



HAL
open science

Gestion d'intentions multiples pour agents ambiants coopératifs

Arthur Casals, Assia Belbachir, Amal El Fallah-Seghrouchni, Anarosa Alves
Franco Brandão

► **To cite this version:**

Arthur Casals, Assia Belbachir, Amal El Fallah-Seghrouchni, Anarosa Alves Franco Brandão. Gestion d'intentions multiples pour agents ambiants coopératifs. Journées Francophones sur les Systèmes Multi-Agents, Oct 2018, Métabief, France. hal-02893406

HAL Id: hal-02893406

<https://hal.sorbonne-universite.fr/hal-02893406>

Submitted on 14 Sep 2020

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

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

Gestion d'intentions multiples pour agents ambiants coopératifs

Arthur Casals^{a,b}
arthur.casals@lip6.fr

Assia Belbachir^{a,c}
assia.belbachir@ipsa.fr

Amal El-Fallah Seghrouchni^a
amal.elfallah@lip6.fr

Anarosa Alves Franco Brandão^b
anarosa.brandao@usp.br

^aSorbonne Universités, UPMC Université Paris 06, CNRS, LIP6, Paris - France

^bEscola Politécnica - USP, São Paulo, Brazil

^cIPSA (Institut Polytechnique des Sciences Avancées), Ivry-sur-Seine, France

Résumé

L'environnement dynamique des systèmes ambiants offre des informations contextuelles aux agents intelligents qui s'y déploient. Dans de tels environnements, ces agents peuvent-ils collaborer pour mieux atteindre leurs objectifs individuels et collectifs, et ce en considérant leurs intentions multiples ? Cette coopération dépendra fortement des intentions des agents. Dans cet article, nous proposons de doter les agents ambiants d'un mécanisme de planification contextuelle appelé CPS qui peut s'étendre dans un contexte collectif. Nous présentons d'abord le CPS qui génère des plans contextuels optimaux pour un seul agent tout en satisfaisant plusieurs de ses intentions et en préservant la consistance du plan. Ensuite, nous étendons ce mécanisme coopératif de planification pour prendre en considération plusieurs agents ambiants. Appelé CCPS (collective CPS), il permet aux agents de déléguer partiellement leur plan et de collaborer durant l'exécution de leurs plans. Un scénario de travail extrait du Campus Intelligent est implémenté et discuté.

Mots-clés : Planification Contextuelle ; Système de planification ; Contexte Spatio-temporel ; Coopération ; Agents BDI ; Intelligence ambiante.

Abstract

Ambient Intelligent (AmI) environments dynamically provide contextual information to intelligent agents that interact with them. In such environments, could these agents cooperate to improve their goal achievement, considering multiple intentions from several agents ? With multiple agents, cooperation will depend on each agent's own intentions. Agents adapt to dynamic changes in the environment using context-aware planning mechanisms such as the Contextual Planning System (CPS). First, we present

the CPS which proposes an optimal plan for a single agent based on the current context that satisfies multiple intentions of the agent while preserving their consistency. Then, we present the Collective CPS (CCPS), an opportunistic cooperative planning mechanism for multiple agents in AmI environments. CCPS allows agents to partially delegate their own plans or to collaborate with other agents' plans during their execution, while retaining individual planning capabilities. A working scenario for a realistic AmI environment, such as a smart Campus, is presented.

Keywords: Contextual Planning System ; Spatio-temporal context ; BDI

1 Introduction

L'intelligence Ambiante (AmI) fait référence à des environnements électroniques dans lesquels les dispositifs ou systèmes électroniques peuvent percevoir et répondre à la présence des personnes, tout en étant capables de communiquer entre eux [1]. Dans de tels environnements, la coopération et la coordination entre différents systèmes peuvent exister sous différentes formes et à des fins différentes. Cet article présente une approche coopérative traitant des intentions multiples pour un type spécifique d'agent intelligent qui interagit sur des environnements de type AmI, dénommé agent ambiant (AA). Un exemple d'AA serait un agent déployé sur un téléphone intelligent capable de prendre les informations fournies par différents dispositifs existant dans une ville intelligente, et d'adapter leur comportement en conséquence. Le modèle coopératif mis en place est opportuniste, c'est-à-dire que la coopération ne peut exister que s'il y a des conditions favorables à sa réalisation. Puisque les environnements AmI fournissent des informations contextuelles dynamiques, le pro-

cessus de raisonnement des AA doit s'adapter aux changements de l'environnement tout en permettant d'atteindre ses objectifs.

AmI sont fréquemment utilisés en conjonction avec les applications contextuelles [8, 4, 33, 3, 24], car la prise en considération du contexte peut être définie comme la capacité du système à percevoir les caractéristiques actuelles de son environnement afin de décider des mesures adéquates à prendre en considération. Dans la littérature, plusieurs types de contextes ont été défini, et cela en fonction des informations représentées. Par exemple, Olaru *et al.* [21] se concentre sur l'emplacement et l'heure, Ryan *et al.* [25] utilise l'emplacement, l'heure, l'identité et d'autres propriétés instantanées de l'environnement physique. Dey et Abowd [2] indiquent que les applications contextuelles *perçoivent le "qui", "où", "quand" et "quoi" des entités présentes dans l'environnement et utilisent ces informations pour déterminer pourquoi la situation se produit (sic).*

L'utilisation de systèmes multi-agents (MAS) pour traiter des informations contextuelles dans de telles applications est intéressant, où différentes entités interagissent de manière autonome dans un environnement dynamique et incertain [6, 23]. Néanmoins, MAS dont les agents suivent le modèle BDI sont conçus pour AmI en raison de son utilisation d'informations contextuelles sous la forme de *croyances*. Le mécanisme de raisonnement BDI associé permet de percevoir les changements d'environnement et de mettre à jour les *croyances*, ce qui alimente le mécanisme de planification de l'agent afin de sélectionner et d'atteindre différents objectifs parmi ses intentions [14]. D'autres travaux connexes impliquant BDI, MAS, et la planification incluant la recherche sur l'interaction des objectifs entre les agents, le processus de raisonnement BDI et la gestion des échecs en planification [19, 29, 26, 27, 30, 7, 32].

Plusieurs agents peuvent avoir des objectifs indépendants et peuvent avoir besoin de coopération pour exécuter leurs tâches. Cela peut être illustré sur une situation concrète comme suit : si une personne a un rendez-vous mais ne peut pas être là à temps (un objectif irréalisable), alors la personne peut demander de faire son trajet avec une autre personne afin de réaliser son objectif. Des situations similaires peuvent être observés dans d'autres domaines incluant des scénarios AmI, comme l'alerte cooperative de collisions dans les réseaux véhiculaires ad-hoc [15] et la résolution de problèmes opportunistes basés sur

l'information en utilisant la négociation ontologique [5].

Malgré qu'il y ait des recherches approfondies dans le domaine de la planification coopérative d'agents intelligents, il existe différentes contraintes ou restrictions qui rendent l'utilisation du travail existant dans le domaine AmI difficile. Un MAS auto-adaptatif temps réel dénommé par AMOEBA a été récemment présenté [20], où il a été possible de générer des modèles coopératifs auto-adaptatifs entre agents intelligents. Cependant, le système repose toujours sur une unité de traitement centralisée (sous la forme d'un agent Head, similaire à la plateforme TAEMS), et ses fonctionnalités sont orientées vers l'adaptation du contexte en temps réel. Dans ce cas, la coopération est utilisée comme une forme d'adaptation du contexte plutôt que comme la réalisation d'objectifs individuels par différents agents.

Nous proposons donc, un mécanisme de planification coopératif qui peut être utilisé par les AA afin de déléguer des plans spécifiques à d'autres agents, tout en prenant en considération des contraintes contextuelles spatiales et temporelles et en conservant leur processus de planification individuel. L'objectif d'un tel mécanisme est de fournir une coopération dans chacun des processus de planification des agents sans avoir à s'engager à des objectifs communs, ou d'utiliser des mécanismes de planification centralisés. En termes de contribution scientifique, ce travail permet également d'utiliser le parallélisme, la séquence et l'alternance en traitant simultanément des intentions multiples tout en maintenant le mécanisme de collaboration entre agents intelligents et strictement opportuniste. En outre, nous améliorons le mécanisme de raisonnement de l'agent BDI en utilisant un mécanisme de planification préemptif en conjonction avec la planification contextuelle. Ainsi, le mécanisme proposé permet à différents agents de coopérer entre eux afin d'atteindre des objectifs réalisable qu'en déléguant l'exécution de certains plans. Il est important de noter que le mécanisme de coopération n'assume pas d'agents bienveillants, ni qu'il est basé sur des engagements existants de toute sorte : les AA ne coopèrent entre eux que si c'est dans leur intérêt.

Ce papier est organisé comme suit : La section 2 présente les aspects pertinents de la planification et de la coopération utilisés dans ce papier. La section 3 décrit le système de planification contextuelle (CPS [13]). La section 4 explique le CPS collectif (CCPS) en présentant sa struc-

ture formelle et les algorithmes associés. La section 5 présente, comme preuve de concept, une implémentation de CCPS dans un scénario basé sur le monde réel, et décrit les expérimentations effectuées. Enfin, la section 6 conclut ce papier et présente quelques perspectives et les travaux futurs.

2 Planification et Cooperation

Le processus de planification utilisé par un agent BDI implique l'utilisation de ses informations internes pour choisir un ensemble d'actions à exécuter et où le processus de planification est activé à chaque fois qu'une nouvelle information est perçue. La re-planification en utilisant de nouvelles informations permet à l'agent d'adapter ses plans en conséquence et de réévaluer ses objectifs à chaque interaction, afin d'adopter à cet instant la meilleure ligne de conduite.

Les changements environnementaux peuvent se produire indépendamment de l'agent (comme l'écoulement du temps), ou les conséquence de l'exécution de certaines actions. Dans un MAS, les actions effectuées par un agent peuvent avoir un impact sur les perceptions des autres agents s'exécutant dans le même environnement. Si différents agents ont des objectifs différents, les actions d'un agent peuvent avoir un tel effet dans l'environnement que les objectifs d'un autre agent peuvent devenir irréalisables.

Dans le cas où plusieurs agents ont les mêmes objectifs, un mécanisme de coordination peut être utilisé pour minimiser l'impact négatif de l'exécution des actions sur l'objectif global réalisable [16]. Cependant, il existe des situations où plusieurs agents ont des objectifs indépendants ou des tâches individuelles visant à résoudre un problème plus complexe. Ces situations peuvent nécessiter une coopération, par exemple, en augmentant l'efficacité globale de la planification pour un environnement dynamique [18, 22]. Il se peut également qu'une planification coordonnée soit nécessaire pour résoudre le problème.

Des situations comme celles-ci nécessitent des processus de planification qui prennent en compte la coopération. Bien qu'il existe des solutions visant à résoudre des problèmes ou des situations spécifiques citedi2016agent, torreno2014fmap, engesser2017cooperative, ces solutions considèrent des objectifs conjoints, soit en s'appuyant sur des systèmes de coordination centraux (à des fins de communication) soit limitées par le domaine.

Une coopération complète entre les agents ne peut pas conduire à des solutions optimales sans l'utilisation de techniques auxiliaires telles que l'apprentissage par renforcement [10]. Néanmoins, AA pourrait bénéficier d'une planification coopérative dans des situations où l'efficacité individuelle est affectée par les conditions environnementales. Dans ce cas, les agents exposés à différents contextes pourraient exécuter certaines tâches de manière plus efficace. Si plusieurs agents coopèrent, des tâches dépendant du contexte pourraient être déléguées parmi les agents, et des objectifs irréalisables pourraient être atteints avec succès.

Notre approche vise à explorer des situations où des tâches contextuelles peuvent être déléguées par un agent dans le cadre de son processus de planification, donnant au processus de coordination une caractéristique contextuelle dynamique. Nous détaillerons cette approche dans les paragraphes suivants allant du formalisme impliqué au mécanisme de planification.

3 Système de Planification Contextuel (CPS)

Dans le modèle original agent-BDI, un mécanisme qui gère les différents états d'un agent liés à ses croyances (B), ses désirs (D) et ses intentions (I) afin d'atteindre les objectifs grâce à la planification. Soit \mathbb{A} l'ensemble des agents et $A_i \in \mathbb{A} = \{A_1, A_2, \dots, A_n\}$.

Dans [13], le modèle proposé traite plusieurs intentions pour un agent A_i . Il représente un plan d'un agent $\tilde{P}(A_i)$ comme étant une structure arborescente composée d'un ensemble de plans d'intentions (\hat{P}_j) et de plans élémentaires ($P_{l,k}$). $\tilde{P}(A_i)$ est composé de plusieurs plans d'intentions \hat{P}_j , et chaque plan d'intention correspond à la réalisation d'une intention d'un agent. L'ensemble des plans d'intention est géré en utilisant les opérateurs $|||$ ou séquentiels \ggg simultanés, car ces plans d'intention peuvent être exécutés en parallèle ou en séquence dans le plan de l'agent. Soit $\bar{\mathcal{P}}(A_i)$ l'ensemble des noms qualifiant les plans possibles d'un agent $\tilde{P}(A_i) \in \bar{\mathcal{P}}(A_i)$ et $\hat{\mathcal{P}}$ l'ensemble des noms utilisées pour identifier une possible intention avec $\hat{P} \in \hat{\mathcal{P}}$, où $\tilde{P}(A_i)$ est un plan d'agent défini par :

$$\tilde{P}(A_i) ::= \hat{P} \mid \tilde{P}(A_i) ||| \tilde{P}(A_i) \mid \tilde{P}(A_i) \ggg \tilde{P}(A_i)$$

Chaque plan d'intention \hat{P}_k ($1 \leq k \leq q$) peut être une alternative de plusieurs plans élémentaires : $\hat{P}_k = \{P_{k,0}, \dots, P_{k,q}\}$. L'alternance des plans élémentaires permet d'atteindre une intention spécifique de différentes manières, chacune exprimée par un plan élémentaire.

Cette relation de composition alternative est prise en considération dans AgLOTOS [11] en utilisant l'opérateur \diamond . Par conséquent, une intention est satisfaite si et seulement si au moins un de ses plans élémentaires associés est exécuté avec succès. Formellement, soit \mathcal{P} l'ensemble des noms qualifiant les plans élémentaires possibles avec $P \in \mathcal{P}$. Nous définissons un plan d'intention \hat{P} comme :

$$\hat{P} ::= P \mid \hat{P} \diamond \hat{P}$$

Les plans élémentaires $P_{k,j}$ sont décrits par les expressions AgLOTOS, en se référant aux expressions de comportement $E_{i,k}$ exprimées dans un ensemble fini ordonné d'actions observables a_i à exécuter par l'agent. Il est important de noter que pour chaque action il est associée une dimension temporelle, donnée par sa durée (en secondes) et éventuellement comme conditions d'exécution réussie (une action donnée peut ne pas pouvoir être exécutée avant ou après un certain temps). Toute expression AgLOTOS est associée à des informations contextuelles liées à l'état actuel d'un agent BDI. Plus de détails sur AgLOTOS ont été fournis par Chaouche et al. [11]. Ces plans élémentaires sont obtenus à partir d'une bibliothèque de plans (*LibP*).

L'avantage de cette structure particulière est que les plans correspondant à chaque intention sont réalisés simultanément en utilisant un mécanisme de pondération afin de privilégier certaines intentions par rapport aux autres. En ordonnant les plans d'intention (\hat{P}_i), on évite les situations conflictuelles et, lorsque les intentions ne sont pas contradictoires, le même poids permet l'accomplissement simultané de leurs plans d'intention d'une manière combinée. De plus, l'utilisation de plans alternatifs permet au processus de planification d'envisager différents moyens pour réaliser leurs intentions associées. En outre, nous supposons que pour chaque agent, son *LibP* est indexé par l'ensemble de toutes les intentions possibles, fournissant des plans élémentaires ($P_{i,k}$) qui sont utilisés pour atteindre chacune des intentions de l'agent.

En utilisant ce processus en conjonction avec des informations contextuelles, Chaouche *et al.* [12]

a proposé un mécanisme de planification prédictif appelé Contextual Planning Guidance. Ce mécanisme utilise les informations contextuelles perçues par l'agent pour vérifier quelles actions (parmi les actions existantes connues de l'agent) sont réalisables. La "faisabilité" d'une action donnée est déterminée par toutes les conditions que l'agent doit remplir pour l'exécuter. Par exemple, passer du point A au point B n'est réalisable que si l'agent est au point A. Une fois cette vérification effectuée, le système de planification contextuelle (Contextual Planning System - CPS) sélectionne un chemin d'exécution qui maximise le nombre d'intentions pouvant être satisfaites [12], produisant ainsi un plan optimal $\tilde{P}(A_i)$. Dans la section suivante, nous expliquerons comment ce plan est utilisé par le système de planification contextuelle collective (Collective Contextual Planning System - CCPS) en conjonction avec son mécanisme coopératif.

4 Système de Planification Contextuelle Collective (CCPS)

La structure du système de planification contextuelle collective (CCPS) est réalisé pour traiter les plans à partir de plusieurs CPS agents. En comparaison avec le mécanisme du CPS, la nouveauté est que lorsqu'un plan d'agent ne peut être exécuté pour une raison quelconque, l'agent peut demander de l'aide et déléguer une partie de son plan (plutôt que de déclencher un processus de re-planification sur ses objectifs) à d'autres agents. Cela permet à l'agent d'éliminer le problème tout en maintenant la cohérence de ses objectifs. Ceci est fait en transférant l'intention(s) associée avec le plan non exécuté à d'autres agents, ou en éliminant l'intention(s).

Dans la première étape du mécanisme CCPS, un agent a un plan initial résultant de l'exécution du mécanisme CPS et vérifie s'il peut être exécuté sans l'aide d'un autre agent. Si ce n'est pas le cas, l'agent détecte les actions qui empêchent l'exécution du plan. Ces actions sont répertoriées dans un message de demande d'aide envoyé à d'autres agents.

Après avoir reçu ce message, chaque agent décide pour chaque action de l'aider ou de ne pas l'aider par envoi de message de réponse. L'agent demandeur décide d'accepter de l'aide et informe l'agent auxiliaire de sa décision.

Dans le cas où aucun autre agent n'est en mesure de l'aider, l'agent tente de remplacer l'intention associée au plan qui ne peut pas être exécutée

avec une autre qui pourrait fournir des résultats similaires. Si aucune intention n'est trouvée, l'intention originale est supprimée.

L'idée derrière ce mécanisme est similaire au protocole net de contrat [28], qui a été conçu pour la résolution de problèmes distribués. Les aspects liés à la connexion et à la négociation de contrat sont toutefois simplifiés - soit en considérant qu'il est fourni par l'environnement de l'agent (abstraction de la manière dont les agents sont connus et contactés) ou en prenant en considération l'aspect opportuniste (s'il y a pas de connexion, il n'y a pas de coopération). Les concepts impliqués dans ce mécanisme seront formalisés et expliqués dans les paragraphes suivants.

4.1 Formalisation

Dans cette section, nous présentons quelques définitions afin de décrire formellement le CCPS et son algorithme.

- **Environnement** : Nous définissons un ensemble d'informations contextuelles qui peuvent être perçues par n'importe quel agent à un instant t , comme étant un environnement $Env(t)$. Nous supposons que tous les AA peuvent percevoir l'environnement de la même manière et le changer en exécutant une action.

Nous formalisons $Env(t)$ comme un ensemble fini non ordonné de propositions logiques rencontrées dans l'environnement, au temps donné t , $Env(t) = \{l_1, \dots, l_n\}$. Chacune des propositions logiques l_i ($1 \leq i \leq n$) décrit un aspect différent de l'environnement perçu par l'agent. Par exemple, être à un emplacement déterminé X à un instant donné t est perçu comme $in(X) \in Env(t)$.

- **Action** : Une action a représente une activité effectuée par l'agent dans l'environnement, de manière granulaire la plus fine. Si un agent A_i se déplace du point X au point Y , il y a une action $move(X, Y)$ qui représente cette activité. Chaque action a est soumise à trois ensembles de propositions logiques (comme c'est le cas dans le formalisme STRIPS [17]) : liste des pré-conditions ($Pre(a)$) - la liste des propositions logiques doit être vraie avant que l'action ne soit effectuée; liste de suppression ($Del(a)$) la liste des propositions logiques à supprimer après l'exécution de l'action et la liste des post-conditions ($Post(a)$) est la liste des propositions logiques qui devient vraie après que l'action a été effectuée. Les actions possèdent également

une durée (en secondes), et sa liste de conditions préalables peut inclure une limite de temps spécifique. Pour qu'une action a soit réalisable, toutes ses conditions préalables $Pre(a)$ doivent être satisfaites par les conditions de $Env(t)$. Cela signifie qu'une action a est réalisable dans un environnement $Env(t)$, si $\forall y \in Pre(a), y \in Env(t)$. Par contre, la même action a est irréalisable si au moins l'une de ses conditions préalables ne peut être remplie. Par conséquent, a est irréalisable dans un environnement $Env(t)$, if $\exists y \in Pre(a), y \notin Env(t)$. Il est également important de noter que des temps différents sont implicites aux conditions pré-conditions et post-conditions : les pré-conditions se réfèrent toujours au temps t , tandis que les post-conditions ne sont satisfaites qu'au $t + 1$ (après l'exécution de l'action).

- **Intention** : Chaque agent A_i a un ensemble d'intentions $I(A_i)$, et chaque intention est associée à un facteur de pondération $w \in \mathbb{N}^*$ qui sert d'indice de comparaison. Pour un ensemble d'intentions donné $I(A_i) = \{(i_0, w_0), \dots, (i_n, w_n)\}$, deux intentions $(i_p, w_p), (i_q, w_q) \in I(A_i), 0 \leq p, q \leq n$ sont comparables en fonction de leurs facteurs de pondération : (i_p, w_p) est au moins aussi important que (i_q, w_q) if $w_q \leq w_p$. De même pour les actions, toute intention donnée $(i_j, w_j) \in I(A_i)$ ($1 \leq j \leq n$) a des pré-conditions ($Pre((i_j, w_j))$) et post-conditions ($Post((i_j, w_j))$). Les intentions peuvent également être satisfaites ou insatisfaites. Nous considérons une intention (i_j, w_j) satisfaite pour un environnement $Env(t)$ si et seulement si toutes ses post-conditions peuvent être satisfaites dans l'environnement actuel : $Post((i_j, w_j)) \subseteq Env(t)$. Par conséquent, une intention (i_j, w_j) est insatisfaite si $\exists p \in Post((i_j, w_j)) \wedge p \notin Env(t)$. L'ensemble des intentions non satisfaites d'un agent A_i est défini dans $I_{Fail}(A_i)$.

- **Plan** : Comme mentionné précédemment, le plan d'un agent $\tilde{P}(A_i)$ est composé de plusieurs plans d'intention $\hat{P}_j : \tilde{P}(A_i) = \{\hat{P}_1, \dots, \hat{P}_j\}$. Chaque \hat{P}_q ($1 \leq q \leq j$) est aussi composé de plusieurs plans élémentaires : $\hat{P}_q = \{P_{q,0}, \dots, P_{q,k}\}$. Les plans élémentaires sont des ensembles d'actions ordonnées a_i à exécuter par l'agent. Par conséquent, un plan d'agent $\tilde{P}(A_i)$ peut être défini comme un ensemble de n actions partiellement ordonnées $a_r : \tilde{P}(A_i) = \{a_1, a_2, \dots, a_n\}$ ($1 \leq r \leq n$) connu par un agent A_i . En tant qu'ensemble partiellement ordonné, certaines actions d'un plan doivent être exécutées

tées avant ou en parallèle avec d'autres actions.

• **Faisabilité du Plan :** Étant donné un ensemble d'actions et partiellement ordonnées, la faisabilité d'un plan élémentaire $P_{i,j}$ peut être donnée par la faisabilité de ses actions : $P_{j,k}$ est réalisable dans un environnement $Env(t)$ si et seulement si toutes ses actions sont réalisables dans $Env(t)$. Comme indiqué précédemment, chaque plan d'intention \hat{P}_j est réalisable si et seulement si au moins un de ses plans élémentaires est réalisable. Dans le cas d'un plan d'un agent $\tilde{P}(A_i)$, nous déterminons qu'il est complètement réalisable si et seulement si tous ses plans d'intention sont réalisables. Cette expression peut être mieux exprimée en définissant $L_{nR}(t)$ comme étant un ensemble d'actions irréalisables à un instant t (dans un environnement $Env(t)$, où toutes les actions font partie d'un plan $\tilde{P}(A_i)$).

Néanmoins, il peut y avoir des situations où il est nécessaire de considérer la faisabilité partielle d'un plan d'agent $\tilde{P}(A_i)$.

Supposons que $\tilde{P}(A_i)$ contienne un ensemble de plans d'intention $\{\hat{P}_1, \dots, \hat{P}_q\}$ faisant référence à $I_j = \{(i_0, w_0), \dots, (i_j, w_j)\}$, $I_j \text{ subseteq } I(A_i)$. Dans ce cas où seulement \hat{P}_j est irréalisable et w_j est le poids le moins élevé que relativement aux autres poids des autres intentions, cela a du sens pour le plan $\tilde{P}(A_i)$ afin d'être exécuté, même si l'intention I_j est toujours irréalisable. D'autre part, si - pour le même scénario - nous avons une situation où w_j est relativement plus grand que les poids des autres intentions, nous ne devrions pas vouloir continuer avec $\tilde{P}(A_i)$ et la re-planification devient nécessaire. Considérons aussi les notations suivantes : le poids moyen de toutes les intentions associées à tous les plans d'intention dans $\tilde{P}(A_i)$ est donné par $\overline{w_I}$, et le poids moyen de toutes les intentions associées à tous les plans d'intention faisable dans $\tilde{P}(A_i)$ sont donnés par $\overline{w_F}$. Dans cet esprit, nous définissons le concept de faisabilité partielle pour un plan d'agent : $\tilde{P}(A_i) = \{\hat{P}_1, \dots, \hat{P}_j\}$ est partiellement réalisable si $\overline{w_F} \geq \overline{w_I}$. Dans la suite de ce travail, nous considérerons un plan d'agent $\tilde{P}(A_i)$ irréalisable s'il n'est pas partiellement réalisable. De la même manière, un plan $\tilde{P}(A_i)$ sera considéré comme faisable s'il est au moins partiellement réalisable.

Il est également important de formaliser quelques aspects liés à la communication des agents - en particulier, les aspects impliquant les

messages échangés entre les agents, et leurs possibles conséquences sur le mécanisme du CCPS. Lorsqu'un agent A_i demande de l'aide à d'autres agents, cela se fait via un message M contenant au moins une action irréalisable (du point de vue de l'expéditeur). Ce message contient également les *limites temporelles* et une *deadline*. Les limites temporelles sont associées à des actions et sont formées par deux limites différentes : un début b (quand l'action doit commencer) et une fin e (quand l'action doit être terminée). Le délai ($t_{deadline}$), est associé au message et représente la limite de temps pour répondre au message. Une fois le délai expiré, l'agent n'attendra aucune réponse à ce message. En outre, lorsqu'un agent A_i envoie un message, il choisit les destinataires du message en utilisant un ensemble de filtres $F = f_1, \dots, f_n$. Ces filtres sont utilisés pour sélectionner un sous-ensemble d'agents Ag_F parmi les agents connus par A_i ($Ag(A_i) \subseteq A$) qui pourrait potentiellement aider A_i à exécuter les actions contenues dans M . Le sous-ensemble complémentaire d'agents connus qui pourraient ne pas aider l'agent A_i (étant donné les actions contenues dans M) est également généré à ce moment, et appelé comme Ag_E ($Ag_F \cup Ag_E = Ag(A_i)$).

5 Illustration

Afin d'illustrer le bon fonctionnement du CCPS, nous avons implémenté une version de ce mécanisme et l'avons utilisée en conjonction avec un scénario d'application. Le mécanisme CCPS est illustré dans la figure 1. Pour valider correctement la mise en œuvre du CCPS, nous avons suivi un processus d'expérimentation formel approprié pour des études de cas dans le domaine de l'ingénierie logicielle [31].

Un autre objectif de cette mise en œuvre était d'observer comment le CCPS fonctionnerait dans différentes situations. À ce stade de notre recherche, notre intention était d'observer si le CCPS pouvait identifier correctement les situations impliquant une coopération opportuniste et agir en conséquence. Pour ce faire, nous avons décidé d'étudier un scénario spécifique en considérant (i) des agents isolés, sans communication entre eux, et (ii) des agents capables de communiquer et de déléguer des tâches individuelles entre eux. Nous avons utilisé la formalisation du CCPS afin de déterminer quel serait le comportement dans les deux situations, et de comparer le résultat des expériences avec une base théorique. En utilisant le même scénario dans différentes conditions contextuelles (en simu-

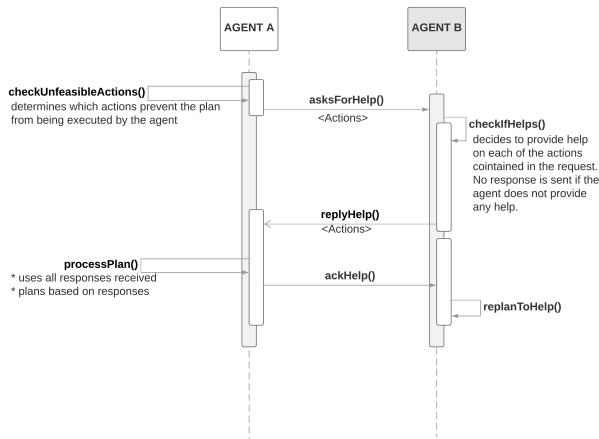


FIGURE 1 – Aperçu du mécanisme du CCPS

lant un environnement dynamique), nous avons comparé comment l'utilisation du nouveau mécanisme de coopération opportuniste affecterait la performance de chaque agent en termes d'atteinte des objectifs individuels. Nous nous attendions à ce que l'utilisation du mécanisme CCPS permette aux agents d'atteindre leurs objectifs individuels plus fréquemment lorsqu'ils sont exposés à des conditions environnementales différentes.

Le protocole d'étude utilisé dans le processus d'illustration était : (i) En utilisant le concept de coopération opportuniste comme base, créer un scénario avec des résultats prévisibles; (ii) Définir les situations dans lesquelles le scénario serait exécuté; (iii) Définir le résultat théorique du scénario pour chacune des situations; (iv) Exécuter le scénario pour chacune des situations, en utilisant l'implémentation de preuve de concept; (v) Analyser les résultats, en comparant les données obtenues avec les résultats attendus pour chaque situation; et (vi) Discuter des résultats, en soulignant toutes les questions pertinentes concernant l'évaluation et ses éléments.

Les sous-sections suivantes détaillent les éléments décrits dans le protocole d'étude, ainsi que la mise en œuvre du CCPS et d'autres informations pertinentes.

5.1 Implémentation

La mise en œuvre d'un mécanisme de preuve de concept pour le mécanisme CCPS impliquait trois aspects différents : (i) une structure BDI qui pourrait être utilisée pour les agents, ainsi que des plans et des actions; (ii) représenter des informations contextuelles d'une manière

qui pourrait être utilisée de manière coopérative par le mécanisme de planification; et enfin (iii) le mécanisme de planification. L'implémentation a été faite en Java en réutilisant une implémentation CPS précédente [9] comme base, qui incluait également une structure BDI. Cette même implémentation de CPS a été utilisée pour les agents non communicants utilisés dans l'expérience.

5.2 Scénario

Afin d'évaluer la mise en œuvre du CCPS, nous avons modélisé un scénario impliquant quatre agents différents : Alice, Bob, Claire et Damien (Figure 2). Tous les agents jouent différents rôles au sein du campus d'une université. Certains agents pourraient coopérer pour atteindre leurs objectifs personnels, comme récupérer une ressource spécifique (livre) ou se rendre à un endroit précis (stationnement) à une heure précise (prendre l'autobus en respectant l'horaire d'autobus). La coopération entre les agents a été passivement stimulée par l'utilisation de restrictions contextuelles, telles que les moyens de transport, les emplacements initiaux des agents, les restrictions temporelles (durée des différentes actions, délais dans lesquels elles doivent être exécutées), etc. Le cadre du scénario est un campus universitaire, plus précisément quelques emplacements de ce campus : un laboratoire, trois bureaux différents, un stationnement, un arrêt d'autobus et la résidence de chaque agent. Chacun des agents possède les routines suivantes (utilisées pour modéliser ses intentions) : (i) *Préparation du matin* (actions qui ont lieu avant que chaque agent quitte sa résidence); (ii) *Aller à l'université*; (iii) *Aller travailler*; et (iv) *Retour à la maison*. De plus, des conditions individuelles ont été attribuées aux agents (Alice et Claire ont des voitures, mais Bob et Damien prennent l'autobus, Alice et Bob travaillent dans le même bureau).

5.3 Situations et résultats attendus

Le scénario décrit ci-dessus a été exécuté sur trois conditions d'environnement différentes (contextes), en considérant le temps de départ (suffisamment de temps pour atteindre des intentions sensibles au temps), des intentions intrinsèquement irréalisables et la présence d'intentions équivalentes. Compte tenu des situations expérimentales, les résultats attendus dans les paragraphes suivants se réfèrent aux situations où la communication entre les agents est possible. La

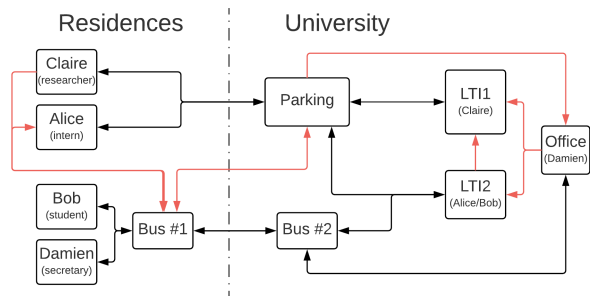


FIGURE 2 – Campus : les flèches indiquent les routines de mouvement ; les flèches rouges sont des routines modifiées pour la coopération

situation où les agents ne peuvent pas communiquer entre eux a été utilisée comme référence pour mesurer l'efficacité de la mise en œuvre.

Dans le premier contexte (1), le scénario a commencé à 8 heures et chaque agent a reçu un ensemble d'intentions à évaluer et éventuellement à réaliser le jour (par exemple, aller travailler, travailler, revenir à la maison). Ces intentions ne nécessitaient pas nécessairement une coopération des agents, et les restrictions temporelles liées à toutes les actions en cause étaient établies de telle manière qu'aucune coopération n'était nécessaire à aucun moment du processus de planification. Dans ce contexte, nous nous attendions à ce qu'aucun des agents ne demande de l'aide.

Dans le deuxième contexte (2), l'heure de départ était maintenue à 8 heures, mais quelques intentions insatisfaisantes étaient également incluses (par exemple, demander un livre aux agents quand aucun d'entre eux ne l'avait), sans intention équivalente ou complémentaire être utilisé. Dans le cas des intentions satisfaisantes, dans certains cas nous avons inclus des intentions équivalentes pour atteindre le même but (rentrer à la maison en bus ou faire un tour dans la voiture d'un autre agent). Dans ce contexte, nous nous attendions à ce que les agents ayant des intentions insatisfaisantes demandent de l'aide aux autres agents ; Cependant, comme ils étaient conçus pour être absolument irréalisables, nous nous attendions à ce que les agents demandent de l'aide et, en l'absence de réponses, qu'ils essaient de trouver des intentions équivalentes qui pourraient être utilisées. En fin de compte, nous nous attendions à ce que les agents rejettent les intentions irréalisables et planifient en utilisant les intentions restantes.

Dans le troisième contexte (3), nous avons démarré le scénario à 9 heures du matin, mais incluons des intentions irréalisables avec des intentions réalisables équivalentes. Ce contexte était supposé (i) pour permettre une coopération opportuniste et (ii) pour permettre aux agents d'utiliser correctement le mécanisme d'intention équivalent. Dans ce contexte, nous nous attendions à ce que les agents utilisent plus de temps dans le processus de planification, mais aussi pour atteindre plus d'objectifs (par rapport à la situation de non-communication).

5.4 Résultats d'exécution

Une fois codé et paramétré, le scénario décrit dans la section 5.2 a été exécuté, et les résultats ont été comparés aux résultats attendus pour chaque contexte. Dans la première expérience (contexte 1), tous les agents se sont comportés comme prévu. Les agents non-communicants et les agents communicants étaient capables de générer des plans réalisables pour toutes leurs intentions en utilisant le même temps.

Dans la deuxième expérience (contexte 2), les agents communicants ont passé beaucoup plus de temps (environ 200% de plus) à demander de l'aide et à essayer de trouver une intention équivalente avant d'atteindre le même plan que les agents non communicants. Cependant, le résultat était toujours conforme aux résultats attendus.

Dans la troisième expérience (contexte 3), les agents communicants ont également pris beaucoup plus de temps pour atteindre un plan final, mais dans ce cas leurs plans avaient aussi des intentions beaucoup plus réalisables que les agents non-communicants. Dans certains cas, lorsque toutes les intentions dépendaient de la réalisabilité d'une seule (durée et limite temporelle pour atteindre des emplacements spécifiques), certains agents non communicants avaient beaucoup moins d'intentions dans leur plan, puisque les autres n'étaient tout simplement pas réalisables (comme attendu). D'un autre côté, le plan créé par les agents communicants, bien que réalisable, a pris beaucoup de temps à être généré. La différence dans les temps d'exécution pour tous les contextes est représenté sur la Figure 3.

6 Conclusion et perspectives

Dans cet article, nous présentons un nouveau mécanisme de planification contextuelle coopérative (CCPS) utilisé par les AA. CCPS a été

		Execution	
		NC	C
Context	1	<10s	<10s
	2	<1m	<5m
	3	<3m	<15m

FIGURE 3 – Temps d’exécution pour les agents non communicants (NC) et communicants (C)

conçu pour réutiliser un mécanisme de planification contextuelle existant (CPS) en conjonction avec une coopération basée sur la délégation. La structure résultante permet à un agent d’atteindre des objectifs irréalisables et cela grâce à la coopération avec d’autres agents.

Afin d’illustrer la structure proposée, nous avons mis en œuvre une preuve de concept du mécanisme CCPS et l’avons testé sur un scénario basé sur un campus universitaire. Ce scénario impliquait plusieurs agents avec des intentions prédéfinies, organisées de telle sorte qu’il ne serait pas possible pour tous les agents de réaliser toutes leurs intentions sans coopération par délégation. Après avoir testé le CCPS avec ce scénario, nous avons observé que tous les agents étaient capables de réaliser leurs intentions, ce qui nous a permis de valider ce mécanisme.

Selon nos expérimentations, il y a quelques considérations à prendre en compte. Le premier est lié à l’efficacité du plan par rapport au temps nécessaire de génération du plan : dans nos expérimentations, nous avons remarqué une différence considérable sur le temps utilisé par les agents communicants pour atteindre un plan réalisable. Cela posera problème lors de la mise en œuvre pratique. Il nous semble plus appropriée, afin d’optimiser les résultats des algorithmes, d’utiliser un langage de programmation différent pouvant atténuer la plupart des problèmes rencontrés. Concernant la consommation de ressources informatiques, le mécanisme était particulièrement intense lors du calcul d’intentions équivalentes. Nous avons l’intention d’explorer ces limites dans des travaux futurs.

Une autre considération était liée au processus de communication. Lorsque le mécanisme CCPS trouve une intention irréalisable, le processus de communication qui se déroule est simple, mais pas optimal. Cela signifie que dans certains cas - s’il y a suffisamment d’agents avec des actions insatisfaisantes - l’environnement peut être inondé par des messages de demande d’aide, ce qui aura certainement un impact négatif sur le processus global de génération de

plan. Nous avons également l’intention d’étudier cette limitation en testant différents mécanismes de contraintes, tels que la limitation du nombre de demandes d’aide d’un agent ou la détermination de conditions plus flexibles mais restrictives concernant les agents pouvant recevoir une demande d’aide d’un autre agent. Dans les scénarios du monde réel, ces restrictions se produisent naturellement sous la forme de distance (e.g. à quelle distance quelqu’un est, afin d’entendre ou de recevoir une demande d’aide).

Enfin, de nombreux aspects doivent être explorés dans le mécanisme CCPS proposé. Nous pouvons également étudier l’effet en cascade dans des scénarios plus complexes (c.à.d. la possibilité pour un agent de déléguer une tâche qui lui a été déléguée). Pour l’instant, cette possibilité a été exclue dans notre mise en œuvre. Différentes parties de l’algorithme peuvent être étudiées en détail, et de nouveaux concepts incorporés tels que des intentions complémentaires et équivalentes peuvent être affinés. La mise en œuvre peut également être étendue pour prendre en charge plusieurs variantes de scénario dans l’environnement d’exécution, et des scénarios plus complexes peuvent être utilisés pour vérifier le fonctionnement du mécanisme CCPS dans différentes conditions.

Références

- [1] Emile Aarts and Reiner Wichert. Ambient intelligence. In *Technology Guide*, pages 244–249. Springer, 2009.
- [2] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer Berlin Heidelberg, 1999.
- [3] Giovanni Acampora, Diane J Cook, Parisa Rashidi, and Athanasios V Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12) :2470–2494, 2013.
- [4] Mohammed F Alhamid, Majdi Rawashdeh, Haiwei Dong, M Anwar Hossain, Abdulhameed Alelaiwi, and Abdulmotaleb El Saddik. Recam : a collaborative context-aware framework for multimedia recommendations in an ambient intelligence environment. *Multimedia Systems*, 22(5) :587–601, 2016.
- [5] Sidney C Bailin and Walt Truszkowski. Cooperation between intelligent information agents. In *International Workshop on Cooperative Information Agents*, pages 223–228. Springer, 2001.
- [6] Matteo Baldoni, Jörg P. Müller, Ingrid Nunes, and Rym Zalila-Wenkstern, editors. *Engineering Multi-Agent Systems - 4th International Workshop, EMAS 2016, Singapore, Singapore, May 9-10, 2016, Revised, Selected, and Invited Papers*, volume 10093 of *Lecture Notes in Computer Science*. Springer, 2016.

- [7] Radja Boukharrou, Ahmed-Chawki Chaouche, Amal El Fallah Seghrouchni, Jean-Michel Ili , and Djamel Eddine Saïdouni. Dealing with temporal failure in ambient systems : a dynamic revision of plans. *Journal of Ambient Intelligence and Humanized Computing*, 6(3) :325–336, 2015.
- [8] Jos  Bravo, Diane Cook, and Giuseppe Riva. Ambient intelligence for health environments, 2016.
- [9] Arthur Casals, Amal El Fallah Seghrouchni, and Anarosa Alves Franco Brand o. Augmented agents : Contextual perception and planning for bdi architectures. In *(Informal) Proceedings of the 5th International Workshop on Engineering Multi-Agent Systems (EMAS), May, S o Paulo, Brazil*, volume 1, pages 33–48, 2017.
- [10] Muthukumaran Chandrasekaran, Prashant Doshi, Yifeng Zeng, and Yingke Chen. Can bounded and self-interested agents be teammates? application to planning in ad hoc teams. *Autonomous Agents and Multi-Agent Systems*, 31(4) :821–860, 2017.
- [11] Ahmed-Chawki Chaouche, Amal El Fallah Seghrouchni, Jean-Michel Ili , and Djamel Eddine Saïdouni. A Higher-order Agent Model with Contextual Management for Ambient Systems. In *Transactions on Computational Collective Intelligence XVI*, volume 8780 of LNCS, pages 146–169. Springer Berlin Heidelberg, 2014.
- [12] Ahmed-Chawki Chaouche, Amal El Fallah-Seghrouchni, Jean-Michel Ili , and Djamel Eddine Saïdouni. From intentions to plans : A contextual planning guidance. In *Intelligent Distributed Computing VIII*, pages 403–413. Springer International Publishing, 2015.
- [13] Ahmed-Chawki Chaouche, Amal El Fallah-Seghrouchni, Jean-Michel Ili , and Djamel-Eddine Saïdouni. Learning from situated experiences for a contextual planning guidance. *J. Ambient Intelligence and Humanized Computing*, 7(4) :555–566, 2016.
- [14] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artif. Intell.*, 42(2-3) :213–261, 1990.
- [15] Amit Dua, Seema Guide Bawa, and Neeraj Guide Kumar. *Efficient Data Dissemination in Vehicular Ad Hoc Networks*. PhD thesis, 2016.
- [16] Edmund H Durfee. Scaling up agent coordination strategies. *Computer*, 34(7) :39–46, 2001.
- [17] Richard E Fikes and Nils J Nilsson. Strips : A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4) :189–208, 1971.
- [18] David Jung and Alexander Zelinsky. An architecture for distributed cooperative planning in a behaviour-based multi-robot system. *Robotics and Autonomous Systems*, 26(2-3) :149–174, 1999.
- [19] Brian Logan, John Thangarajah, and Neil Yorke-Smith. Progressing intention progression : A call for a goal-plan tree contest. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, S o Paulo, Brazil, May 8-12, 2017*, pages 768–772. ACM, 2017.
- [20] Julien Nigon, Marie-Pierre Gleizes, and Fr d ric Migeon. Self-adaptive model generation for ambient systems. *Procedia Computer Science*, 83 :675–679, 2016.
- [21] Andrei Olaru, Adina Magda Florea, and Amal El Fallah-Seghrouchni. A context-aware multi-agent system as a middleware for ambient intelligence. *MONET*, 18(3) :429–443, 2013.
- [22] Eiichi Osawa. A metalevel coordination strategy for reactive cooperative planning. In *ICMAS*, volume 95, pages 297–303, 1995.
- [23] Ferdinand Piette, Costin Caval, C dric Dinont, Amal El Fallah-Seghrouchni, and Patrick Taillibert. A multi-agent solution for the deployment of distributed applications in ambient systems. In *Engineering Multi-Agent Systems - 4th International Workshop, EMAS 2016, Singapore, Singapore, May 9-10, 2016, Revised, Selected, and Invited Papers*, pages 156–175, 2016.
- [24] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3) :579–590, 2013.
- [25] N. S. Ryan, J. Pascoe, and D. R. Morse. Enhanced reality fieldwork : the context-aware archaeological assistant. In V. Gaffney, M. van Leusen, and S. Exxon, editors, *Computer Applications in Archaeology 1997*, British Archaeological Reports, pages 182–196, Oxford, October 1998. Tempus Reparatum.
- [26] Sebastian Sardina and Lin Padgham. Goals in the context of bdi plan failure and planning. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 7. ACM, 2007.
- [27] Sebastian Sardina and Lin Padgham. A bdi agent programming language with failure handling, declarative goals, and planning. *Autonomous Agents and Multi-Agent Systems*, 23(1) :18–70, 2011.
- [28] Reid G Smith. The contract net protocol : High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, (12) :1104–1113, 1980.
- [29] Michael Winikoff. An integrated formal framework for reasoning about goal interactions. In *International Workshop on Declarative Agent Languages and Technologies*, pages 16–32. Springer, 2011.
- [30] Michael Winikoff and Stephen Cranefield. On the testability of bdi agent systems. *Journal of Artificial Intelligence Research*, 2014.
- [31] Claes Wohlin, Per Runeson, Martin H st, Magnus C Ohlsson, Bj rn Regnell, and Anders Wessl n. *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [32] Yuan Yao, Brian Logan, and John Thangarajah. Robust execution of bdi agent programs by exploiting synergies between intentions. In Dale Schuurmans and Michael P. Wellman, editors, *AAAI*, pages 2558–2565. AAAI Press, 2016.
- [33] Daqiang Zhang, Hongyu Huang, Chin-Feng Lai, Xuedong Liang, Qin Zou, and Minyi Guo. Survey on context-awareness in ubiquitous media. *Multimedia tools and applications*, 67(1) :179–211, 2013.