 311-318.

[Jaeger, 1994] Jaeger, M. *Probabilistic reasoning in terminological logics*. In Principles of Knowledge Representation and Reasoning : Proceedings of the Fourth International Conference, J. Doyle, E. Sandewall, and P. Torasso, Eds., Morgan Kaufmann Publishers, 1994.

[Jakkula et Cook, 2007] Vikramaditya Jakkula, & Diane J. Cook.. *Learning temporal relations in smart home data*. Proceedings of the second International Conference on Technology and Aging. Canada.(To appear), 2007.

[Khachian, 1979] L.G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl. 20, 191-194, 1979.

[Kautz, 1991] Kautz, H. *A Formal Theory of Plan Recognition and its Implementation*, Reasoning about plans, 1991, pp.69-126.

[Kautz et Allen, 1986] Kautz, H., and Allen, J. *Generalized plan recognition*. In Proceedings of the Fifth National Conference on Artificial Intelligence, AAAI, Ed., pp. 32-37, 1986.

[Kautz et Ladkin, 1991] H. A. Kautz, P. B. Ladkin : *Integrating metric and qualitative temporal reasoning*, AAAI, pp. 241-246, 1991.

[Kemke, 2003] Kemke, C. *A formal approach to describing action concepts in taxonomical knowledge bases*. In Foundations of Intelligent Systems, N. Zhong, Z. W. Ras, S. Tsumoto, and E. Suzuki, Eds., Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, pp. 657-662.

[Koller et al., 1997] Koller, D., Levy, A., and Pfeffer, A. *P-CLASSIC : A tractable probabilistic description logic*. In Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI, pp. 390-397, 1997.

[Ladkin et Maddux, 1989] P.B. Ladkin and R.D. Maddux, *On binary constraint networks*, Tech. Rept., Kestrel Institute, Palo Alto, CA, 1989.

[Laprise et al., 2010] Laprise H., Bouchard J., Bouchard B., Bouzouane A.: *Creating tools and trial data sets for smart home researchers: experimenting activities of daily living with normal subjects to compare with Alzheimer's patients*, Proc. of International Conference IADIS e-Health 2010 (EH 2010), Freiburg, Germany, 29 - 31 July, pp. 1-8, 2010.

[Leiserson et Saxe, 1983] C.E. Leiserson and J.B. Saxe, *A mixed-integer linear programming problem which is efficiently solvable*, in : *Proceedings 21st Annual Allerton Conference on Communications, Control, and Computing*, 204-213, 1983.

[Levesque et Brachman, 1987] H. Levesque et R.J. Brachman. *Expressiveness an Tractability. Knowledge Representation and Reasoning* , Computational Intelligence 3, 78-93, 1987.

[Mackworth, 1977] A. K. Mackworth. *Consistency in networks of relations*. *Artificial Intelligence*, 8(1) :99-118, 1977.

[Malik et Binford, 1983] J. Malik and T.O. Binford, *Reasoning in time and space*, in : *Proceedings IJCAI-83*, Karlsruhe, FRG, 1983, 343-345.

[McCarthy, 1980] J. McCarthy. *Circumscription - A Form of Non-Monotonic Reasoning*. *Artificial Intelligence* 13, 27-39, 1980.

[McDermott, 1982] D. McDermott: *A temporal logic for reasoning about processes and plans*, Cognitive Science, vol. 6 n°2, 1982, pp. 101-155.

[Mays et al., 1991] E. Mays, R. Dionne, and R. Weida. *K-rep system overview*. Sigart Bulletin, Special Issue on Implemented Knowledge Representation and Reasoning Systems, 2(3) : 93-97, 1991.

[McGregor, 1988] R. McGregor. *A deductive Pattern Matcher*. In Proc. of the 7th AAAI, Saint-Paul, Minnesota, 403–408, 1988.

[Meiri, 1991] I. Meiri: *Combining qualitative and quantitative constraints in temporal reasoning*, Proceedings of the AAAI, Los Angeles, pp. 260-267, 1991.

[Mihailidis et al., 2007] A. Mihailidis, J. Boger, M. Canido, and J. Hoey, *The use of an intelligent prompting system for people with dementia: A case study*, ACM Interactions, vol. 14, no. 4, pp. 34-37, 2007.

[Mörchen, 2006] Mörchen, F. *A better tool than Allen's relations for expressing temporal knowledge in interval data*. In Proceedings the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Philadelphia, PA, USA, 2006.

[Nebel, 1990] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*. Lecture Notes in Artificial Intelligence, Springer-Verlag 422, 1–270, 1990.

[Napoli, 1997] A. Napoli. *Une introduction aux logiques de descriptions*. Technical report, Technical Report N.3314, Projet SYCO, INRIA Lorraine, 1997.

[Pigot, 2003] Pigot H., Mayers A, Giroux S. *The intelligent habitat and everyday life activity support*, in 5th International Conference on Simulations in Biomedicine, 2003.

[Patterson et al., 2003] D.J. Patterson, L. Liao, D. Fox, et H. Kautz. *Inferring High-Level Behavior from Low-Level Sensors*. In Proceeding of UBICOMP 2003 : The 5th International Conference on Ubiquitous Computing, Anind Dey, Albrecht Schmidt, et Joseph F. McCarthy, editors, volume LNCS 2864, 73–89. Springer-Verlag, 2003.

PowerLoom [PowerLoom, 2002]. *Powerloom Knowledge Representation System Manual*, ISI, University of Southern California, 2002.

- [Pollack, 2005] Pollack, M.. *Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment*. AI Magazine, 26(2):9-24, 2005.
- [Ramos et al., 2008] Ramos C., Augusto J. C. and D. Shapiro, *Ambient Intelligence: the Next Step for Artificial Intelligence*, IEEE Intelligent Systems, vol. 23, pp. 5-18, 2008.
- [Rocher et al., 2010] Rocher P-O, Bouchard B., Bouzouane A : *A New Platform to Easily Experiment Activity Recognition Systems Based on Passive RFID Tags: Experimentation with Data Mining Algorithms*, International Journal of Smart Home (IJSH), Science and Engineering Support Society (SERSC), (en révision), pp. 1-18, 2011.
- [Roy et al., 2007] Roy P., Bouchard B., Bouzouane A., Giroux S. : *A hybrid plan recognition model for Alzheimer's patients : interleaved-erroneous dilemma*. In The 2007 IEEE / WIC / ACM International Conference on Intelligent Agent Technology (IAT 2007), pp. 131-137, 2007.
- [Roy et al., 2010] Roy P., Bouchard B., Bouzouane A., Giroux S.: *Challenging issues of ambient activity recognition for cognitive assistance*. IGI global, F. Mastrogiovanni and N. Chong Editors, Handbook of research on Ambient Intelligence and Smart Environments: Trends and Perspectives, pp. 1-25, (to appear), 2010.
- [Russell et al., 2003] Russell, S., and Norvig, P. *Artificial Intelligence: A Modern Approach*, second ed. Prentice Hall, 2003.
- [Singla et al., 2008] Singla G., Cook D., Schmitter-Edgecombe M: *Incorporating Temporal Reasoning into Activity Recognition for Smart Home Residents*, In proceeding of international conference AAAI'08, Chicago, Illinois, July 13-17, 2008.
- [Schmidt et al., 1978] Schmidt, C F. , Sridharan, N. S., and Goodson, J. L. *The plan recognition problem: An intersection of psychology and artificial intelligence*. Artificial Intelligence 11 1978, 45-83.
- [Schmidt, 2000] R. Schmidt. *Relational Grammars for Knowledge Representation*. Variable-Free Semantics, Artikulation und Sprache, Secolo Verlag 3, 162–180, 2000.
- [Schmolze et Lipkis, 1983] J.G. Schmolze et T.A. Lipkis. *Classification in the KL-ONE Knowledge Representation System*. In Proc. of the 8th IJCAI, Karlsruhe, 330–332, 1983.

- [Schmidt, 1991] R. Schmidt. *Algebraic Terminological Representation. Technical report*, Rapport MPI-I-91-216 de MPI (Max-Planck-Institut fur informatik), Allemagne, 1991.
- [Song et Cohen, 1991] F. Song, R. Cohen : *Temporal reasoning during plan Recognition*, Proc. of the AAAI, pp.247-252, 1991.
- [Tolba et Masini, 1990] H. Tolba and G. Masini. *Temporal Constraint Propagation*. Technical report, CRIN 90-R-177, CRIN-INRIA, 1990.
- [United Nation, 2002] United Nations, *World population ageing: 1950-2050*, 2002.
- [Valdés, 1986] R.E. Valdés-Pérez, *Spatio-Temporal reasoning and linear inequalities*, Artificial Intelligence Laboratory, AIM-875, MIT, Cambridge, MA, 1986.
- [Valentia, 2000] E. Valentia. *Outils de topologie algébrique pour la gestion de l'hétérogénéité sémantique entre agents dialogiques*. Thèse de Doctorat, Université paris-XI, France, 2000.
- [VanBeek, 1989] P. Van Beek, *Approximation algorithms for temporal reasoning*, in : Proceedings IJCAI-89, Detroit, MI 1291-1296, 1989.
- [VanBeek, 1992] P. Van Beek : *Reasoning about qualitative temporal information*, Artificial Intelligence n°58, 1992, pp. 297-326.
- [Vilain et al., 1986] M. Vilain and H. Kautz. *Constraint Propagation Algorithms for Temporal Reasoning*. In AAAI, pages 377-382, 1986.
- [Vilain et al., 1989] M. Vilain, H. Kautz, and P. Van Beek. *Constraint propagation algorithms for temporal reasoning : a revised report*. In J. deKleer and D. Weld, editors, *Readings in Qualitative Reasoning About Physical Systems*. Morgan Kaufmann, LosAltos, CA, 1989.
- [Weida, 1993] Weida R., *Terminological Constraint Network Reasoning and its Application to plan Recognition*, Thèse de doctorat, University of Columbia, 1993, pages 9- 34.
- [Weiss, 2000] Weiss G., *Multiagent systems*, MIT Press, 2000.
- [Written, 2005] Written H., Frank E, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, Second Edition, 2005.

[Youngblood et al., 2005] G. Michael Youngblood, Lawrence B. Holder, & Diane J.Cook. *Managing Adaptive Versatile Environments*. Proceedings of the IEEE International Conference on Pervasive Computing and Communications, 2005.