Réparation de plans dans les HTNs réactifs en utilisant la planification symbolique

Lydia Ould Ouali[1] Charles Rich[2] Nicolas Sabouret[1]

[1] LIMSI-CNRS, UPR 3251, Orsay, France
Univ. Paris-Sud, Orsay, France
[2] Worcester Polytechnic Institute
Worcester, MA, USA

ouldouali@limsi.fr

Résumé

Mots Clef

Abstract

Keywords

1 Introduction

Le domaine de la planification s'intéresse à la construction de séquences d'actions pour accomplir une tâche de manière automatique. Pour ce faire, les systèmes de planification sont définis autours de trois principaux modules. Le premier module définit une représentation formelle du monde appelée domaine de connaissances. Le second est un moteur de planification (planificateur) pour raisonner sur le domaine de connaissance et produire des plans qui achèvent les buts du système. Enfin, un moniteur d'exécution pour surveiller la bonne exécution des actions du plan dans le monde. Dans ce contexte deux approches complémentaires existent. La première approche est la planification symbolique. Cette dernière propose de définir une représentation logique complète et fidèle du monde pour permettre au système de planification de construire à l'avance des plans viables et ensuite les exécuter dans le monde. Le système de planification le plus répondu utilisant cette approche est HTN (i.e Hierarchical Task Network) [1]. Les HTNs permettent une décomposition récursives des tâches complexes en sous tâches plus simple qui peuvent être directement exécutées dans le monde.
Cependant, l'approche symbolique est basées sur deux hypothèses erronées. La première hypothèse affirme qu'il est possible de construire un domaine de connaissance complet d'un monde dynamique, mais plusieurs recherches ont prouvés que la modélisation complète d'un tel monde requiert un effort considérable d'ingénierie de la connaissance [2] si ce n'est impossible à réaliser [3]. La seconde hypothèse soutient que le plan est l'unique entités capable de modifier l'état du monde, alors qu'un monde dynamique peut être changé par différente entités comme d'autres agents évoluant dans ce monde. Ces limites ont pour conséquences de faire échouer l ?exécution du plan car d'une part si le monde est mal modélise, le plan produit ne représentera pas fidèlement le monde réel et d'une autre part les actions des autres entités du monde peuvent faire échouer l ?exécution d'une action du plan. L'échec de l'exécution du plan est appelé *cassure*. Par exemple, imaginons un agent qui planifie pour déplacer un objet d'une *Pièce1* a une *Pièce2*. La représentation HTN est définie dans figure 1. L'agent entame l ?exécution de la tâche *PickUpBox*, poursuit avec la tâche *NavigateDoor(Pièce1, Pièce2)*, imaginons maintenant qu'après avoir exécuté la tâche pour ouvrir la porte, un courant d'air passa et claqua la porte. Par conséquent, la tâche *walkThrough* se trouve bloquée car la porte est à présent fermée. L'exécution du plan est alors bloquée et une *cassure* est détectée.

La planification réactive [4] au contraire, fait le choix d'abandonner la phase de construction de plan et propose de définir un modèle d'exécution qui choisis la prochaine action à exécuter à partir d'un ensemble de différentes alternatives prédéfinies dans le domaine de connaissance et en tenant en compte les changements observés dans le monde. Le domaine de connaissance est défini en utilisant l'architecture hiérarchique des HTN avec un formalisme procédural pour la définition des tâches, pour cette raison nous appelons les planificateurs réactifs des *HTNs réactifs*. La définition du formalisme procédural se rapproche du monde réel ce qui le rends plus facile à modéliser et réduit par conséquent la complexité de la planification dans les domaines complexes. Néanmoins, les HTN réactifs restent limité. En effet, le formalise réactif n'utilise pas de modèle symbolique : l'exécution d'un HTN "réactif" repose sur des scripts procéduraux (Ex : JavaScript), fonctionnant comme des boîtes noires dont la seule fonctionnalité est l'exécution dans l'état courant. Donc si une cassure est détectéeă ; une cassure apparaît lorsque le système ne trouve

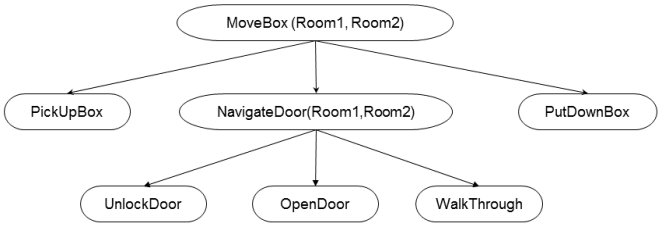

FIGURE 1 – la décomposition HTN pour la tâche MoveBox

plus, dans le modèle, d'actions pour continuer l'exécution. Les HTNs réactifs se trouvent dans l'impossibilité de raisonner pour trouver un plan de réparation. Pourtant, l'expérience montre que la majorité de ces scripts ont la forme de propositions, qui pourraient être représentées de manière symbolique. L'idée que nous défendons dans cet article est qu'il est possible, à moindre coût, d'augmenter la planification réactive à l'aide de connaissances symboliques pour permettre au système de réparer des cassures.
Dans la prochaine section, nous présentons un exemple qui illustre le problème des breakdowns en planification réactive. Nous présentons ensuite les travaux en planification symbolique et en planification réactive autour de la répération de breakdowns qui nous ont conduit à proposer notre modèle. Nous présentons dans la section 4 notre système Discolog et l'algorithme de replanification. La section 5 présente les résultats d'une première expérimentation qui étudie l'impact des connaissances symboliques sur la réparation de plans.

2 État de l'art

2.1 Le formalisme des HTNs

Le domaine de connaissances des HTNs est représenté comme un arbre AND/OR. les n ?uds AND représentent les tâches. Chaque tâche est définie avec des préconditions pour vérifier l'applicabilité de la tâche, et des post-conditions pour vérifier si la tâche s'est exécutée correctement. Les HTNs sont définis avec deux types de tâches ; les tâches primitives (les feuilles de l'arbre). Ces dernières peuvent être directement exécutées dans le monde. Les tâches non primitives qui doivent être décomposées en sous tâches en utilisant une recette de décomposition spécifique. Les recettes sont des n ?uds Or qui représentent des méthodes de décompositions d'une tâche non primitive. Les recettes sont définis avec des conditions d'applicabilités. Les HTNs planifient pour satisfaire des tâches buts en décomposant les tâches en sous tâches plus primitive jusqu'à l'obtention d'une séquence de tâches primitives qui une fois exécutées satisfassent la tâche bute.

2.2 Travaux connexes

Dans cette section, nous relatons les travaux éxistants qui traitent du problème de cassure dans les systèmes de planification. Les systèmes de planifications symboliques et spécialement les HTNs sont devenues performants pour produire des plans prometeurs avec d'excellents résultats dans des problèmes de planifications complexe. Nous citons SHOP [5] ou encore SIPE [6]. Cependant, ils restent sensibles aux cassures a cause des deux principales limitations présentées précédement.

Références

[1] K. Erol, "Hierarchical task network planning : formalization, analysis, and implementation," 1996.

[2] H. H. Zhuo, D. H. Hu, C. Hogg, Q. Yang, and H. Munoz-Avila, "Learning htn method preconditions and action models from partial observations.," in *IJCAI*, pp. 1804–1810, 2009.

[3] P. Maes, *Designing autonomous agents : Theory and practice from biology to engineering and back*. MIT press, 1990.

[4] R. J. Firby, "An investigation into reactive planning in complex domains.," in *AAAI*, vol. 87, pp. 202–206, 1987.

[5] D. Nau, Y. Cao, A. Lotem, and H. Muñoz-Avila, "Shop : Simple hierarchical ordered planner," in *Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*, pp. 968–973, Morgan Kaufmann Publishers Inc., 1999.

[6] D. E. Wilkins, *Practical planning : Extending the classical AI planning paradigm*. Morgan Kaufmann Publishers Inc., 1988.

[7] G. Boella and R. Damiano, "A replanning algorithm for a reactive agent architecture," in *Artificial Intelligence : Methodology, Systems, and Applications*, pp. 183–192, Springer, 2002.

[8] R. Van Der Krogt and M. De Weerdt, "Plan repair as an extension of planning.," in *ICAPS*, vol. 5, pp. 161–170, 2005.

[9] N. F. Ayan, U. Kuter, F. Yaman, and R. P. Goldman, "Hotride : Hierarchical ordered task replanning in dynamic environments," in *Planning and Plan Execution for Real-World Systems–Principles and Practices for Planning in Execution : Papers from the ICAPS Workshop. Providence, RI*, 2007.

[10] I. Warfield, C. Hogg, S. Lee-Urban, and H. Munoz-Avila, "Adaptation of hierarchical task network plans.," in *FLAIRS Conference*, pp. 429–434, 2007.

[11] C. Brom, "Hierarchical reactive planning : Where is its limit," *Proceedings of MNAS : Modelling Natural Action Selection. Edinburgh, Scotland*, 2005.